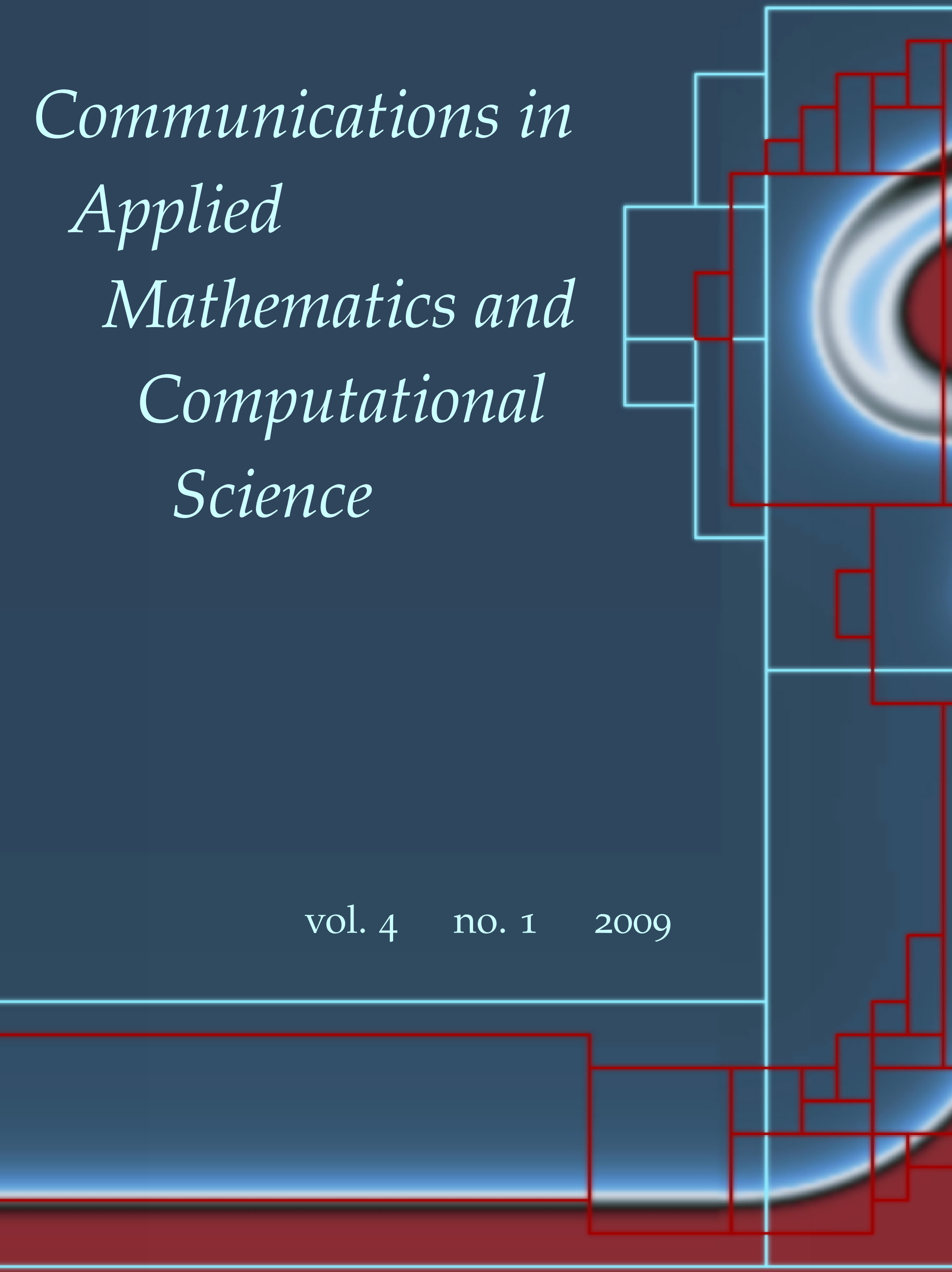
Communications in Applied Mathematics and Computational Science

vol. 4 no. 1 2009



mathematical sciences publishers

COMM. APP. MATH. AND COMP. SCI.
Vol. 4, No. 1, 2009

PARALLEL OVERLAPPING DOMAIN DECOMPOSITION METHODS FOR COUPLED INVERSE ELLIPTIC PROBLEMS

XIAO-CHUAN CAI, SI LIU AND JUN ZOU

We study an overlapping domain decomposition method for solving the coupled nonlinear system of equations arising from the discretization of inverse elliptic problems. Most algorithms for solving inverse problems take advantage of the fact that the optimality system has a natural splitting into three components: the state equation for the constraints, the adjoint equation for the Lagrange multipliers, and the equation for the parameter to be identified. Such algorithms often involve interiterations between the three separate solvers, and the intercomponent iteration is sequential. Several fully coupled or so-called one-shot approaches exist, and the main challenges in these approaches are that the system has stronger nonlinearity, and the corresponding Jacobian system is more ill-conditioned, in addition to being three times larger. Here we investigate a class of overlapping Newton–Krylov–Schwarz algorithms for solving such coupled systems, obtained with a pointwise ordering of the variables, and show numerically that, with a reasonably large overlap, the algorithm is capable of finding the solution even with noise and discontinuous coefficients. More importantly, we show that this approach is fully parallel and scalable with respect to the size of the problems.

1. Introduction

As parallel computers become more powerful, researchers are paying more attention to inverse problems which are more difficult and expensive to solve than forward problems [1; 11; 15; 24]. A key step in designing a high performance parallel algorithm is to formulate the problem with as few sequential calculations as possible. Here we study a parallel domain decomposition method for solving the system of nonlinear equations arising from the fully coupled finite difference discretization of some inverse elliptic problems in two-dimensional space.

MSC2000: 65N21, 65N55, 65Y05.

Keywords: inverse problems, domain decomposition, parallel computing, partial differential equations constrained optimiztion, inexact Newton.

The work of XCC and SL was partially supported by DOE FC02-04ER25595, NSF CNS 0722023 and CCF-0634894; the work of JZ was partially supported by the Hong Kong RGC grants (Projects 404105 and 404407).

Traditionally these problems are solved by using Uzawa-type algorithms which split the system into two or three subsystems solved individually. Subiterations are required between the subsystems. The subsystems are easier to solve than the global coupled system, but the iterations between subsystems are sequential in nature. There are several fully coupled approaches in which all variables are solved at the same time. They are often referred to as the one-shot method or the all-at-once method; see for example [3; 12; 16; 21]. In these approaches, the resulting linear and nonlinear systems of equations are three times larger, have stronger nonlinearity and are more ill-conditioned. Solving these fully coupled systems is a major challenge for any iterative methods.

The focus of this paper is to investigate a parallel domain decomposition preconditioning technique for the coupled systems. We show that with the powerful domain decomposition based preconditioner the convergence of the iterative methods can be obtained even for some difficult cases when the solution is discontinuous and when the observation data has high level of noise.

We consider an inverse elliptic problem [14]: Find the coefficient function $\rho(x)$ in the system

$$\begin{cases} -\nabla\cdot(\rho\nabla u) &= f, \quad x\in\Omega, \\ u(x) &= 0, \quad x\in\partial\Omega. \end{cases} \tag{1}$$

A widely used approach for solving the inverse problem is the output least-squares Tikhonov regularization method, which formulates the ill-posed inverse problem into different stabilized optimization problems, depending on the type of data available [6; 8; 13; 14]. For example, when the measurement of $u(x)$ is given, denoted as $z(x)$, the inverse problem can be transformed into the minimization problem:

$$\text{minimize} \;\; J(\rho,u) = \frac{1}{2}\int_\Omega (u-z)^2\,dx + \frac{\beta}{2}\int_\Omega |\nabla\rho|^2\,dx, \tag{2}$$

which is often referred to as the L^2 *least-squares problem*.

When the measurement of $\nabla u(x)$ is given, denoted as $\nabla z(x)$, the inverse problem can be transformed into the minimization problem:

$$\text{minimize} \;\; J(\rho,u) = \frac{1}{2}\int_\Omega \rho\,|\nabla u-\nabla z|^2\,dx + \frac{\beta}{2}\int_\Omega |\nabla\rho|^2\,dx, \tag{3}$$

which is often referred to as the H^1 *least-squares problem*. Both minimization problems (2) and (3) are subject to the constraint (1) satisfied by the pair (ρ,u), and the β-term is called the regularization term, and the constant β is the regularization parameter.

Instead of solving the constraint optimization problems (2) and (3), we turn to solving the saddle-point problems associated with the Lagrangian functional $\mathscr{L}$:

$$\mathscr{L}(\rho, u, \lambda) = \frac{1}{2}\int_\Omega (u-z)^2\,dx - \int_\Omega (\nabla\cdot\rho\nabla u + f)\lambda\,dx + \frac{\beta}{2}\int_\Omega |\nabla\rho|^2\,dx \tag{4}$$

for the L^2 case, or

$$\mathscr{L}(\rho, u, \lambda) = \frac{1}{2}\int_\Omega \rho|\nabla u - \nabla z|^2\,dx - \int_\Omega (\nabla\cdot\rho\nabla u + f)\lambda\,dx + \frac{\beta}{2}\int_\Omega |\nabla\rho|^2\,dx \tag{5}$$

for the H^1 case [8]. Hence the solutions to both minimization problems can be obtained by solving the corresponding optimality systems: Find (ρ, u, λ) such that

$$\begin{cases} (\nabla_\rho\mathscr{L})p = 0, \\ (\nabla_u\mathscr{L})w = 0, \\ (\nabla_\lambda\mathscr{L})\mu = 0, \end{cases} \tag{6}$$

for any (p, w, μ). More explicitly, we can reduce (6) to

$$\begin{cases} F^{(\rho)} \equiv -\beta\Delta\rho + \nabla u\cdot\nabla\lambda = 0, \\ F^{(u)} \equiv -\nabla\cdot(\rho\nabla\lambda) + (u-z) = 0, \\ F^{(\lambda)} \equiv -\nabla\cdot(\rho\nabla u) - f = 0, \end{cases} \tag{7}$$

in the L^2 case. Similarly, in the H^1 case, we have

$$\begin{cases} F^{(\rho)} \equiv -\beta\Delta\rho + \nabla u\cdot\nabla\lambda + 1/2|\nabla u - \nabla z|^2 = 0, \\ F^{(u)} \equiv -\nabla\cdot(\rho\nabla\lambda) + \nabla\cdot(\rho\nabla z) + f = 0, \\ F^{(\lambda)} \equiv -\nabla\cdot(\rho\nabla u) - f = 0. \end{cases} \tag{8}$$

Both systems (7) and (8) use the same boundary conditions

$$\begin{cases} (\partial\rho/\partial n) = 0, \\ u = 0, \\ \lambda = 0, \end{cases} \tag{9}$$

on $\partial\Omega$. The Dirichlet boundary conditions for u and λ are natural. The homogeneous Neumann boundary condition on ρ is the side effect of the H^1 regularization in (2) and (3). A derivation of the boundary condition $\partial\rho/\partial n = 0$ is given in the Appendix.

For solving the coupled systems, several techniques are available. For example, in [6; 13; 14], an augmented Lagrangian method was used and the solution was obtained by Uzawa-type algorithms, which decouples the problems into subproblems associated with ρ, u and λ separately, and as a result, only smaller problems need to be solved. The global convergence of these approaches was established in [7; 13]. In [2], a fully coupled discretization was used for some source term inverse problems, and the coupled system was first reduced by block eliminations and then solved by a conjugate gradient (CG) method. Tremendous progress has been made in using domain decomposition methods for optimization problems with partial differential equations constraints [1; 2; 17; 18; 19]. The Newton–Krylov–Schwarz methodology [18] studied here is a general purpose approach that is not attached to any specific formulation of the inverse problem. We require a globally convergent inexact Newton's method with line search for the nonlinear systems; a Krylov subspace method for the Jacobian systems; and, most importantly, an additive Schwarz preconditioner, which works well with a scheme we use for the orderings of the unknowns and the functions. Although we are still unable to show theoretically that the linear and nonlinear iterative solvers are convergent, our parallel numerical experiments have shown clearly that this approach works well for some rather difficult test cases with jump coefficients and a high level of noise.

Section 2 addresses some basic properties of the linear and nonlinear systems and a reordering scheme to avoid zero pivoting. We also describe a Newton–Krylov–Schwarz algorithm for solving the fully coupled systems. We give numerical experiments in Section 3, followed by concluding remarks in Section 4.

2. Scalable solvers

2.1. ***A reordered fully coupled system.*** We use the so-called fully coupled ordering, by which we mean that all three variables defined at the same mesh point are always together throughout the calculations. The unknowns are ordered mesh point by mesh point, in contrast to physical variable by physical variable as required by most existing sequential quadratic programming (SQP) methods. At each mesh point, x_{ij}, the unknowns are ordered in the order of $\rho_{ij}, u_{ij}, \lambda_{ij}$, that is,

$$U = (\rho_{11}, u_{11}, \lambda_{11}, \rho_{21}, u_{21}, \lambda_{21}, \ldots, \rho_{nx\,ny}, u_{nx\,ny}, \lambda_{nx\,ny})^T. \tag{10}$$

If we order the functions in exactly the same order,

$$F = (F_{11}^{(\rho)}, F_{11}^{(u)}, F_{11}^{(\lambda)}, F_{21}^{(\rho)}, F_{21}^{(u)}, F_{21}^{(\lambda)}, \ldots, F_{nx\,ny}^{(\rho)}, F_{nx\,ny}^{(u)}, F_{nx\,ny}^{(\lambda)})^T = 0, \tag{11}$$

then the Jacobian in the L^2 case is a symmetric matrix of the form (12) with dense 3×3 blocks of the form (13)

$$J = \left(\begin{array}{ccc|ccc|ccc} \times & \times & & \times & & & & & \\ \times & \times & \times & & \times & & & & \\ & \times & \times & & & \times & & & \\ \hline \times & & & \times & \times & & \times & & \\ & \times & & \times & \times & \times & & \times & \\ & & \times & & \times & \times & & & \times \\ \hline & & & \times & & & \times & \times & \\ & & & & \times & & \times & \times & \times \\ & & & & & \times & & \times & \times \end{array} \right)_{n \times n(block)} , \tag{12}$$

where

$$\times = \begin{pmatrix} * & * & * \\ * & * & * \\ * & * & 0 \end{pmatrix}_{3\times 3}. \tag{13}$$

However, the zero value on the diagonal of (13) causes a pivoting problem in our LU factorization based solvers. To avoid the zero pivot situation, we reorder the unknowns (switching the u variable with the λ variable):

$$U = (\rho_{11}, \lambda_{11}, u_{11}, \rho_{21}, \lambda_{21}, u_{21}, \dots, \rho_{nx\,ny}, \lambda_{nx\,ny}, u_{nx\,ny})^T = 0, \tag{14}$$

but keep the ordering of the functions unchanged as in (11). This reordering of function values does not change the block structure of the matrix, but the 3 × 3 block now takes the form:

$$\times = \begin{pmatrix} * & * & * \\ * & * & * \\ * & 0 & * \end{pmatrix}_{3\times 3}, \tag{15}$$

which is no longer symmetric. In this paper, our algorithms are not based on the structure of (13), but on the structure of (15), which is based on the ordering scheme (11) + (14).

For the purpose of parallel processing, the mesh points are ordered subdomain by subdomain. The ordering of the subdomains is not important since we use an additive method whose performance has nothing to do with the subdomain ordering.

2.2. ***Newton–Krylov method.*** The Newton–Krylov–Schwarz (NKS) methods [4] are a family of general-purpose parallel algorithms for solving systems of nonlinear algebraic equations. NKS has three main components: (i) an inexact Newton method for the nonlinear system; (ii) a Krylov subspace linear solver for the Jacobian systems (restarted GMRES [20]); and (iii) a Schwarz type preconditioner [22; 23]. We only study the regular additive Schwarz preconditioner in this paper, even though in some cases, the restricted version of the additive Schwarz method [5] maybe better.

We carry out the Newton iterations:

$$U_{k+1} = U_k - \lambda_k J(U_k)^{-1} F(U_k), \quad k = 0, 1, \dots, \tag{16}$$

where U_0 is an initial approximation to the solution, $J(U_k) = F'(U_k)$ is the Jacobian at U_k, and λ_k is the steplength determined by a linesearch procedure [9; 10]. The inexactness of Newton's method is reflected in the fact that we do not solve the Jacobian systems exactly. The accuracy of the Jacobian solver is determined by some $\eta_k \in [0, 1)$ and the condition

$$\|F(U_k) + J(U_k)s_k\| \le \eta_k \|F(U_k)\|. \tag{17}$$

The overall algorithm can be described as follows:

(1) Inexactly solve the linear system $J(U_k)s_k = -F(U_k)$ for s_k using a preconditioned GMRES(30).

(2) Perform a full Newton step with $\lambda_0 = 1$ in the direction s_k.

(3) If the full Newton step is unacceptable, we backtrack using the cubic backtracking procedure until a new λ is obtained that makes $U_+ = U_k + \lambda_k s_k$ an acceptable step.

(4) Set $U_{k+1} = U_+$ and return to step 1 unless a stopping condition has been met.

In step 1 above we use a right-preconditioned restarted GMRES to solve the linear system; that is, the vector s_k is obtained by approximately solving the right preconditioned Jacobian system

$$J(U_k)M_k^{-1}s_k' = -F(U_k),$$

where M_k^{-1} is a one-level additive Schwarz preconditioner and $s_k = M_k^{-1}s_k'$.

2.3. ***One-level additive Schwarz preconditioning.*** To formally define M_k^{-1}, we need to introduce a partition of Ω. We first partition the domain into nonoverlapping subdomains Ω_l, $l = 1, \dots, N$, where N is the same as the number of processors (np). In order to obtain an overlapping decomposition of the domain, we extend each subdomain Ω_l to a larger region Ω_l', that is, $\Omega_l \subset \Omega_l'$. Only simple box decomposition is considered in this paper — all the subdomains Ω_l and Ω_l' are rectangular and made up of integral numbers of fine mesh cells. The size of Ω_l is $H_x \times H_y$ and the size of Ω_l' is $H_x' \times H_y'$, where the H's are chosen so that the overlap (ovlp) is uniform in the number of fine grid cells all around the perimeter, that is,

$$\text{ovlp} = (H_x' - H_x)/2 = (H_y' - H_y)/2$$

for interior subdomains. For boundary subdomains, we simply cut off the part that is outside Ω.

On each extended subdomain Ω_l', we construct a subdomain preconditioner B_l which is the discretization of the Frechet derivative taken at the current iteration,

$$J = \begin{pmatrix} \dfrac{\partial F^{(\rho)}}{\partial \rho} & \dfrac{\partial F^{(\rho)}}{\partial \lambda} & \dfrac{\partial F^{(\rho)}}{\partial u} \\[2ex] \dfrac{\partial F^{(u)}}{\partial \rho} & \dfrac{\partial F^{(u)}}{\partial \lambda} & \dfrac{\partial F^{(u)}}{\partial u} \\[2ex] \dfrac{\partial F^{(\lambda)}}{\partial \rho} & \dfrac{\partial F^{(\lambda)}}{\partial \lambda} & \dfrac{\partial F^{(\lambda)}}{\partial u} \end{pmatrix}. \tag{18}$$

In the L^2 case, the Frechet derivative at the point (ρ, λ, u) takes the form

$$F'_{L^2} = \begin{pmatrix} -\beta\Delta & \nabla u\cdot\nabla & \nabla\lambda\cdot\nabla \\ -(\Delta\lambda+\nabla\lambda\cdot\nabla) & -\nabla\cdot(\rho\nabla) & I \\ -(\Delta u+\nabla u\cdot\nabla) & 0 & -\nabla\cdot(\rho\nabla) \end{pmatrix}. \tag{19}$$

Similarly, in the H^1 case, we have

$$F'_{H^1} = \begin{pmatrix} -\beta\Delta & \nabla u\cdot\nabla & \nabla\lambda\cdot\nabla+(\nabla u-\nabla z)\cdot\nabla \\ -(\Delta\lambda+\nabla\lambda\cdot\nabla)+(\Delta z+\nabla z\cdot\nabla) & -\nabla\cdot(\rho\nabla) & 0 \\ -(\Delta u+\nabla u\cdot\nabla) & 0 & -\nabla\cdot(\rho\nabla) \end{pmatrix}. \tag{20}$$

Using the derivative and some boundary conditions, we can define the subdomain problems. For example, in the L^2 case, we have

$$\begin{cases} -\beta\Delta p+\nabla\lambda\cdot\nabla w+\nabla u\cdot\nabla\mu = g_1 & \text{in } \Omega'_l, \\ -\nabla\cdot(p\nabla\lambda)+w-\nabla\cdot(\rho\nabla\mu) = g_2 & \text{in } \Omega'_l, \\ -\nabla\cdot(p\nabla u)-\nabla\cdot(\rho\nabla w) = g_3 & \text{in } \Omega'_l, \\ p = w = \mu = 0 & \text{on } \partial\Omega'_l\cap\Omega, \\ (\partial p/\partial n) = w = \mu = 0 & \text{on } \partial\Omega'_l\cap\partial\Omega. \end{cases} \tag{21}$$

The solution (p, w, μ) and the right side (g_1, g_2, g_3) of the subdomain problem are not important at all. We only need the operator form (21) to construct a local solver B_l defined on the subdomain $\partial\Omega'_l$. Note that homogeneous Dirichlet boundary conditions are used on the internal subdomain boundary, and the original boundary conditions are used on the physical boundary, if present. A similar system is used for the H^1 least-squares problem.

Alternatively we can obtain B_l by extracting its elements from the global Jacobian matrix; that is, $B_l^{i,j} = \{J_{ij}\}$, where the node indexed by (i, j) belongs to the interior of Ω'_l. The entry J_{ij} is calculated with finite differences $J_{ij} = (F_i(U_j+\epsilon) - F_i(U_j))/\epsilon$, where $0 < \epsilon \ll 1$ is a constant. The additive Schwarz preconditioner can be written as

$$M_k^{-1} = I_1 B_1^{-1}(I_1)^T + \cdots + I_N B_N^{-1}(I_N)^T. \tag{22}$$

Let n be the total number of mesh points, and n'_l the total number of mesh points in Ω'_l, then I_l is an $3n \times 3n'_l$ extension matrix that extends each vector defined on Ω'_l to a vector defined on the entire fine mesh by padding an $3n'_l \times 3n'_l$ identity matrix with zero rows.

3. Numerical experiments

In this paper, we assume the problem is defined on $\Omega = (0, l_x) \times (0, l_y)$, which is covered by a uniform mesh of size h. To discretize the equations we use the usual

5-point central finite difference method for all variables. For the L^2 formulation (7), we define

$$F_{ij}^{(\rho)} = \beta\,\frac{4\rho_{ij} - \rho_{i-1\,j} - \rho_{i+1\,j} - \rho_{i\,j-1} - \rho_{i\,j+1}}{h^2} + \frac{u_{i+1\,j} - u_{i-1\,j}}{2h}\,\frac{\lambda_{i+1\,j} - \lambda_{i-1\,j}}{2h} + \frac{u_{i\,j+1} - u_{i\,j-1}}{2h}\,\frac{\lambda_{i\,j+1} - \lambda_{i\,j-1}}{2h},$$

$$F_{ij}^{(u)} = -\,\frac{\rho_{i+1/2\,j}(\lambda_{i+1\,j} - \lambda_{i\,j}) - \rho_{i-1/2\,j}(\lambda_{i\,j} - \lambda_{i-1\,j})}{h^2} - \frac{\rho_{i\,j+1/2}(\lambda_{i\,j+1} - \lambda_{i\,j}) - \rho_{i\,j-1/2}(\lambda_{i\,j} - \lambda_{i\,j-1})}{h^2} + (u_{ij} - z_{ij}),$$

$$F_{ij}^{(\lambda)} = -\,\frac{\rho_{i+1/2\,j}(u_{i+1\,j} - u_{i\,j}) - \rho_{i-1/2\,j}(u_{i\,j} - u_{i-1\,j})}{h^2} - \frac{\rho_{i\,j+1/2}(u_{i\,j+1} - u_{i\,j}) - \rho_{i\,j-1/2}(u_{i\,j} - u_{i\,j-1})}{h^2} - f_{ij},$$

where the half-point values of ρ are calculated using the average of the two neighboring values. Similarly, we obtain the discretization of the H^1 formulation (8):

$$F_{ij}^{(\rho)} = \beta\,\frac{4\rho_{ij}-\rho_{i-1\,j}-\rho_{i+1\,j}-\rho_{i\,j-1}-\rho_{i\,j+1}}{h^2} + \frac{u_{i+1\,j}-u_{i-1\,j}}{2h}\,\frac{\lambda_{i+1\,j}-\lambda_{i-1\,j}}{2h} + \frac{u_{i\,j+1}-u_{i\,j-1}}{2h}\,\frac{\lambda_{i\,j+1}-\lambda_{i\,j-1}}{2h} + \frac{1}{2}\left(\left(\frac{u_{i+1\,j}-u_{i-1\,j}}{2h} - \nabla_x z|_{ij}\right)^2 + \left(\frac{u_{i\,j+1}-u_{i\,j-1}}{2h} - \nabla_y z|_{ij}\right)^2\right),$$

$$F_{ij}^{(u)} = -\,\frac{\rho_{i+1/2\,j}(\lambda_{i+1\,j}-\lambda_{i\,j})-\rho_{i-1/2\,j}(\lambda_{i\,j}-\lambda_{i-1\,j})}{h^2} - \frac{\rho_{i\,j+1/2}(\lambda_{i\,j+1}-\lambda_{i\,j})-\rho_{i\,j-1/2}(\lambda_{i\,j}-\lambda_{i\,j-1})}{h^2} + \frac{\rho_{i+1\,j}\nabla_x z|_{i+1\,j}-\rho_{i-1\,j}\nabla_x z|_{i-1\,j}}{2h} + \frac{\rho_{i\,j+1}\nabla_y z|_{i\,j+1}-\rho_{i\,j-1}\nabla_y z|_{i\,j-1}}{2h} + f_{ij},$$

$$F_{ij}^{(\lambda)} = -\,\frac{\rho_{i+1/2\,j}(u_{i+1\,j}-u_{i\,j})-\rho_{i-1/2\,j}(u_{i\,j}-u_{i-1\,j})}{h^2} - \frac{\rho_{i\,j+1/2}(u_{i\,j+1}-u_{i\,j})-\rho_{i\,j-1/2}(u_{i\,j}-u_{i\,j-1})}{h^2} - f_{ij}.$$

To form an algebraic system of nonlinear equations from the finite difference equations, we need to order the unknowns and the corresponding functions. The ordering of the unknowns and the equations is not a big deal at all for accuracy concerns, but is critically important for the linear Jacobian solver and for the preconditioning techniques. For example, if we order the unknowns variable by variable — that is, first ρ values for all mesh points, and second u values for all mesh points, and last λ values for all mesh points, then the Jacobian matrix takes the following block form:

$$J = \begin{pmatrix} \dfrac{\partial F^{(\rho)}}{\partial \rho} & \dfrac{\partial F^{(\rho)}}{\partial u} & \dfrac{\partial F^{(\rho)}}{\partial \lambda} \\ \dfrac{\partial F^{(u)}}{\partial \rho} & \dfrac{\partial F^{(u)}}{\partial u} & \dfrac{\partial F^{(u)}}{\partial \lambda} \\ \dfrac{\partial F^{(\lambda)}}{\partial \rho} & \dfrac{\partial F^{(\lambda)}}{\partial u} & \dfrac{\partial F^{(\lambda)}}{\partial \lambda} \end{pmatrix} = \begin{pmatrix} \dfrac{\partial F^{(\rho)}}{\partial \rho} & \dfrac{\partial F^{(\rho)}}{\partial u} & \dfrac{\partial F^{(\rho)}}{\partial \lambda} \\ \dfrac{\partial F^{(u)}}{\partial \rho} & I & \dfrac{\partial F^{(u)}}{\partial u} \\ \dfrac{\partial F^{(\lambda)}}{\partial \rho} & \dfrac{\partial F^{(\lambda)}}{\partial u} & 0 \end{pmatrix}_{3n\times 3n} . \tag{23}$$

Many interesting algorithms are designed based on the particular block structure of (23), and many algorithms fail to work also because of the structure of (23).

We study the performance of the proposed algorithm using four test cases. The first test case is from [14], and the purpose is to verify the accuracy of the algorithm. To understand the scalability of the algorithm, we introduce three more test problems.

To test the robustness of the algorithms, we add some noise to the observation data as

$$z^{\delta} = z + \delta \ \text{rand}\ (x, y) \tag{24}$$

or

$$\nabla z^{\delta} = \nabla z + \delta\,(\text{rand}\ (x, y),\ \text{rand}\ (x, y))^{T}, \tag{25}$$

depending on whether the formulation is L^2 or H^1. Here rand (x, y) is a random scalar function available in the C library, and δ is responsible for the magnitude of the noise. Some results with different levels of noise ($\delta = 0\%$, 1% and 10%) will be presented. Since u needs to satisfy the elliptic equation, we assume that u has some continuity and differentiability, as does ∇u. Therefore, we smooth u in L^2 formulation or ∇u in H^1 formulation before we start the Newton iteration. This is necessary especially when the noise level is high. In particular, when the noise level is 10%, we replace the value of u or ∇u by the weighted average value around it. And the weight function is defined as

$$\begin{tikzcd} \frac{1}{16} \arrow[dr] & \frac{1}{8} \arrow[d] & \frac{1}{16} \arrow[dl] \\ \frac{1}{8} \arrow[r] & \frac{1}{4} & \frac{1}{8} \arrow[l] \\ \frac{1}{16} \arrow[ur] & \frac{1}{8} \arrow[u] & \frac{1}{16} \arrow[ul] \end{tikzcd} .$$

We repeat this operation 3 times in all of our tests when $\delta = 10\%$. No smoothing is applied when δ is smaller than 10%.

To measure the accuracy of the numerical solution, we assume that the exact solution of the test cases are known, and error_u and error_ρ are the normalized discrete L^2 norms of the errors defined by

$$\text{error}_u = \sqrt{\sum (u_{ij} - u_{ij}^{\text{exact}})^2 \frac{h_x h_y}{l_x l_y}} \qquad \text{and} \qquad \text{error}_\rho = \sqrt{\sum (\rho_{ij} - \rho_{ij}^{\text{exact}})^2 \frac{h_x h_y}{l_x l_y}},$$

where h_x and h_y are mesh sizes along x and y directions, and l_x and l_y are sizes of the computational domain along the x and y directions, respectively.

In Newton's method, we use the initial guess

$$U_0 = (\rho^{(0)}, u^{(0)}, \lambda^{(0)})^T = (1, z, 0)^T.$$

For the L^2 formulation, z is the observation value. For the H^1 formulation, z is not directly available, but is obtained as a line integral of $\nabla_x z$ or $\nabla_y z$ along the x or y direction from one of the boundary points. In our test, at the point (x_i, y_j),

$$z(x_i, y_j) = z(x_0, y_j) + \sum_{l=1}^{i} (\nabla_x z)|_{x_l} h_x$$

if we take the integral along the x direction, or

$$z(x_i, y_j) = z(x_i, y_0) + \sum_{l=1}^{j} (\nabla_y z)|_{y_l} h_y$$

if we take the integral along the y direction.

In the test runs, we stop the Newton iteration if the following condition is satisfied

$$\|F(U_k)\| \le \max\left\{10^{-6}\|F(U_0)\|, 10^{-10}\right\}. \tag{26}$$

For the Jacobian solver, the GMRES iteration is stopped if

$$\|F(U_k) + J(U_k)s_k\| \le \max\left\{10^{-6}\|F(U_k)\|, 10^{-10}\right\}. \tag{27}$$

All the subdomain problems are solved with LU factorization. We implement the proposed algorithms using the Portable Extensible Toolkit for Scientific Computation (PETSc), developed at Argonne National Laboratory. Note that the timing results are obtained on a cluster of Linux PCs. The timings are just for references and should not be taken too seriously since the network of the PC cluster is slow and is also shared by many users.

3.1. ***Test cases.*** We next describe four test cases with the observation function

$$z(x, y) = \sin(\pi x)\sin(\pi y),$$

and several different ρ functions and on several different computational domains.

Test 1. In the first test we take $\Omega = (0, 1) \times (0, 1)$, and the right side f is constructed such that

$$\rho = 1 + 6x^2 y(1 - y)$$

is the elliptic coefficient to be identified. Note that this function does not satisfy $\partial\rho/\partial n = 0$ on the north boundary ($y = 1$), the south boundary ($y = 0$) and the east boundary ($x = 1$) of the domain.

Test 2. In the second test we take $\Omega = (0, 1) \times (0, 1)$ and the right side f is chosen so that the elliptic coefficient to be identified is

$$\rho = 1 + 100(xy(1 - x)(1 - y))^2.$$

Note that this function satisfies $\partial\rho/\partial n = 0$ on the entire boundary of the domain.

Test 3. In the third test we take a domain $\Omega = (0, l_x) \times (0, l_y)$ and the right side f is chosen so that the elliptic coefficient to be identified is

$$\rho = 1 + (-1)^{l_i+l_j} 6\big[(x - l_i)(y - l_j)(1 - (x - l_i))(1 - (y - l_j))\big]^2$$

when $(x, y) \in [l_i, l_i + 1) \times [l_j, l_j + 1)$, l_i and l_j are integers less than l_x and l_y, respectively.

We mention that ρ is a smooth function. Several different values of l_x and l_y are tested and the details are given where the test results are shown.

Test 4. In the fourth test we take a domain $\Omega = (0, l_x) \times (0, l_y)$ and the right side f is chosen so that the elliptic coefficient to be identified is

$$\rho = \begin{cases} 1 & \text{for } x - l_i \le 1/2, \\ 2 & \text{for } x - l_i > 1/2, \end{cases} \tag{28}$$

when l_i is even, $x \in [l_i, l_i + 1)$ and

$$\rho = \begin{cases} 2 & \text{for } x - l_i \le 1/2, \\ 1 & \text{for } x - l_i > 1/2, \end{cases} \tag{29}$$

when l_i is odd, $x \in [l_i, l_i + 1)$, and l_i is an integer less than l_x. This is a piecewise constant function defined on the computational domain and this function has several jumps along the x-direction.

Our discretization scheme and the solution algorithms do not require any a priori knowledge of the locations of the jumps. We mention that there are several techniques that are designed specifically for problems with discontinuous coefficients [6; 7; 14].

3.2. *Results and discussions of numerical experiments.*

3.2.1. *Test 1.* In this test, we solve the Jacobian systems using a global Gaussian elimination, therefore the domain decomposition preconditioned iterative solver introduced in the previous section plays no role at all. As mentioned earlier, this equation does not satisfy the boundary condition

$$\frac{\partial \rho}{\partial n} = 0.$$

We see from Figure 1 that the numerical solution has some visible difference from the exact solution. To satisfy the above boundary condition, the numerical solution has some distortion on the north, south and east boundaries. This is more obvious near the two corners. Nevertheless, the results match that of [14]. When the noise level is high, the effect of $\partial \rho / \partial n = 0$ is larger near the corners for the L^2 formulation of the inverse problem. The H^1 formulation is less sensitive to this boundary condition.

From Table 1, we see that our method converges well. It takes about 3–5 Newton iterations to converge. For a given level of noise, the results are more accurate when we choose a finer mesh. When the noise level is high, we need larger β values for the Newton to converge. Comparing the solution plots in Figure 1, we see that the H^1 formulation generally gives us better solutions than the L^2 formulation, but somehow it takes slightly more Newton iterations to converge than the L^2 case. This is also true for our other test problems.

3.2.2. *Test 2.* The exact and numerical solutions are shown in Figure 2. It can be seen that the numerical solutions are quite accurate (Table 2) in the whole domain since the equation satisfies the Neumann boundary condition $\partial \rho / \partial n = 0$ on all its boundary. This is different from Test 1 where the accuracy is low near the corners. We tested three meshes 41×41, 81×81, and 161×161. When the Jacobian systems are solved exactly with a global Gaussian elimination, the total number of Newton iterations ranges from 3–6, and the iteration numbers are not sensitive to the level of noise.

We next look into the performance of the Newton–Krylov–Schwarz algorithm, in particular, we would like to know how the convergence depends on the mesh size, the number of subdomains, and the overlapping size. First, we study the processor-scalability. We solve the problem on a 321×321 mesh using different numbers of processors. We show the results, in terms of iteration numbers and the computing time totals, in Table 3. The number of Newton iterations does not change when we change the number of processors or the overlapping size. If we fix the number of subdomains or the number of processors, as we increase the overlapping size, the number of GMRES iterations decreases. The computing time decreases to a point and then begins to increase. This suggests that an optimal overlapping

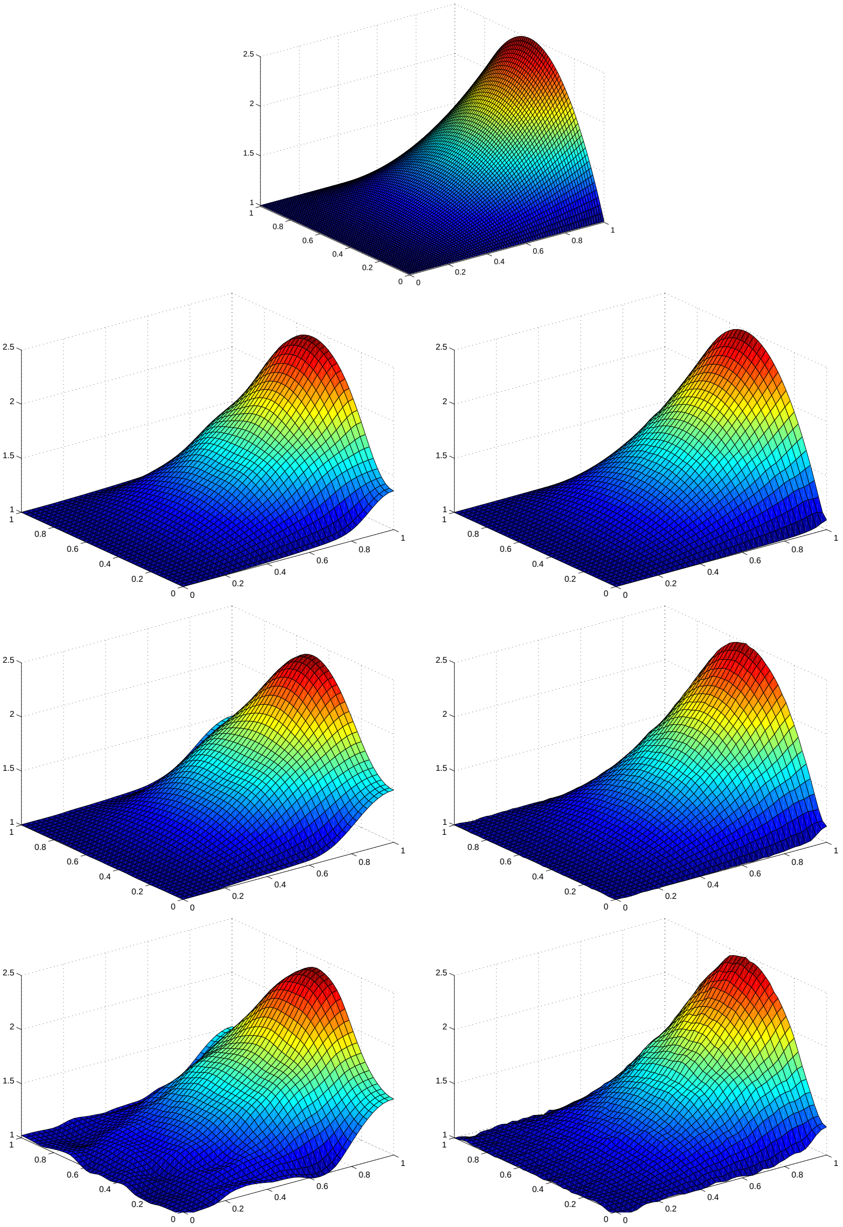


Figure 1. Test 1. Top: surface plot of the exact solution ρ. Bottom six pictures are numerical solutions with $\delta = 0\%$, $\delta = 1\%$ and $\delta = 10\%$. Left: L^2 formulation; right: H^1 formulation.

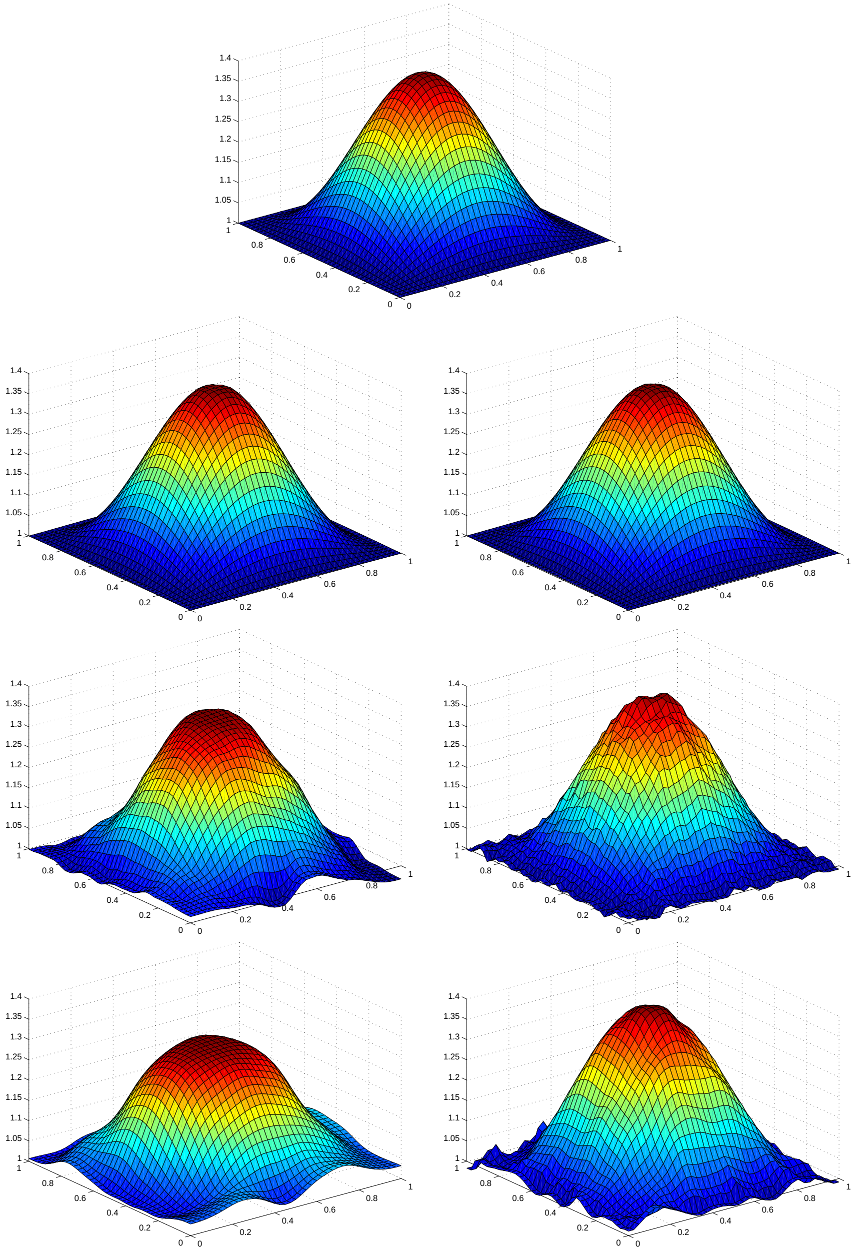


Figure 2. Test 2. Top: surface plot of exact solution ρ. Bottom six pictures are numerical solutions with $\delta = 0\%$, $\delta = 1\%$ and $\delta = 10\%$. Left: L^2 formulation; right: H^1 formulation.

	error$_u$	error$_\rho$	Newton
L^2 formulation, 41 × 41 mesh			
$\beta = 10^{-6}, \delta = 0$	0.000535	0.042644	3
$\beta = 10^{-5}, \delta = 1\%$	0.002032	0.073478	4
$\beta = 10^{-4}, \delta = 10\%$	0.009951	0.143455	4
L^2 formulation, 81 × 81 mesh			
$\beta = 10^{-6}, \delta = 0$	0.000455	0.034766	3
$\beta = 10^{-5}, \delta = 1\%$	0.001759	0.062192	4
$\beta = 10^{-4}, \delta = 10\%$	0.007615	0.119419	4
L^2 formulation, 161 × 161 mesh			
$\beta = 10^{-6}, \delta = 0$	0.000424	0.031326	3
$\beta = 10^{-5}, \delta = 1\%$	0.001683	0.058537	4
$\beta = 10^{-4}, \delta = 10\%$	0.006975	0.113078	4
H^1 formulation, 41 × 41 mesh			
$\beta = 10^{-5}, \delta = 0$	0.000277	0.020434	5
$\beta = 10^{-5}, \delta = 1\%$	0.000302	0.020677	5
$\beta = 10^{-4}, \delta = 10\%$	0.006932	0.036644	5
H^1 formulation, 81 × 81 mesh			
$\beta = 10^{-5}, \delta = 0$	0.000083	0.010343	4
$\beta = 10^{-5}, \delta = 1\%$	0.000103	0.010697	4
$\beta = 10^{-4}, \delta = 10\%$	0.001959	0.021829	4
H^1 formulation, 161 × 161 mesh			
$\beta = 10^{-6}, \delta = 0$	0.000018	0.003760	5
$\beta = 10^{-5}, \delta = 1\%$	0.000039	0.007377	4
$\beta = 10^{-4}, \delta = 10\%$	0.000496	0.017599	5

Table 1. Test 1. Errors and number of Newton iterations for three meshes with three levels of noise in L^2 and H^1 formulations.

size exists if the goal is to minimize the total computing time when the number of processors is not changed. On the fixed mesh, and with a fixed overlapping size, the number of GMRES iterations increases as we use more processors. This is expected since this is a single-level algorithm. Second, we check the h-scalability of our algorithm. Here, we increase the mesh size for the test problem and the number of processors at the same ratio in order for each processor to have a fixed number of mesh points. Table 4 shows the results with different overlapping sizes for np= 4,

	error_u	error_ρ	Newton
L^2 formulation, 41×41 mesh			
$\beta = 10^{-6}, \delta = 0$	0.000078	0.003163	3
$\beta = 10^{-5}, \delta = 1\%$	0.000765	0.010723	3
$\beta = 10^{-4}, \delta = 10\%$	0.008222	0.038667	3
L^2 formulation, 81×81 mesh			
$\beta = 10^{-6}, \delta = 0$	0.000073	0.003177	3
$\beta = 10^{-5}, \delta = 1\%$	0.000532	0.010070	3
$\beta = 10^{-4}, \delta = 10\%$	0.003849	0.029056	3
L^2 formulation, 161×161 mesh			
$\beta = 10^{-6}, \delta = 0$	0.000072	0.003203	3
$\beta = 10^{-5}, \delta = 1\%$	0.000504	0.009908	3
$\beta = 10^{-5}, \delta = 10\%$	0.002064	0.026190	4
H^1 formulation, 41×41 mesh			
$\beta = 10^{-5}, \delta = 0$	0.000385	0.001559	5
$\beta = 10^{-5}, \delta = 1\%$	0.000377	0.005097	6
$\beta = 10^{-4}, \delta = 10\%$	0.006927	0.020951	4
H^1 formulation, 81×81 mesh			
$\beta = 10^{-5}, \delta = 0$	0.000089	0.000386	4
$\beta = 10^{-5}, \delta = 1\%$	0.000108	0.003493	4
$\beta = 10^{-4}, \delta = 10\%$	0.001907	0.009897	4
H^1 formulation, 161×161 mesh			
$\beta = 10^{-6}, \delta = 0$	0.000022	0.000098	4
$\beta = 10^{-5}, \delta = 1\%$	0.000029	0.002355	4
$\beta = 10^{-4}, \delta = 10\%$	0.000460	0.006295	4

Table 2. Test 2. Errors and number of Newton iterations for three meshes with three levels of noise in L^2 and H^1 formulations.

16 and 64. Both the number of Newton iterations and the number of GMRES iterations are almost constants. The computing time is close to be quadrupled when the size of the problem is quadrupled with np fixed.

3.2.3. *Test 3.* For forward elliptic problems, one can always obtain a large test problem (with a large number of degree of freedoms) by refining the mesh, but for inverse elliptic problems, sometimes it does not make sense to use a very fine mesh

np	Newton	ovlp= 1	ovlp= 2	ovlp= 4	ovlp= 8	ovlp= 16
L^2 formulation, $\beta = 10^{-6}$, $\delta = 0\%$						
4	3	45(134.68)	33(124.00)	19(111.96)	13(111.61)	8(118.08)
16	3	66(28.17)	46(24.35)	34(23.18)	22(24.73)	14(32.06)
64	3	128(8.73)	92(7.36)	63(6.64)	42(7.20)	25(9.17)
L^2 formulation, $\beta = 10^{-5}$, $\delta = 1\%$						
4	3	43(131.74)	26(115.07)	19(111.84)	14(111.91)	9(118.45)
16	3	61(26.86)	45(23.89)	31(22.37)	23(24.55)	15(32.46)
64	3	134(8.99)	94(7.44)	62(6.59)	45(7.41)	25(9.74)
L^2 formulation, $\beta = 10^{-5}$, $\delta = 10\%$						
4	4	49(184.48)	36(168.30)	23(154.74)	16(152.48)	10(159.11)
16	4	72(39.41)	54(34.92)	40(32.98)	25(34.12)	19(44.94)
64	4	179(15.21)	118(11.82)	79(10.30)	54(10.99)	35(14.75)
H^1 formulation, $\beta = 10^{-5}$, $\delta = 0\%$						
4	5	52(233.36)	34(202.76)	21(184.87)	14(182.29)	12(194.77)
16	4	96(47.03)	63(37.67)	41(32.62)	26(33.54)	17(43.48)
64	4	171(14.44)	110(11.07)	71(9.50)	46(9.81)	25(12.77)
H^1 formulation, $\beta = 10^{-5}$, $\delta = 1\%$						
4	5	48(227.45)	33(200.85)	20(184.23)	14(182.46)	10(194.16)
16	4	75(39.69)	55(34.75)	30(28.65)	22(31.67)	15(42.30)
64	4	142(11.60)	89(9.45)	53(7.79)	40(9.14)	23(12.28)
H^1 formulation, $\beta = 10^{-4}$, $\delta = 10\%$						
4	4	61(199.28)	43(176.87)	26(156.70)	18(151.73)	12(159.66)
16	4	89(44.54)	60(36.26)	45(34.41)	26(33.77)	18(43.76)
64	4	141(12.23)	104(10.58)	66(9.01)	46(9.82)	26(12.94)

Table 3. Test 2. Processor scalability in L^2 and H^1 formulations on a 321 × 321 mesh, with total number of Newton iterations, average number of GMRES iterations per Newton, and total computing time in seconds shown in ( ), with different overlapping sizes.

since the accuracy is determined by the mesh size and the level of noise. When the level of noise is fixed, one may not always obtain a better solution even if a finer mesh is used. To test the scalability of the proposed algorithm and software, we construct larger test problems by increasing the computational domain.

In this test case the solution has multiple peaks, and the exact and numerical solutions are shown in Figure 3. We observe that the errors are kept at the same level when we change the size of the computational domain for different numbers of processors. The H^1 results are a little better than the L^2 results. According to the results shown in Table 5 we see that the number of Newton iterations is almost a constant for different numbers of processors, but the number of GMRES iterations slightly increases, which is expected for a single-level method. In some cases when the computational domain is very large and the noise level is high, Newton fails to converge unless we use a larger β.

3.2.4. *Test 4.* This problem is also defined on a large domain as in Test 3. The difference is that ρ is a discontinuous function with several jumps in the x-direction as shown in Figure 4. The surface plots of the computational results of ρ in L^2 and H^1 formulation are shown in Figure 4. The results obtained with the L^2 formulation are continuous and quite smooth even if the actual solutions should have jumps. The H^1 formulation leads to piecewise continuous solutions and keeps the discontinuity much better than the L^2 formulation. As for the scalability of the algorithm, Table 6 shows that the performance is very similar to that of Test 3.

4. Concluding remarks

A parallel one-level Newton–Krylov–Schwarz method was investigated for solving the nonlinear system of algebraic equations arising from the fully coupled finite difference discretization of inverse elliptic problems in both the L^2 and H^1 formulations. We tested the algorithms for problems with smooth solutions and for problems with discontinuous solutions. Acceptable solutions were obtained even when the level of noise is quite large. The mesh and processor scalabilities of the algorithm were studied for meshes up to 641 × 641 in two dimensions and with up to 64 processors. The iterative method was optimally scalable with respect to the mesh size. The number of iterations increases as the number of processors increases, which was expected for the one-level method. The algorithmic and software framework is applicable to a wide range of inverse problems, and our future research includes the extension of the algorithms to three dimensions, the study of other regularization techniques and their impact on the linear and nonlinear solvers, and the development of multilevel versions of the algorithm which will be needed for parallel computers with a large number of processors.

Appendix

In this section, we prove that if we choose the regularization term

$$N(\rho) = \frac{1}{2}\int_\Omega |\nabla \rho|^2 \, dx,$$

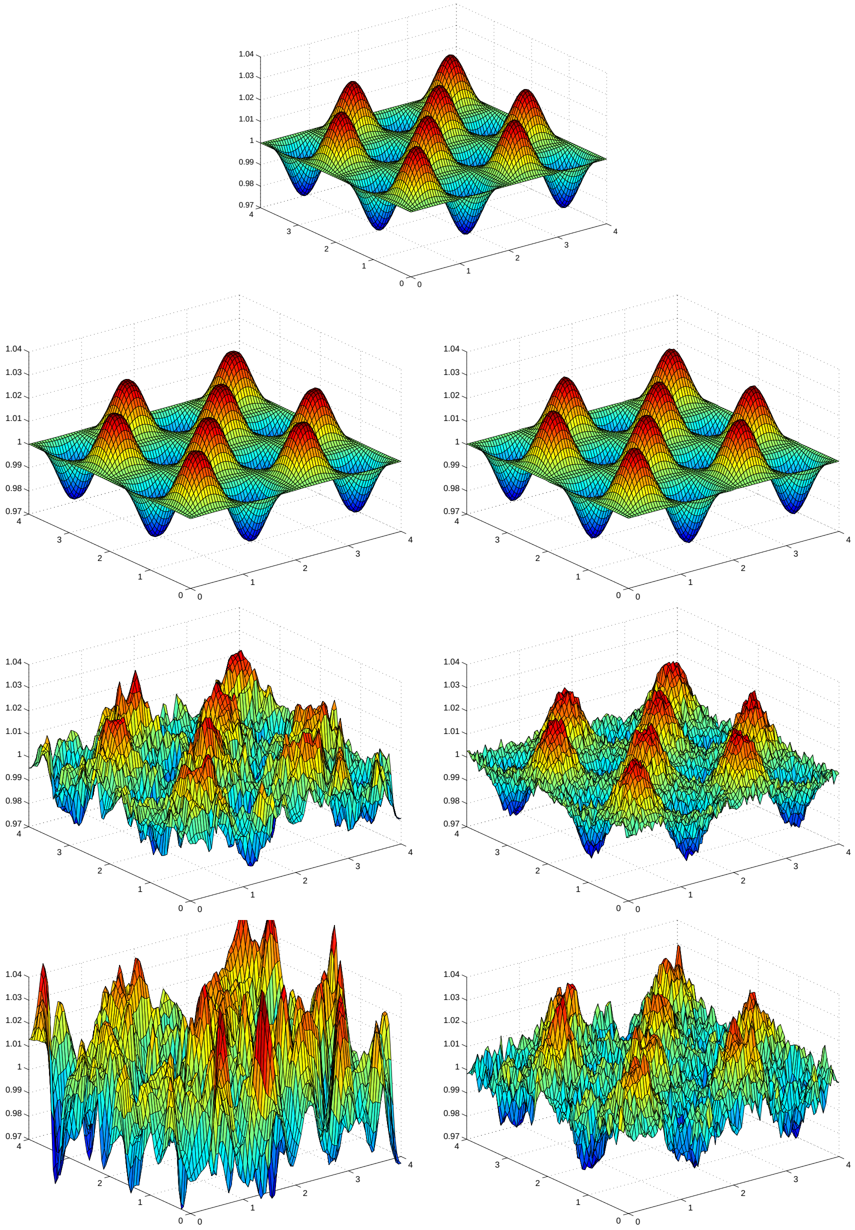


Figure 3. Test 3 on computational domain $(0,4) \times (0,4)$. Top: surface plot of the exact solution ρ. Bottom six pictures are numerical solutions with $\delta = 0\%$, $\delta = 1\%$ and $\delta = 10\%$. Left: L^2 formulation; right: H^1 formulation.

np	Newton	GMRES	Newton	GMRES	Newton	GMRES
	81×81 mesh		161×161 mesh		321×321 mesh	
L^2 formulation, $\beta = 10^{-6}$, $\delta = 0\%$						
4	3	7(2.84)	3	6(18.97)	3	6(142.96)
16	3	14(0.69)	3	14(4.51)	3	14(32.06)
64	3	38(0.32)	3	40(1.16)	3	42(7.21)
L^2 formulation, $\beta = 10^{-5}$, $\delta = 1\%$						
4	3	7(2.85)	3	7(19.19)	3	6(143.01)
16	3	18(0.75)	3	17(4.73)	3	15(32.48)
64	3	47(0.38)	3	45(1.24)	3	45(7.41)
L^2 formulation, $\beta = 10^{-4}$, $\delta = 10\%$						
4	3	9(2.99)	3	8(19.72)	3	8(146.54)
16	3	24(0.86)	3	23(5.34)	3	22(35.48)
64	3	75(0.54)	3	72(1.69)	3	66(9.38)
H^1 formulation, $\beta = 10^{-5}$, $\delta = 0\%$						
4	4	7(3.61)	4	7(24.77)	4	7(234.55)
16	4	17(0.94)	4	19(4.80)	4	17(43.50)
64	4	44(0.47)	4	47(1.23)	4	46(9.79)
H^1 formulation, $\beta = 10^{-5}$, $\delta = 1\%$						
4	4	8(3.66)	4	7(24.55)	5	7(234.02)
16	4	16(0.93)	4	15(5.89)	4	15(42.30)
64	4	44(0.47)	4	41(1.49)	4	40(9.15)
H^1 formulation, $\beta = 10^{-4}$, $\delta = 10\%$						
4	4	8(3.70)	4	8(25.23)	4	8(190.87)
16	4	17(0.95)	4	17(6.16)	4	18(43.77)
64	4	41(0.45)	4	48(1.66)	4	46(9.82)

Table 4. Test 2. Mesh size scalability in L^2 and H^1 formulations, with total number of Newton iterations, average number of GMRES iterations per Newton, and total computing time in seconds, shown in ( ), for different meshes and numbers of processors; ovlp $= 1/5$ diameter of the subdomain.

np	domain	mesh	error$_u$	error$_\rho$	Newton	GMRES
L^2 formulation, $\beta = 10^{-6}$, $\delta = 0\%$						
1	$(0, 1) \times (0, 1)$	81×81	0.000004	0.000177	2	1
4	$(0, 2) \times (0, 2)$	161×161	0.000004	0.000190	2	15
16	$(0, 4) \times (0, 4)$	321×321	0.000004	0.000195	2	31
64	$(0, 8) \times (0, 8)$	641×641	0.000005	0.000220	2	130
L^2 formulation, $\beta = 10^{-5}$, $\delta = 1\%$						
1	$(0, 1) \times (0, 1)$	81×81	0.000359	0.004668	3	1
4	$(0, 2) \times (0, 2)$	161×161	0.000383	0.004655	3	14
16	$(0, 4) \times (0, 4)$	321×321	0.000381	0.004637	3	30
64	$(0, 8) \times (0, 8)$	641×641	0.000378	0.004637	3	61
L^2 formulation, $\beta = 10^{-4}$, $\delta = 10\%$						
1	$(0, 1) \times (0, 1)$	81×81	0.003209	0.015846	3	1
4	$(0, 2) \times (0, 2)$	161×161	0.002886	0.017710	3	12
16	$(0, 4) \times (0, 4)$	321×321	0.002876	0.014799	3	24
64	$(0, 8) \times (0, 8)$	641×641	0.002844	0.014890	3	43
H^1 formulation, $\beta = 10^{-5}$, $\delta = 0\%$						
1	$(0, 1) \times (0, 1)$	81×81	0.000066	0.000263	4	1
4	$(0, 2) \times (0, 2)$	161×161	0.000064	0.000260	5	30
16	$(0, 4) \times (0, 4)$	321×321	0.000065	0.000263	5	58
64	$(0, 8) \times (0, 8)$	641×641	0.000071	0.000278	5	123
H^1 formulation, $\beta = 10^{-4}$, $\delta = 1\%$						
1	$(0, 1) \times (0, 1)$	81×81	0.000085	0.001706	4	1
4	$(0, 2) \times (0, 2)$	161×161	0.000067	0.001624	4	18
16	$(0, 4) \times (0, 4)$	321×321	0.000078	0.001569	4	42
64	$(0, 8) \times (0, 8)$	641×641	0.000091	0.001550	4	76
H^1 formulation, $\beta = 10^{-3}$, $\delta = 10\%$						
1	$(0, 1) \times (0, 1)$	81×81	0.001894	0.006133	3	1
4	$(0, 2) \times (0, 2)$	161×161	0.001752	0.005660	4	14
16	$(0, 4) \times (0, 4)$	321×321	0.001907	0.005773	4	26
64	$(0, 8) \times (0, 8)$	641×641	0.001933	0.004450	4	41*

Table 5. Test 3 with ovlp $= 16$. *Bottom line: $\beta = 10^{-2}$.

<table>
<tr><th>np</th><th>domain</th><th>mesh</th><th>error$_u$</th><th>error$_\rho$</th><th>Newton</th><th>GMRES</th></tr>
<tr><td colspan="7">L^2 formulation, $\beta = 10^{-6}$, $\delta = 0\%$</td></tr>
<tr><td>1</td><td>$(0, 1) \times (0, 1)$</td><td>81×81</td><td>0.000879</td><td>0.142454</td><td>3</td><td>1</td></tr>
<tr><td>4</td><td>$(0, 2) \times (0, 2)$</td><td>161×161</td><td>0.000879</td><td>0.142054</td><td>3</td><td>7</td></tr>
<tr><td>16</td><td>$(0, 4) \times (0, 4)$</td><td>321×321</td><td>0.000863</td><td>0.142271</td><td>3</td><td>21</td></tr>
<tr><td>64</td><td>$(0, 8) \times (0, 8)$</td><td>641×641</td><td>0.000861</td><td>0.142213</td><td>3</td><td>64</td></tr>
<tr><td colspan="7">L^2 formulation, $\beta = 10^{-5}$, $\delta = 1\%$</td></tr>
<tr><td>1</td><td>$(0, 1) \times (0, 1)$</td><td>81×81</td><td>0.002342</td><td>0.175353</td><td>4</td><td>1</td></tr>
<tr><td>4</td><td>$(0, 2) \times (0, 2)$</td><td>161×161</td><td>0.002313</td><td>0.174540</td><td>4</td><td>7</td></tr>
<tr><td>16</td><td>$(0, 4) \times (0, 4)$</td><td>321×321</td><td>0.002280</td><td>0.174360</td><td>4</td><td>17</td></tr>
<tr><td>64</td><td>$(0, 8) \times (0, 8)$</td><td>641×641</td><td>0.002265</td><td>0.174064</td><td>4</td><td>50</td></tr>
<tr><td colspan="7">L^2 formulation, $\beta = 10^{-4}$, $\delta = 10\%$</td></tr>
<tr><td>1</td><td>$(0, 1) \times (0, 1)$</td><td>81×81</td><td>0.006600</td><td>0.220451</td><td>4</td><td>1</td></tr>
<tr><td>4</td><td>$(0, 2) \times (0, 2)$</td><td>161×161</td><td>0.006522</td><td>0.219846</td><td>4</td><td>8</td></tr>
<tr><td>16</td><td>$(0, 4) \times (0, 4)$</td><td>321×321</td><td>0.006148</td><td>0.217108</td><td>4</td><td>19</td></tr>
<tr><td>64</td><td>$(0, 8) \times (0, 8)$</td><td>641×641</td><td>0.005972</td><td>0.215940</td><td>4</td><td>42</td></tr>
<tr><td colspan="7">H^1 formulation, $\beta = 10^{-5}$, $\delta = 0\%$</td></tr>
<tr><td>1</td><td>$(0, 1) \times (0, 1)$</td><td>81×81</td><td>0.000093</td><td>0.078365</td><td>4</td><td>1</td></tr>
<tr><td>4</td><td>$(0, 2) \times (0, 2)$</td><td>161×161</td><td>0.000093</td><td>0.078148</td><td>6</td><td>29</td></tr>
<tr><td>16</td><td>$(0, 4) \times (0, 4)$</td><td>321×321</td><td>0.000093</td><td>0.078040</td><td>6</td><td>56</td></tr>
<tr><td>64</td><td>$(0, 8) \times (0, 8)$</td><td>641×641</td><td>0.000093</td><td>0.077985</td><td>6</td><td>95</td></tr>
<tr><td colspan="7">H^1 formulation, $\beta = 10^{-4}$, $\delta = 1\%$</td></tr>
<tr><td>1</td><td>$(0, 1) \times (0, 1)$</td><td>81×81</td><td>0.000301</td><td>0.106186</td><td>4</td><td>1</td></tr>
<tr><td>4</td><td>$(0, 2) \times (0, 2)$</td><td>161×161</td><td>0.000298</td><td>0.105925</td><td>4</td><td>29</td></tr>
<tr><td>16</td><td>$(0, 4) \times (0, 4)$</td><td>321×321</td><td>0.000299</td><td>0.105777</td><td>4</td><td>38</td></tr>
<tr><td>64</td><td>$(0, 8) \times (0, 8)$</td><td>641×641</td><td>0.000301</td><td>0.105717</td><td>4</td><td>64</td></tr>
<tr><td colspan="7">H^1 formulation, $\beta = 10^{-3}$, $\delta = 10\%$</td></tr>
<tr><td>1</td><td>$(0, 1) \times (0, 1)$</td><td>81×81</td><td>0.002268</td><td>0.143252</td><td>4</td><td>1</td></tr>
<tr><td>4</td><td>$(0, 2) \times (0, 2)$</td><td>161×161</td><td>0.002167</td><td>0.142860</td><td>4</td><td>15</td></tr>
<tr><td>16</td><td>$(0, 4) \times (0, 4)$</td><td>321×321</td><td>0.002293</td><td>0.142457</td><td>4</td><td>28</td></tr>
<tr><td>64</td><td>$(0, 8) \times (0, 8)$</td><td>641×641</td><td>0.002334</td><td>0.142601</td><td>5</td><td>55</td></tr>
</table>

Table 6. Test 4 with ovlp $= 16$.

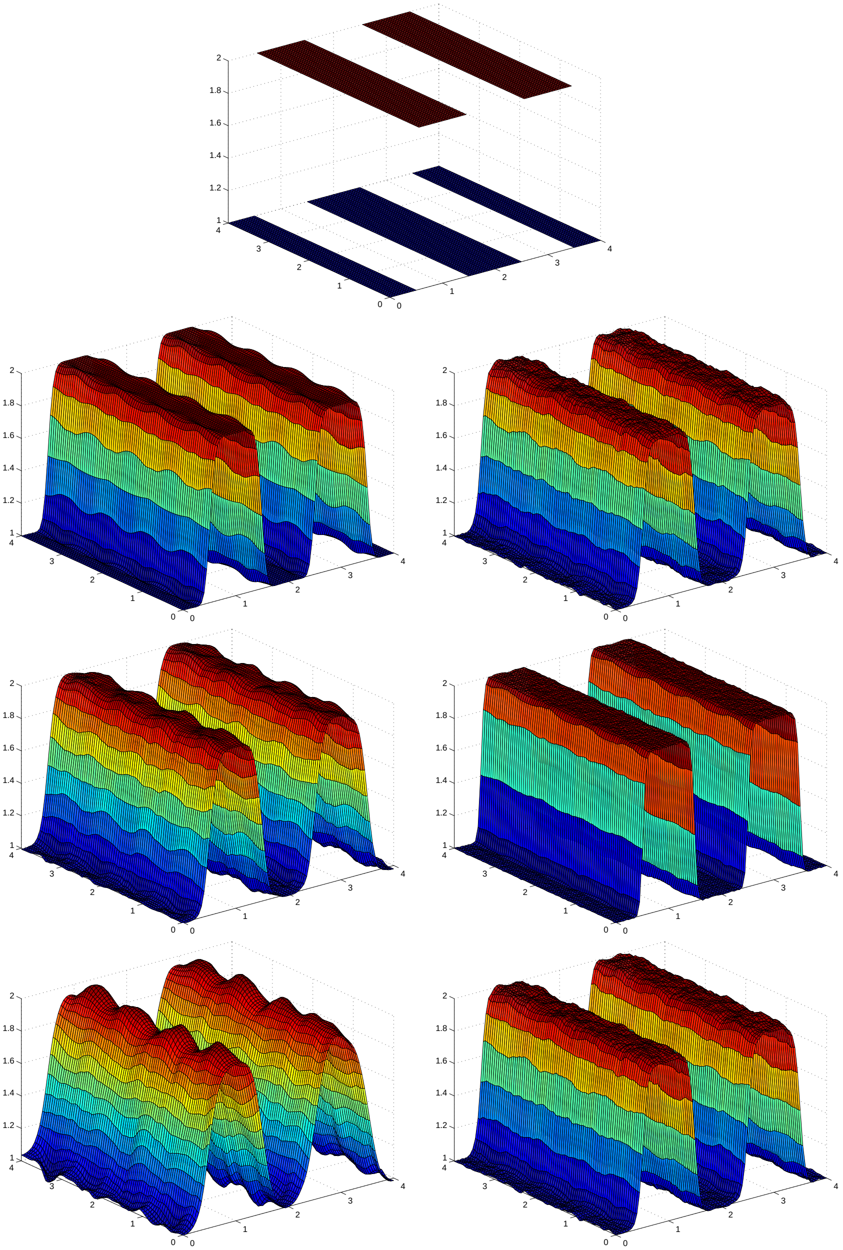


Figure 4. Test 4 on computational domain $(0,4)\times(0,4)$. Top: surface plot of the exact solution ρ. Bottom six pictures are numerical solutions with $\delta = 0\%$, $\delta = 1\%$ and $\delta = 10\%$. Left: L^2 formulation; right: H^1 formulation.

as in (4) and (5), then ρ automatically satisfies the zero Neumann boundary condition

$$\frac{\partial \rho}{\partial n} = 0.$$

To see this, we take the derivative of $\mathscr{L}$ in (4) with respect to ρ at any direction $p \in H^1(\Omega)$ to obtain

$$(\nabla_\rho \mathscr{L})p = \beta \int_\Omega \nabla \rho \cdot \nabla p \, dx + \int_\Omega p \nabla u \cdot \nabla \lambda \, dx = 0\,. \tag{A.1}$$

Applying the integration by parts to the first term in (A.1),

$$\int_\Omega \nabla \rho \cdot \nabla p \, dx = -\int_\Omega \Delta \rho \; p \, dx + \int_{\partial\Omega} \frac{\partial \rho}{\partial n} p \, ds\,,$$

we obtain

$$\int_\Omega (-\beta \Delta \rho + \nabla u \cdot \nabla \lambda) \; p \, dx + \beta \int_{\partial\Omega} \frac{\partial \rho}{\partial n} \; p \, ds = 0$$

for any p. This implies that, if $\beta \neq 0$,

$$\frac{\partial \rho}{\partial n} = 0$$

on $\partial\Omega$. The same result can be obtained for the H^1 formulation (4).

References

[1] V. Akcelik, G. Biros, A. Draganescu, O. Ghattas, J. Hilland, and B. van Bloeman Waanders, *Dynamic data-driven inversion for terascale simulations: real-time identification of airborne contaminants*, Proceedings of Supercomputing, 2005.

[2] V. Akçelik, G. Biros, O. Ghattas, K. Long, and B. van B. Waanders, *A variational finite element method for source inversion for convective-diffusive transport*, Finite Elem. Anal. Des. **39** (2003), no. 8, 683–705, 14th Robert J. Melosh Competition (Durhan, NC, 2002). MR 1985391

[3] U. M. Ascher and E. Haber, *A multigrid method for distributed parameter estimation problems*, Electron. Trans. Numer. Anal. **15** (2003), 1–17, Tenth Copper Mountain Conference on Multigrid Methods (Copper Mountain, CO, 2001). MR 2005a:65148 Zbl 1031.65108

[4] X.-C. Cai, W. D. Gropp, D. E. Keyes, R. G. Melvin, and D. P. Young, *Parallel Newton–Krylov–Schwarz algorithms for the transonic full potential equation*, SIAM J. Sci. Comput. **19** (1998), no. 1, 246–265, Special issue on iterative methods (Copper Mountain, CO, 1996). MR 99b:65138 Zbl 0917.76035

[5] X.-C. Cai and M. Sarkis, *A restricted additive Schwarz preconditioner for general sparse linear systems*, SIAM J. Sci. Comput. **21** (1999), no. 2, 792–797. MR 2000f:65133 Zbl 0944.65031

[6] T. F. Chan and X.-C. Tai, *Identification of discontinuous coefficients in elliptic problems using total variation regularization*, SIAM J. Sci. Comput. **25** (2003), no. 3, 881–904. MR2005c:65091 Zbl 1046.65090

[7] Z. Chen and J. Zou, *Finite element methods and their convergence for elliptic and parabolic interface problems*, Numer. Math. **79** (1998), no. 2, 175–202. MR 99d:65313 Zbl 0909.65085

[8] ______, *An augmented Lagrangian method for identifying discontinuous parameters in elliptic systems*, SIAM J. Control Optim. **37** (1999), no. 3, 892–910. MR 2000d:65203 Zbl 0940.65117

[9] J. E. Dennis, Jr. and R. B. Schnabel, *Numerical methods for unconstrained optimization and nonlinear equations*, Classics in Applied Mathematics, no. 16, Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 1996, Corrected reprint of the 1983 original. MR 96i:90002 Zbl 0847.65038

[10] S. C. Eisenstat and H. F. Walker, *Globally convergent inexact Newton methods*, SIAM J. Optim. **4** (1994), no. 2, 393–422. MR 95c:65078 Zbl 0814.65049

[11] H. W. Engl, M. Hanke, and A. Neubauer, *Regularization of inverse problems*, Mathematics and its Applications, no. 375, Kluwer Academic Publishers Group, Dordrecht, 1996. MR97k:65145 Zbl 0859.65054

[12] E. Haber and U. M. Ascher, *Preconditioned all-at-once methods for large, sparse parameter estimation problems*, Inverse Problems **17** (2001), no. 6, 1847–1864. MR 2002j:35308 Zbl 0995.65110

[13] K. Ito and K. Kunisch, *The augmented Lagrangian method for parameter estimation in elliptic systems*, SIAM J. Control Optim. **28** (1990), no. 1, 113–136. MR 91d:35228 Zbl 0709.93021

[14] Y. L. Keung and J. Zou, *An efficient linear solver for nonlinear parameter identification problems*, SIAM J. Sci. Comput. **22** (2000), no. 5, 1511–1526. MR 2001m:65156 Zbl 0978.35096

[15] A. Kirsch, *An introduction to the mathematical theory of inverse problems*, Applied Mathematical Sciences, no. 120, Springer, New York, 1996. MR 99c:34023 Zbl 0865.35004

[16] G. Kuruvila, S. Ta'asan, and M. D. Salas, *Airfoil optimization by the one-shot method*, AGARD Report 803, 1994.

[17] T. P. Mathew, M. Sarkis, and C. E. Schaerer, *Analysis of block matrix preconditioners for elliptic optimal control problems*, Numer. Linear Algebra Appl. **14** (2007), no. 4, 257–279. MR 2008d:65038

[18] E. E. Prudencio, R. Byrd, and X.-C. Cai, *Parallel full space SQP Lagrange–Newton–Krylov–Schwarz algorithms for PDE-constrained optimization problems*, SIAM J. Sci. Comput. **27** (2006), no. 4, 1305–1328. MR 2006k:49088 Zbl 1092.49024

[19] E. E. Prudencio and X.-C. Cai, *Parallel multilevel restricted Schwarz preconditioners with pollution removing for PDE-constrained optimization*, SIAM J. Sci. Comput. **29** (2007), no. 3, 964–985. MR 2008d:76033

[20] Y. Saad, *Iterative methods for sparse linear systems*, second ed., Society for Industrial and Applied Mathematics, Philadelphia, PA, 2003. MR 2004h:65002 Zbl 1031.65046

[21] A. Shenoy, M. Heinkenschloss, and E. M. Cliff, *Airfoil design by an all-at-once method*, Int. J. Comput. Fluid Dyn. **11** (1998), no. 1-2, 3–25, Flow control and optimization. MR 1682727 Zbl 0940.76084

[22] B. F. Smith, P. E. Bjørstad, and W. D. Gropp, *Domain decomposition: parallel multilevel methods for elliptic partial differential equations*, Cambridge University Press, Cambridge, 1996. MR 98g:65003 Zbl 0857.65126

[23] A. Toselli and O. Widlund, *Domain decomposition methods—algorithms and theory*, Springer Series in Computational Mathematics, no. 34, Springer, Berlin, 2005. MR 2005g:65006 Zbl 1069.65138

[24] C. R. Vogel, *Computational methods for inverse problems*, Frontiers in Applied Mathematics, no. 23, Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 2002, With a foreword by H. T. Banks. MR 2003i:65004 Zbl 1008.65103

Received March 24, 2008.

XIAO-CHUAN CAI: cai@cs.colorado.edu
Department of Computer Science, University of Colorado at Boulder, Boulder, CO 80309, United States

SI LIU: si.liu@colorado.edu
Department of Applied Mathematics, University of Colorado at Boulder, Boulder, CO 80309, United States

JUN ZOU: zou@math.cuhk.edu.hk
Department of Mathematics, The Chinese University of Hong Kong, Shatin, N. T., Hong Kong, China

COMM. APP. MATH. AND COMP. SCI.
Vol. 4, No. 1, 2009

COMMENTS ON HIGH-ORDER INTEGRATORS EMBEDDED WITHIN INTEGRAL DEFERRED CORRECTION METHODS

ANDREW CHRISTLIEB, BENJAMIN ONG AND JING-MEI QIU

A class of novel deferred correction methods, integral deferred correction (IDC) methods, is studied. This class of methods is an extension of ideas introduced by Dutt, Greengard and Rokhlin on spectral deferred correction (SDC) methods for solving ordinary differential equations (ODEs). The novel nature of this class of defect correction methods is that the correction of the defect is carried out using an accurate integral form of the residual instead of the more familiar differential form. As a family of methods, these schemes are capable of matching the efficiency of popular high-order RK methods.

The smoothness of the error vector associated with an IDC method is an important indicator of the order of convergence that can be expected from a scheme (Christlieb, Ong, and Qiu; Hansen and Strain; Skeel). It is demonstrated that embedding an r-th order integrator in the correction loop of an IDC method does not always result in an r-th order increase in accuracy. Examples include IDC methods constructed using non-self-starting multistep integrators, and IDC methods constructed using a nonuniform distribution of quadrature nodes.

Additionally, the integral deferred correction concept is reposed as a framework to generate high-order Runge–Kutta (RK) methods; specifically, we explain how the prediction and correction loops can be incorporated as stages of a high-order RK method. This alternate point of view allows us to utilize standard methods for quantifying the performance (efficiency, accuracy and stability) of integral deferred correction schemes. It is found that IDC schemes constructed using uniformly distributed nodes and high-order integrators are competitive in terms of efficiency with IDC schemes constructed using Gauss–Lobatto nodes and forward Euler integrators. With respect to regions of absolute stability, however, IDC methods constructed with uniformly distributed nodes and high-order integrators are far superior. It is observed that as the order of the embedded integrator increases, the stability region of the IDC method increases.

MSC2000: primary 65L05, 65L06, 65L20, 65L70; secondary 65B05.

Keywords: spectral deferred correction methods, integral deferred correction methods, Runge–Kutta methods, multistage methods, multistep methods, efficiency, stability, accuracy.

Research supported by Air Force Office of Scientific Research and Air Force Research Labs (Edward's and Kirtland). Grant nos. FA9550-07-1-0092 and FA9550-07-1-0144.

1. Introduction

In this paper, we construct and analyze a class of novel correction methods, integral deferred correction methods (IDC), which are constructed using high-order single-step and multistep integrators in the prediction and correction loops. IDC methods were first introduced in [4], and further developed and analyzed in [2; 5; 11; 18; 15]. Essentially, a deferred correction procedure is applied to an integral formulation of the error equation. This error equation is then solved by choosing a distribution of quadrature nodes and an integrator; these choices are crucial in determining the accuracy and stability of the scheme. For example, the selection of quadrature nodes is discussed in [15] and the selection of integrators for the prediction and correction loops are discussed in [16; 14; 18]. The authors in [11; 12] use Gaussian quadrature nodes and Krylov subspace methods to accelerate the convergence of the scheme. In [2], the advantages of using high-order RK integrators in SDC framework are shown analytically and numerically.

To study the properties of IDC schemes, the error arising from these schemes has to be analyzed. This error has two separate components: the first component is the error between the collocation solution on a given set of quadrature nodes and the exact solution [5; 11; 12]; this component limits the maximum achievable accuracy of IDC methods. The second component is the error that arises from using deferred correction iterations to approximate the collocation solution [2; 8; 9; 21]. In Section 3 of this paper, we focus on the second component of the abovementioned error. We will show that IDC methods constructed with p-th order multistage RK integrators (IDC-RK) and a uniform distribution of quadrature nodes give a p-th order increase in accuracy after each correction loop (under mild assumptions), whereas IDC-RK methods constructed with a nonuniform distribution of quadrature nodes do not give a p-th order increase ($p \geq 2$) after each correction loop. When multistep methods — for example, Adams–Bashforth (AB) methods — are used within an IDC method, the smoothness of the rescaled error vector prevents a high-order accuracy increase after each correction loop, regardless of the distribution of quadrature nodes.

In Section 4, we address a commonly perceived drawback of IDC methods: the additional computational overhead needed to implement these schemes. By formulating IDC-RK methods into an RK method, the local truncation error arising from IDC-RK schemes can be estimated. We show that a smaller truncation error offsets the computational overhead, making IDC methods constructed using uniformly distributed nodes and high-order integrators, as well as IDC methods constructed using Gaussian–Lobatto nodes and forward Euler integrators, competitive (in terms of efficiency) with known RK methods for eighth- and higher-order schemes. Additionally, the formulation of IDC-RK methods as an RK method gives a systematic way to generate arbitrary-order RK methods *without* solving complicated order conditions; an added bonus is that the entries of the RK Butcher tableau [6] can be computed exactly using a symbolic manipulator. Accuracy plots are generated in Section 4.3, validating the error estimates while stability plots in Section 4.4 show that IDC methods offer a much larger stability region compared with known RK methods. In fact, as the order of the embedded integrator is increased, the stability region of an IDC method also increases. These superior stability regions are one of the promising features of IDC-RK methods.

This paper is organized into three main sections. In Section 2, a brief review of IDC methods is given. In Section 3, properties of IDC methods constructed using general high-order integrators and various distributions of quadrature nodes are given, along with some examples. IDC methods are then reformulated into high-order RK methods in Section 4, and a detailed comparison between IDC methods and RK methods is given. Section 5 contains the conclusion and closing remarks.

2. Review of IDC methods

This section is a review of IDC methods from [4]. Our discussion of these methods is based on notation introduced below. We consider an IVP consisting of a system of ODEs and initial conditions,

$$\begin{cases} y'(t) = f(t, y), & t \in [0, T], \\ y(0) = y_0. \end{cases} \tag{2-1}$$

The time domain, $[0, T]$, is discretized into intervals,

$$0 = t_1 < t_2 < \cdots < t_n < \cdots < t_N = T,$$

and each interval, $I_n = [t_n, t_{n+1}]$, is further discretized into subintervals,

$$t_n = t_{n,0} = t_{n,1} < \cdots < t_{n,m} < \cdots < t_{n,M} = t_{n+1}. \tag{2-2}$$

We refer to $t_{n,m}$ as quadrature nodes, whose index m runs from 0 to M.

An IDC method on each time interval $[t_n, t_{n+1}]$, described below, is iterated completely to define the starting value for the next interval, $[t_{n+1}, t_{n+2}]$. We drop the subscript n, so that $t_{n,m} =: t_m$ in (2-2), with the understanding that the IDC method is described for that one time interval.

- (prediction step) Use an r_0-th order numerical method to obtain a numerical solution, $\vec{\eta}^{[0]} = (\eta_0^{[0]}, \eta_1^{[0]}, \ldots, \eta_m^{[0]}, \ldots, \eta_M^{[0]})$, which is an r_0-th order approximation to $\vec{y} = (y_0, y_1, \ldots, y_m, \ldots, y_M)$, where $y_m = y(t_m)$ is the exact solution at t_m.

- (correction loop) Use the error function to improve the accuracy of the scheme at each iteration.

  For $k = 1, \dots, k_l$ (k_l is number of correction steps)

  (1) Denote the *error function* from the $(k-1)$-st loop as

  $$e^{(k-1)}(t) = y(t) - \eta^{(k-1)}(t), \tag{2-3}$$

  where $y(t)$ is the exact solution and $\eta^{(k-1)}(t)$ is an M-th degree polynomial interpolating $\vec{\eta}^{[k-1]}$. Note that the error function, $e^{(k-1)}(t)$, is not a polynomial in general.

  (2) Compute the *residual function*, $\epsilon^{(k-1)}(t) = (\eta^{(k-1)})'(t) - f(t, \eta^{(k-1)}(t))$. In the literature, the residual function is often called the pointwise, or differential defect.

  (3) Compute the *numerical error vector*, $\vec{\delta}^{[k]} = (\delta_0^{[k]}, \dots, \delta_m^{[k]}, \dots, \delta_M^{[k]})$, using an r_k-th order numerical method to discretize the integral form of the *error equation*,

  $$\begin{aligned}\Big(e^{(k-1)} + \int_0^t \epsilon^{(k-1)}(\tau)\,d\tau\Big)'(t) &= f(t, \eta^{(k-1)}(t) + e^{(k-1)}(t)) - f(t, \eta^{(k-1)}(t)) \\ &\doteq F(t, e^{(k-1)}(t)),\end{aligned} \tag{2-4}$$

  where $F(t, e(t)) = f(t, \eta(t) + e(t)) - f(t, \eta(t))$, $\vec{\delta}^{[k]}$ is an r_k-th order approximation to

  $$\vec{e}^{[k-1]} = (e_0^{[k-1]}, \dots, e_m^{[k-1]}, \dots, e_M^{[k-1]}),$$

  and $e_m^{[k-1]} = e^{(k-1)}(t_m)$ is the value of the exact error function at t_m.

  (4) Update the numerical solution $\vec{\eta}^{[k]} = \vec{\eta}^{[k-1]} + \vec{\delta}^{[k]}$.

Notationally, superscripts with a round bracket, for example (k), denote a function, while superscripts with a square bracket, $[k]$, denote a vector at the k-th correction step. English letters are reserved for functions or vectors in the exact solution space, for example an exact solution $y(t)$ and an exact error function $e(t)$, while Greek letters denote functions or vectors in the numerical solution space, for example a numerical solution $\eta(t)$ and a numerical error function $\delta(t)$.

A forward Euler discretization of the error (2-4) gives

$$\delta_{m+1}^{[k]} = \delta_m^{[k]} + h_m(f(t_m, \eta_m^{[k-1]} + \delta_m^{[k]}) - f(t_m, \eta_m^{[k-1]})) - \int_{t_m}^{t_{m+1}} \epsilon^{(k-1)}(t)\,dt, \tag{2-5}$$

where $h_m = t_{m+1} - t_m$. Expanding the integral in (2-5),

$$\int_{t_m}^{t_{m+1}} \epsilon^{(k-1)}(t)\,dt = \big[\eta^{(k-1)}(t_{m+1}) - \eta^{(k-1)}(t_m)\big] - \int_{t_m}^{t_{m+1}} f(t, \eta^{(k-1)}(t))\,dt, \tag{2-6}$$

and substituting (2-6) into (2-5) results in

$$\eta_{m+1}^{[k]} = \eta_m^{[k]} + h_m\big[f(t_m, \eta_m^{[k-1]} + \delta_m^{[k]}) - f(t_m, \eta_m^{[k-1]})\big] + \int_{t_m}^{t_{m+1}} f(t, \eta^{[k-1]}(t))\,dt. \tag{2-7}$$

The integral in (2-7) can be evaluated using a Lagrange interpolant constructed from the function values, $\int_{t_m}^{t_{m+1}} L_{(\vec{t},\vec{f})}(\tau)\,d\tau$, where

$$L_{(\vec{t},\vec{f})}(\tau) = \sum_{m=0}^{M} \alpha_m(\tau) f_m, \qquad \text{with } \alpha_m(\tau) = \prod_{n\neq m} \frac{\tau - t_n}{t_m - t_n} \text{ and } f_m = f(t_m, \eta^{[k-1]}(t_m)). \tag{2-8}$$

Hence, (2-7) can also be written as

$$\eta_{m+1}^{[k]} = \eta_m^{[k]} + h_m\left[f(t_m, \eta_m^{[k-1]} + \delta_m^{[k]}) - f(t_m, \eta_m^{[k-1]})\right] + \sum_{j=0}^{M} S_{mj} f(t_j, \eta_j^{[k-1]}),$$

where

$$S_{mj} = \int_{t_m}^{t_{m+1}} \alpha_j(\tau)\,d\tau$$

are the elements of the so-called integration matrix.

IDC methods constructed using s-stage RK integrators (IDC-RKs) are more involved. We provide the following details for discretizing the error (2-4) for uniformly distributed quadrature nodes; generalization to nonuniformly distributed quadrature nodes is straightforward. Denoting by h the interval size for the uniformly distributed nodes and implementing an s-stage RK integrator to discretize (2-4) gives

$$k_1 = F(t_m, \delta_m^{[k-1]}), \tag{2-9a}$$

$$k_2 = F\left(t_m + c_2 h, \delta_m^{[k-1]} + h a_{2,1} k_1 - \int_{t_m}^{t_m + c_2 h} \epsilon^{(k-1)}(\tau)\,d\tau\right), \tag{2-9b}$$

$$k_3 = F\left(t_m + c_3 h, \delta_m^{[k-1]} + h(a_{3,1} k_1 + a_{3,2} k_2) - \int_{t_m}^{t_m + c_3 h} \epsilon^{(k-1)}(\tau)\,d\tau\right),$$

$$\vdots$$

$$k_s = F\left(t_m + c_s h, \delta_m^{[k-1]} + h \sum_{l=1}^{s-1} a_{s,l} k_l - \int_{t_m}^{t_m + c_s h} \epsilon^{(k-1)}(\tau)\,d\tau\right), \tag{2-9c}$$

$$\delta_{m+1}^{[k-1]} = \delta_m^{[k-1]} + h \sum_{l=1}^{s} b_l k_l - \int_{t_m}^{t_m + h} \epsilon^{(k-1)}(\tau)\,d\tau, \tag{2-9d}$$

where $A, \vec{b}, \vec{c}$ are conventional Butcher table entries [7] for an s-stage RK integrator:

$$\begin{array}{c|c} \vec{c} & A \\ \hline & \vec{b}^T \end{array}.$$

Each RK stage, for example (2-9b),

$$k_2 = f\Big(t_m + c_2h,\, \eta^{(k-1)}(t_m + c_2h) + \delta_m^{[k-1]} + ha_{2,1}k_1 - \int_{t_m}^{t_m+c_2h} \epsilon^{(k-1)}(\tau)\,d\tau\Big) \\ - f\big(t_m + c_2h,\, \eta^{(k-1)}(t_m + c_2h)\big), \quad (2\text{-}10)$$

involves the integral of the residual function,

$$\int_{t_m}^{t_m+c_2h} \epsilon^{(k-1)}(\tau)\,d\tau \\ = \big[\eta^{(k-1)}(t_m + c_2h) - \eta^{(k-1)}(t_m)\big] - \int_{t_m}^{t_m+c_2h} f(t, \eta^{(k-1)}(t))\,dt, \quad (2\text{-}11)$$

where the integral in (2-11)

$$\int_{t_m}^{t_m+c_2h} f(t, \eta^{(k-1)}(t))\,dt = \sum_{j=0}^{M} S_{m\cdot s+2,j}\, f(t_j, \eta_j^{[k-1]}),$$

can be evaluated using the integration matrix, $S_{m\cdot s+2,j} = \int_{t_m}^{t_m+c_2h} \alpha_j(t)\,dt$. For future reference, the general expression for the entries of this expanded integration matrix is

$$S_{m\cdot s+l,j} = \begin{cases} \int_{t_m}^{t_m+c_lh} \alpha_j(t)\,dt, & l = 2, \dots, s,\ m = 0, \dots, M-1,\ j = 0, \dots, M, \\ \int_{t_m}^{t_{m+1}} \alpha_j(t)\,dt, & l = 1,\ m = 0, \dots, M-1,\ j = 0, \dots, M. \end{cases} \quad (2\text{-}12)$$

We choose this expanded definition of the integration matrix so that IDC methods constructed with single step integrators can be formulated as a high-order RK method in Section 4.1. Using this definition of the integration matrix, (2-10) can be expressed as

$$k_2 = f\Big(t_m, +c_2h,\, \eta_m^{[k]} + ha_{2,1}k_1 - \sum_{j=0}^{M} S_{m\cdot s+2,j}\, f(t_j, \eta_j^{[k-1]})\Big) \\ - f\big(t_m + c_2h,\, \eta^{(k-1)}(t_m + c_2h)\big). \quad (2\text{-}13)$$

The term $f\big(t_m + c_2h, \eta^{(k-1)}(t_m + c_2h)\big)$ in (2-13) can be computed by evaluating the Lagrangian interpolant at the intermediate stage, $L_{(\vec{t},\vec{f})}(t_m + c_2h)$. This can also be written as

$$L_{(\vec{t},\vec{f})}(t_m + c_2 h) = \sum_{j=0}^{M} L_{m\cdot s+2,j}\, f(t_j, \eta_j^{[k-1]}),$$

where the entries of the interpolation matrix are given by

$$L_{m\cdot s+l,j} = \alpha_j(t_m + c_l h), \quad l = 1, \ldots, s, \quad m = 0, \ldots, M-1. \tag{2-14}$$

The remaining stages and their combinations are evaluated in a similar fashion. In Section 4.1, we systematically formulate IDC methods constructed using RK integrators as high-order RK methods.

We omit details for constructing IDC methods using multistep integrators (such as IDC-AB) because such schemes are not self-starting. We show in Section 3 that the obvious approach of using a high-order RK integrator to compute the first few steps results in an error vector which lacks sufficient smoothness; this lack of smoothness results in a poorer than desired accuracy increase after each correction loop.

3. IDC methods constructed using high-order integrators

In this section, we discuss the accuracy of IDC methods constructed using high-order integrators and various distributions of quadrature nodes. Specifically, IDC methods constructed using multistage RK methods are discussed in Section 3.2 for uniformly spaced quadrature nodes, and Section 3.3 for a nonuniform distribution. IDC methods constructed using high-order multistep methods are given in Section 3.4. The smoothness of the rescaled error vector measured in a discrete Sobolev norm is a crucial tool for both discussions; we review this concept in Section 3.1.

3.1. ***Mathematical preliminaries.*** Several analytical and numerical preliminaries are needed to analyze IDC methods. The smoothness of discrete data sets will be established, analog to the smoothness of functions; this idea of smoothness is used to analyze the error vectors. Let $f(t)$ be a function for $t \in [0, T]$, and denote the corresponding discrete data set,

$$(\vec{t}, \vec{f}) = \{(t_0, f_0), \ldots, (t_M, f_M)\}, \tag{3-1}$$

where

$$0 = t_0 < t_1 < t_2 < \ldots < t_M = H. \tag{3-2}$$

Definition 3.1 (Smoothness of a function). A function $f(t)$, $t \in [0, T]$, possesses S degrees of smoothness if $\|d^s f\|_\infty := \|\partial^s f / \partial t^s\|_\infty$ is bounded for $s = 0, 1, 2, \ldots, S$, where $\|f\|_\infty := \max_{t\in[0,T]} |f(t)|$.

Definition 3.2 (s-th degree spectral differentiation). Consider the discrete data set, $(\vec{t}, \vec{f})$, defined in (3-1), and let $L_{(\vec{t},\vec{f})}(\tau)$ be the Lagrange interpolant described in (2-8). An s-th degree spectral differentiation is a linear mapping that maps $\vec{f}$ into

$$\overrightarrow{\hat{d}_s f}, \qquad \text{where } (\hat{d}_s f)_m = (\partial^s/\partial\tau^s)L_{(\vec{t},\vec{f})}(\tau)|_{\tau=t_m}.$$

This linear mapping can be represented by

$$\overrightarrow{\hat{d}_s f} = \hat{D}_s \cdot \vec{f},$$

where $\hat{D}_s \in \Re^{(M+1)\times(M+1)}$ and $(\hat{D}_s)_{mn} = (\partial^s/\partial\tau^s)c_n(\tau)|_{\tau=t_m}$, $m, n = 0, \dots, M$.

Remark 3.3. Given a distribution of quadrature nodes on [0, 1], the spectral differentiation matrices, $\hat{D}_s^{[0,1]}$, $s = 1, \dots, M$, have constant entries. If this distribution of quadrature nodes is rescaled from [0, 1] to [0, H], then the corresponding differentiation matrices are

$$\hat{D}_1 = \frac{1}{H}\hat{D}_1^{[0,1]} \quad \text{and} \quad \hat{D}_s = \left(\frac{1}{H}\right)^s \hat{D}_s^{[0,1]}.$$

Definition 3.4. The $(\hat{S}, \infty)$ Sobolev norm of a discrete data set $(\vec{t}, \vec{f})$ is defined as

$$\|\vec{f}\|_{\hat{S},\infty} := \|\vec{f}\|_\infty + \sum_{s=1}^{S} \|\overrightarrow{\hat{d}_s f}\|_\infty = \|\vec{f}\|_\infty + \sum_{s=1}^{S} \|\hat{D}_s \cdot \vec{f}\|_\infty.$$

Definition 3.5 (Smoothness of a discrete date set). A discrete data set, (3-1), possesses $S \le M$ degrees of smoothness if $\|\vec{f}\|_{\hat{S},\infty}$ is bounded as $H \to 0$.

Remark 3.6. We emphasize that smoothness is a property of discrete data sets in the limit as $H \to 0$. We also impose $S \le M$, because

$$\overrightarrow{\hat{d}_S f} \equiv \vec{0},$$

for $S > M$. See [2] for a detailed discussion.

Example 3.7 (A discrete data set with only one degree of smoothness). Consider the discrete data set

$$(\vec{t}, \vec{f}) = \left\{(0,0), \left(\frac{H}{4}, \frac{H}{4}\right), \left(\frac{H}{2}, \frac{H}{2}\right), \left(\frac{3H}{4}, \frac{H}{4}\right), (H, 0)\right\}.$$

The first derivative

$$\overrightarrow{\hat{d}_1 f} = \left(-\frac{4}{3}, \frac{10}{3}, 0, -\frac{10}{3}, \frac{4}{3}\right),$$

is bounded independent of H, while the second derivative

$$\overrightarrow{\hat{d}_2 f} = \left(\frac{272}{3H}, -\frac{16}{3H}, -\frac{112}{3H}, -\frac{16}{3H}, \frac{272}{3H}\right),$$

is unbounded as $H \to 0$. Therefore, $(\vec{t}, \vec{f})$ has one and only one degree of smoothness in the discrete sense.

3.2. ***IDC methods constructed using RK integrators and uniformly spaced quadrature nodes.*** Integral deferred correction methods constructed using high-order RK integrators (IDC-RK) and uniformly spaced quadrature nodes boast superior accuracy and stability regions [2]. We restate the following theorem and lemmas from [2], which prove (under mild conditions) the accuracy of these IDC-RK methods. An example is provided to illustrate the main components of the theorem.

Theorem 3.8. *Let $y(t)$, the solution to the IVP (2-1), have at least $S \geq M+2$ degrees of smoothness in the continuous sense. Then, the local error for an IDC method constructed using $(M+1)$ uniformly distributed nodes, $(t_m = mh, m = 0, \dots, M)$, an (r_0)-th order RK method in the prediction step and $(r_1, r_2, \dots, r_{k_l})$-th order RK methods, is $\mathcal{O}(h^{(s_{k_l}+1)})$, where $s_{k_l} = \sum_{j=0}^{k_l} r_j \leq M+1$.*

The proof of Theorem 3.8 follows from the two lemmas below. Lemma 3.9 addresses the case $k = 0$, and Lemma 3.10 addresses the inductive argument. We emphasize that both lemmas not only bound the error vectors, but also guarantee sufficient smoothness in the prediction and correction steps.

Lemma 3.9. *(prediction step) Let $\vec{\eta}^{[0]} = (\eta_0^{[0]}, \dots, \eta_m^{[0]}, \dots, \eta_M^{[0]})$ be the numerical solution obtained after the prediction step. Then, the error vector $\vec{e}^{[0]} = \vec{y} - \vec{\eta}^{[0]}$ satisfies*

$$\|\vec{e}^{[0]}\|_\infty \sim \mathcal{O}(h^{r_0+1}),$$

and the rescaled error vector $\vec{\tilde{e}}^{[0]} = \frac{1}{h^{r_0}} \vec{e}^{[0]}$ has $\min(S - r_0, M)$ degrees of smoothness in the discrete sense.

Proof. We provide the following outline for a proof when a forward Euler integrator is used in the prediction step. Details for the more general case of using an RK integrator is provided in [2].

We drop the superscript [0] as there is no ambiguity. Since $\eta_{m+1} = \eta_m + hf(t_m, \eta_m)$, the error at t_{m+1}, $e_{m+1} = y_{m+1} - \eta_{m+1}$ satisfies

$$e_{m+1} = e_m + h(f(t_m, y_m) - f(t_m, \eta_m)) + \sum_{i=2}^{S-1} \frac{h^i}{i!} y^{(i)}(t_m) + \mathcal{O}(h^S),$$

where we have performed a Taylor expansion of y_{m+1} about $t = t_m$. Let $u_m = f(t_m, y_m) - f(t_m, \eta_m)$, and

$$r_m = \frac{h^2}{2!} y^{(2)}(t_m) + \cdots + \frac{h^{S-1}}{(S-1)!} y^{(S-1)}(t_m).$$

Notice that

$$u_m = e_m f_y(t_m, y_m) + \cdots + \frac{(-1)^{S-1}(e_m)^{S-2}}{(S-2)!} f_{y^{S-2}}(t_m, y_m) + \mathbb{O}((e_m)^{S-1}),$$

where we have performed a Taylor expansion of $f(t, \eta_m)$ about $y = y_m$. We are now ready to bound $\|\vec{e}^{[0]}\|_\infty$ by induction. By definition, $e_0 = 0$, so certainly, $e_0 \sim \mathbb{O}(h^2)$. Assume that $e_m \sim \mathbb{O}(h^2)$. Since $u_m \sim \mathbb{O}(e_m) \sim \mathbb{O}(h^2)$, we have

$$e_{m+1} = e_m + hu_m + r_m + \mathbb{O}(h^S) \sim \mathbb{O}(h^2),$$

which completes the inductive proof that $\|\vec{e}\|_\infty \sim \mathbb{O}(h^2)$. Note that the inductive proof was with respect to m, the index of the grid points.

To prove the smoothness of the rescaled error vector, we will again use an inductive approach, but this time with respect to s, the degree of smoothness. First, note that a discrete differentiation of the rescaled error vector gives

$$(d_1\tilde{e})_m = \frac{\tilde{e}_{m+1} - \tilde{e}_m}{h} = \tilde{u}_m + \frac{r_m}{h^2} + \mathbb{O}(h^{S-2}), \tag{3-3}$$

where

$$\tilde{u}_m = \frac{u_m}{h} = \sum_{i=1}^{S-2} (-1)^{i+1} \frac{h^{i-1}}{i!} f_{y^i}(t_m, y_m)(\tilde{e}_m)^i + \mathbb{O}(h^{2S-3}).$$

We are now ready to prove that $\vec{\tilde{e}}$ has M degrees of smoothness by induction. Since $\|\vec{\tilde{e}}\|_\infty \sim \mathbb{O}(h)$, $\vec{\tilde{e}}$ has at least zero degrees of smoothness in the discrete sense. Assume that $\vec{\tilde{e}}$ has $s \le M - 1$ degrees of smoothness. We will show that $\overrightarrow{d_1\tilde{e}}$ has s degrees of smoothness, from which we can conclude that $\vec{\tilde{e}}$ has $(s + 1)$ degrees of smoothness.

Since f_{y^i} has $(S - i - 1)$ degrees of smoothness in the continuous sense,

$$\overrightarrow{f_{y^i}} = [f_{y^i}(t_0, y_0), \ldots, f_{y^i}(t_M, y_M)]$$

has $(S - i - 1)$ degrees of smoothness in the discrete sense. Consequently, $h^{i-1}\overrightarrow{f_{y^i}}$ has $(S - 2)$ degrees of smoothness, which implies that $\vec{\tilde{u}}$ has min $(S - 2, s)$ degrees of smoothness. Similarly, $\vec{r}/h^2$ has $(S - 2)$ degrees of smoothness in the discrete sense. Hence $\overrightarrow{d_1\tilde{e}}$ has s degrees of smoothness $\Longrightarrow$ $\vec{\tilde{e}}$ has $(s + 1)$ degrees of smoothness. Since this argument holds for $S \ge M + 2$, we can conclude that $\vec{\tilde{e}}$ has M degrees of smoothness. □

Lemma 3.10 (Correction step). *Suppose after the* $(k - 1)$*-st correction loop the error vector satisfies* $\vec{e}^{[k-1]} \sim \mathbb{O}(h^{s_{k-1}+1})$ *and the rescaled error vector*

$$\vec{\tilde{e}}^{[k-1]} = \frac{1}{h^{s_{k-1}}} \vec{e}^{[k-1]}$$

has $M+1-s_{k-1}$ degrees of smoothness in the discrete sense. Then, after the k-th ($k < k_l$) correction loop the updated error vector satisfies

$$\|\vec{e}^{[k]}\|_\infty \sim \mathbb{O}(h^{s_k+1}),$$

and the rescaled error vector

$$\vec{\tilde{e}}^{[k]} = \frac{1}{h^{s_k}}\vec{e}^{[k]}$$

has $M+1-s_k$ degrees of smoothness in the discrete sense.

The proof is similar in spirit to the proof of Lemma 3.9 and is omitted for brevity.

Example 3.11. Consider the IVP

$$y'(t) = y(t); \quad y(0) = 1. \tag{3-4}$$

We solve IVP (3-4) with an IDC method constructed using six uniformly spaced quadrature nodes and the second-order trapezoidal RK method in the prediction and correction loops. Let H be the interval size and $h = \frac{H}{5}$ be the subinterval size. Computing the Taylor expansion of the numerical solution about $t = 0$ with $\mathbb{O}(h^7)$ truncation error, the rescaled error vectors are

$$\begin{aligned}\vec{\tilde{e}}^{[0]} = \Big\{0,\ &\frac{h}{750}+\frac{h^2}{15{,}000}+\frac{h^3}{375{,}000}+\frac{h^4}{11{,}250{,}000},\\ &\frac{h}{375}+\frac{h^2}{1{,}500}+\frac{4h^3}{46{,}875}+\frac{4h^4}{703{,}125},\ \frac{h}{250}+\frac{9h^2}{5{,}000}+\frac{51h^3}{125{,}000}+\frac{71h^4}{1{,}250{,}000},\\ &\frac{2h}{375}+\frac{13h^2}{3{,}750}+\frac{53h^3}{46{,}875}+\frac{166h^4}{703{,}125},\ \frac{h}{150}+\frac{17h^2}{3000}+\frac{181h^3}{75{,}000}+\frac{301h^4}{450{,}000}\Big\}+\mathbb{O}(h^5),\end{aligned}$$

$$\begin{aligned}\vec{\tilde{e}}^{[1]} = \Big\{0,\ &\frac{h}{225{,}000}-\frac{h^2}{4{,}500{,}000},\ \frac{h}{112{,}500}+\frac{7h^2}{2{,}250{,}000},\\ &\frac{h}{75{,}000}+\frac{h^2}{100{,}000},\frac{h}{56{,}250}+\frac{23h^2}{1{,}125{,}000},\frac{h}{45{,}000}+\frac{86111h^2}{250{,}000}\Big\}+\mathbb{O}(h^3),\end{aligned}$$

$$\vec{\tilde{e}}^{[2]} = \mathbb{O}(h).$$

As postulated by the lemmas, the rescaled error vector $\vec{\tilde{e}}^{[0]}$ has five degrees of smoothness, $\vec{\tilde{e}}^{[1]}$ has three degrees of smoothness, and $\vec{\tilde{e}}^{[3]}$ has one degree of smoothness. Table 1 gives the error and order of the implemented IDC method after the prediction step and each correction loop. As expected, second-order convergence is observed after the prediction loop, fourth-order convergence is observed after one correction loop, and sixth-order convergence is observed after the second correction loop.

	1 loop of RK2		2 loops of RK2		3 loops of RK2	
steps	error	order	error	order	error	order
5	7.03E-4	–	1.06E-7	–	5.91E-11	–
10	1.79E-4	1.98	6.36E-9	4.07	9.55E-13	5.95
15	7.97E-5	1.99	1.24E-9	4.04	8.26E-14	6.04
20	4.50E-5	1.99	3.88E-10	4.03	1.20E-14	6.71
25	2.88E-5	1.99	1.59E-10	4.02	4.44E-16	14.77

Table 1. Example 3.11: IDC6-RK2, the sixth-order IDC method constructed using six uniformly distributed quadrature nodes and the trapezoidal RK2, is used to solve IVP (3-4). The error at $T = 1$ is measured after the prediction loop (1 loop of RK2), first correction loop (2 loops of RK2) and second correction loop (3 loops of RK2). The corresponding order of convergence is calculated.

3.3. ***IDC Methods constructed using RK integrators and nonuniform distributions of quadrature nodes.*** One might consider constructing an IDC method using a nonuniform distribution of quadrature nodes, such as Gaussian–Lobatto [1] (or Gaussian–Radau or Gaussian) nodes, because their collocation solution can achieve $2M$ ($(2M+1)$ or $(2M+2)$) orders of accuracy with $M+1$ nodes. Consequently, one would expect that the computational effort for an IDC scheme constructed with Gaussian–Lobatto points should be a fraction of that for an equivalent scheme using a uniform distribution of nodes.

However, when high-order integrators are applied to the error equation, the lack of smoothness of the rescaled error vector associated with such IDC methods destroys the high-order accuracy increase. Consequently, there is little advantage to constructing IDC-RK integrators using a nonuniform distribution of quadrature nodes. However, when considering IDC method with low-order integrators, such as forward/backward Euler, nonuniform quadrature points, such as Gaussian points, might be advantageous because of the reduced sensitivity to interpolation error, and a better conditioned interpolation/integration matrix. This is best illustrated by the following examples. In Example 3.12, we consider IDC methods constructed using the trapezoidal RK2 method and quadrature nodes with linearly increasing interval sizes. In Example 3.13, we consider an IDC method constructed using Gaussian–Lobatto quadrature nodes and the trapezoidal RK2 method.

Example 3.12 (Linearly increasing interval sizes). We solve IVP (3-4) numerically with an IDC method constructed using six quadrature nodes distributed smoothly, though nonuniformly, with each interval satisfying

$$t_m - t_{m-1} = mh, \quad h = \frac{2H}{(M+1)M}, \quad m = 1, \dots, M, \tag{3-5}$$

where H is the interval size. The trapezoidal RK2 integrator is applied in the prediction and correction loops. Computing the Taylor expansion of the numerical solution about $t = 0$ with $\mathbb{O}(h^7)$ truncation error, the rescaled error vector after the prediction step satisfies

$$\begin{aligned}\vec{e}^{[0]} = \Big\{0,\ & \frac{h}{20{,}250} + \frac{h^2}{1,215{,}000} + \frac{h^3}{91{,}125{,}000} + \frac{h^4}{8{,}201{,}250{,}000}, \\ & \frac{h}{2{,}250} + \frac{19h^2}{405{,}000} + \frac{h^3}{375{,}000} + \frac{h^4}{11{,}250{,}000}, \\ & \frac{2h}{1{,}125} + \frac{19h^2}{40{,}500} + \frac{161h^3}{2{,}531{,}250} + \frac{67h^4}{12{,}656{,}250}, \\ & \frac{2h}{405} + \frac{1469h^2}{607{,}500} + \frac{547h^3}{911{,}250} + \frac{31h^4}{328{,}050}, \\ & \frac{h}{90} + \frac{3521h^2}{405{,}000} + \frac{463h^3}{135{,}000} + \frac{707h^4}{810{,}000}\Big\} + \mathbb{O}(h^5).\end{aligned}$$

It can be checked by Definition 3.5 that $\vec{e}^{[0]}$ has one and only one degree of smoothness. Since $\vec{e}^{[0]}$ has only one degree of smoothness in the discrete sense, only one order increase in accuracy is guaranteed after the first correction loop, even when a high-order RK method is applied. By computing the rescaled error vector after subsequent correction loops, one can show that only one order increase in accuracy per loop can be guaranteed until the maximum order is achieved.

This is illustrated in Table 2, which gives the error and order of the IDC method using the quadrature nodes distributed according to (3-5). Second-order accuracy is

	RK2 pred.		1 corr. loop		2 corr. loops		3 corr. loops	
steps	error	order	error	order	error	order	error	order
5	1.16E-3	–	2.16E-6	–	2.84E-9	–	2.3 E-10	–
10	2.96E-4	1.97	3.03E-7	2.83	2.77E-10	3.36	4.02E-12	5.86
15	1.32E-4	1.98	9.29E-8	2.91	6.12E-11	3.73	3.75E-13	5.85
20	7.47E-5	1.99	3.99E-8	2.94	2.04E-11	3.82	7.01E-14	5.83
25	4.79E-5	1.99	2.06E-8	2.95	8.58E-12	3.87	1.82E-14	6.05

Table 2. Example 3.12: The error at $T = 1$ and the order of the implemented IDC-RK2 method are tabulated after the prediction loop, first correction loop, second correction loop, etc. The quadrature nodes are distributed as described in (3-5).

observed after the RK2 prediction loop. Then, only third- and fourth-order accuracy are observed after the first and second RK2 correction loop as per the discussion above. Sixth-order accuracy is observed after the third RK2 correction loop.

Example 3.13. (Gaussian–Lobatto quadrature nodes) IVP (3-4) is solved numerically with an IDC method constructed using six Gaussian–Lobatto quadrature nodes given by

$$\begin{aligned} t_0 &= 0, & t_3 &= \Big(1+\sqrt{\tfrac{1}{21}(7-2\sqrt{7})}\Big)\frac{H}{2}, \\ t_1 &= \Big(1-\sqrt{\tfrac{1}{21}(7+2\sqrt{7})}\Big)\frac{H}{2}, & t_4 &= \Big(1+\sqrt{\tfrac{1}{21}(7+2\sqrt{7})}\Big)\frac{H}{2}, \\ t_2 &= \Big(1-\sqrt{\tfrac{1}{21}(7-2\sqrt{7})}\Big)\frac{H}{2}, & t_5 &= H, \end{aligned} \tag{3-6}$$

where H is the interval size. RK2 is applied in the prediction and correction loops. Computing the Taylor expansion of the numerical solution about $t=0$ with $\mathbb{O}(h^7)$ truncation error, the rescaled error vector after the prediction step satisfies

$$\begin{aligned} \vec{e}^{[0]} = \{0,\ & 0.00027018h + 0.00000793h^2 + 0.00000019h^3, \\ & 0.00257165h + 0.00048115h^2 + 0.00004858h^3 + 0.00000289h^4, \\ & 0.00643924h + 0.00287267h^2 + 0.00065172h^3 + 0.00008973h^4 \\ & 0.00874071h + 0.00603452h^2 + 0.00209675h^3 + 0.00046357h^4, \\ & 0.00901089h + 0.00730769h^2 + 0.00297836h^3 + 0.00078573h^4\} + \mathbb{O}(h^5). \end{aligned}$$

It can be checked by Definition 3.5 that $\vec{e}^{[0]}$ has one and only one degree of smoothness. Since $\vec{e}^{[0]}$ has only one degree of smoothness in the discrete sense, one order increase in accuracy is guaranteed after the first correction loop. Computing the rescaled error vectors after subsequent correction loops, one can show that the smoothness constraint guarantees only one order accuracy increase per loop until the maximum order is increased.

In Table 3, we show the error and rate of convergence of up to nine loops of RK2, to demonstrate that the maximum $2M$ order can be achieved when using Gauss–Lobatto points. The expected second/fourth/sixth-order convergence is observed after one/three/five loops of RK2 steps, respectively. However, fourth/sixth-order accuracy is observed after two/four RK2 loops. This discrepancy can be explained by carefully studying the error vector after the first and third RK2 correction loops.

$$\begin{aligned} \vec{e}^{[1]} = \{0,\ & -0.00000716h^4 + \mathbb{O}(h^5),\ -0.00004306h^4 + \mathbb{O}(h^5), \\ & -0.00004306h^4 + \mathbb{O}(h^5),\ -0.00000716h^4 + \mathbb{O}(h^5),\ 0.00004950h^5 + \mathbb{O}(h^6)\}. \end{aligned}$$

	RK-2 prediction		1 RK-2 corr. loop		2 RK-2 corr. loop	
steps	error	order	error	order	error	order
5	8.28E-3	–	2.13E-5	–	1.23E-6	–
10	2.19E-3	1.92	1.43E-6	3.90	8.51E-8	3.85
15	9.92E-4	1.95	2.89E-7	3.94	1.73E-8	3.93
20	5.63E-4	1.98	9.25E-8	3.96	5.55E-9	3.95
25	3.63E-4	1.97	3.82E-8	3.97	2.29E-9	3.97

	3 RK-2 corr. loops		4 RK-2 corr. loops	
steps	error	order	error	order
5	1.42E-8	–	2.25E-9	–
10	2.49E-10	5.84	3.86E-11	5.87
15	2.27E-11	5.91	3.48E-12	5.93
20	4.11E-12	5.94	6.27E-13	5.96
25	1.09E-12	5.95	1.64E-13	6.01

	6 loops of RK-2		7 loops of RK-2		8 loops of RK-2	
steps	error	order	error	order	error	order
3	8.89E-7	–	4.27E-8	–	1.59E-9	–
6	5.20E-9	7.41	3.70E-11	10.17	4.61E-12	8.43
9	2.29E-10	7.71	2.14E-13	12.70	1.03E-13	9.38
12	2.42E-11	7.81	9.59E-14	2.79	8.88E-15	8.52

Table 3. Example 3.13: The error and order of an IDC method used for solving IVP (3-4) using Gaussian–Lobatto quadrature nodes are tabulated. The error is computed at $T = 1$ after the RK-2 prediction loop, first RK-2 correction loop, second RK-2 correction loop, etc. Note that constructing an IDC method with six Gaussian–Lobatto points allows for up to tenth-order accuracy. Coarse steps are taken for the IDC method constructed with five correction loops or more because of machine precision limitations.

$\mathcal{O}(h^5)$ is observed in the last element of $\vec{e}^{[1]}$, corresponding to the results in the second column of Table 3; third-order, not fourth, is actually consistently achieved at the interior nodes after the first correction loop. After the second correction loop, fourth-order convergence is consistently achieved everywhere. Similarly, the error vector after the third correction loop,

$$\vec{e}^{[3]} = \{0, -0.000000004h^6 + \mathcal{O}(h^7), 0.00000003h^6 + \mathcal{O}(h^7),$$
$$0.00000003h^6 + \mathcal{O}(h^7), -0.000000004h^6 + \mathcal{O}(h^7), -0.00000002h^7 + \mathcal{O}(h^8)\},$$

is not consistently sixth-order at the interior nodes either.

3.4. ***IDC methods constructed using high-order multistep methods.*** A general linear p-step multistep method for solving IVP (2-1) is of the form

$$\begin{aligned}y_{n+p} + a_{p-1}y_{n+p-1} + a_{p-2}y_{n+p-2} + \cdots + a_0 y_n\\ = h(b_p f(t_{n+p}, y_{n+p}) + b_{p-1} f(t_{n+p-1}, y_{n+p-1}) + \cdots + b_0 f(t_n, y_n)).\end{aligned}$$

Examples of popular multistep methods include the explicit Adams–Bashforth methods (AB), implicit Adams–Moulton methods (AM), and backward differential formulas (BDF). For example,

$$y_{n+1} = y_n + h\big(\tfrac{3}{2} f(t_n, y_n) - \tfrac{1}{2} f(t_{n-1}, y_{n-1})\big), \tag{AB}$$

$$y_{n+1} = y_n + h\big(\tfrac{1}{2} f(t_n, y_n) + \tfrac{1}{2} f(t_{n+1}, y_{n+1})\big), \tag{AM}$$

$$y_{n+1} = \tfrac{4}{3} y_n - \tfrac{1}{3} y_{n-1} + \tfrac{2}{3} h f(t_{n+1}, y_{n+1}). \tag{BDF}$$

Most multistep methods are not self-starting; typically, a high-order integrator, such as an RK integrator, is used to compute the first few steps. In the next example, we show that using an RK-2 integrator as a starter ruins the desired accuracy increase which is possible with a high-order multistep method. Similar comments are also made in [14], in which a variable starting technique is suggested in conjunction with the multistep methods. Although this technique showed some promise in test examples (see [14, Figure 2]), we did not observe high-order increase in the correction loops of our numerical experiments.

Example 3.14. Consider an IDC method that is constructed using six uniformly distributed quadrature nodes and three loops of a second-order AB method in the prediction and correction steps (an RK-2 method is used to start the multistep method as necessary). We use this method to solve IVP (3-4). Let H denote the interval size for a single step of the IDC method, and $h = \frac{H}{5}$ the subinterval size. The numerical results are in Table 4 show the inconsistent accuracy increase after the first correction loop. Computing the Taylor expansion of the numerical solution about $t = 0$ with $\mathcal{O}(h^7)$ truncation error, the rescaled error vector satisfies

$$\begin{aligned}\vec{e}^{[0]} = \Big\{0, \frac{h}{750} + \frac{h^2}{15{,}000} + \frac{h^3}{375{,}000} + \frac{h^4}{11{,}250{,}000},\\ \frac{7h}{1500} + \frac{2h^2}{1{,}875} + \frac{4h^3}{46{,}875} + \frac{4h^4}{703{,}125}, \frac{h}{125} + \frac{9h^2}{2{,}500} + \frac{81h^3}{125{,}000} + \frac{81h^4}{1{,}250{,}000},\\ \frac{17h}{1500} + \frac{14h^2}{1{,}875} + \frac{1643h^3}{750{,}000} + \frac{256h^4}{703{,}125}, \frac{11h}{750} + \frac{19h^2}{1500} + \frac{191h^3}{37{,}500} + \frac{5521h^4}{4{,}500{,}000}\Big\} + \mathcal{O}(h^5).\end{aligned}$$

	AB-2 pred.		1 AB-2 corr. loop		2 AB-2 corr. loop	
steps	error	order	error	order	error	order
5	1.55E-3	–	4.41E-6	–	7.74E-9	–
10	3.93E-4	1.98	4.56E-7	3.27	1.80E-10	5.43
15	1.76E-4	1.99	1.26E-7	3.18	1.88E-11	5.57
20	9.90E-5	1.99	5.10E-8	3.14	3.50E-12	5.84
25	6.34E-5	1.99	2.55E-8	3.11	8.61E-13	6.28

Table 4. Error and order of convergence for an IDC method constructed using a 2-step multistep method (AB) and uniformly distributed quadrature nodes. The error and rate of convergence are computed at $T = 1$.

By Definition 3.5, $\vec{e}^{[0]}$ has only one degree of smoothness since the leading term in

$$\overrightarrow{\hat{d}_2 e^{[0]}},$$

by Definition 3.2, is $\mathbb{O}(\frac{1}{h})$. Since $\vec{e}^{[0]}$ has no more than one degree of smoothness in the discrete sense, this limits the increase in convergence rate for IDC methods, although a high-order method is applied in the correction steps.

4. Comparisons between IDC and RK methods

IDC methods constructed using single-step integrators can be formulated into arbitrary high-order RK methods. This is of particular interest because RK methods are traditionally constructed by satisfying order conditions [6]; the number of order conditions to be satisfied grows exponentially as the order increases, making it difficult, if not impossible, to solve for the nodes, weights, and stage weights exactly. Here, we address how IDC methods constructed with RK integrators and uniformly distributed nodes can be formulated as a high-order RK method whose nodes, weights, and stage weights are known exactly. In Section 4.1, we describe the Butcher tableau structure for IDC-FE methods formulated as a high-order RK method. Then, we bound the local truncation error arising from IDC methods formulated as a high-order RK method; this bound on the local truncation error can be used to give an estimate for the global error, in essence, proving the convergence of IDC methods. The efficiency of IDC methods is then compared with known RK methods in Section 4.2. In general, known RK methods are more efficient than IDC methods for low-order schemes. For high-order schemes, comparable efficiency is observed numerically; the accuracy regions in Section 4.3 agree qualitatively with the efficiency comparisons. In Section 4.4, we show that IDC methods offer a much larger stability region compared with known RK methods. Additionally, as the order of the embedded integrator is increased, the stability region of an IDC method also increases.

4.1. ***Constructing RK methods using an IDC-FE scheme.*** The following two points of view are equivalent: RK methods can be constructed using IDC ideas, or an IDC method can be reformulated as an RK method. The node points c_j, weights b_k and stage weights a_{jk} are often conveniently expressed in a Butcher tableau format using matrix A, and vectors b and c.

$$\begin{array}{c|c} \vec{c} & A \\ \hline & \vec{b}^T \end{array} .$$

Here, we illustrate the Butcher tableau structure of IDC methods constructed using forward Euler time integrators. The algorithm is easily generalized for generating IDC-RK Butcher tableaus. In this section, we adopt a Matlab-style notation in our algorithms, where $A(j, :)$ denotes the j-th row of matrix A, and $A(:, j)$ denotes the j-th column of matrix A.

Proposition 4.1. *An IDC method constructed using* $(M+1)$ *quadrature nodes and* (k_l+1) *prediction/correction iterations of an* s*-stage RK method, can be reformulated as an* $((k_l+1)\cdot s\cdot M)$*-stage RK method.*

For example, an IDC method constructed with four quadrature nodes, $(M = 3)$, a forward Euler prediction $(s = 1)$, and three correction loops $(k_l = 3)$, can be reformulated as a 12-stage RK method. Let's examine the structure of this $((k_l + 1)\cdot s\cdot M)$-stage RK method. Suppose $(M + 1)$ quadrature nodes, notated as before in (3-2),

$$0 = t_0 < t_1 < t_2 < \ldots < t_M = H,$$

have subinterval sizes

$$h_m = t_m - t_{m-1}, \quad m = 1, \ldots, M.$$

Then the prediction step of the IDC method constructed using forward Euler updates can be formulated as an RK method with the following Butcher array format:

$$\begin{array}{c|ccccc} t_0 & & & & & \\ t_1 & h_1 & & & & \\ t_2 & h_1 & h_2 & & & \\ \vdots & \vdots & & \ddots & & \\ t_{M-1} & h_1 & h_2 & \ldots & h_{M-1} & \\ \hline & h_1 & h_2 & \ldots & h_{M-1} & h_M \end{array}$$

We label components of the above Butcher tableau conventionally:

$$\begin{array}{c|c} \vec{c}_1 & A_1 \\ \hline & \vec{b}_1^T \end{array}$$

The first correction loop can now be included (2-7). The updated Butcher tableau takes the form

$$\begin{array}{c|cc} \vec{c}_1 & A_1 & Z \\ \vec{c}_2 & P_1 & A_2 \\ \hline & \vec{d}_1^T & \vec{b}_2^T \end{array},$$

where Z is a $M \times M$ matrix of zeros,

$$\vec{c}_2 = \left[t_M, t_1, t_2, \ldots, t_{M-1}\right]^T,$$

$$P_1 = \begin{bmatrix} h_1 & h_2 & h_3 & \ldots & h_M \\ \tilde{S}_{10} & \tilde{S}_{11} & \tilde{S}_{12} & \ldots & \tilde{S}_{1,M-1} \\ \tilde{S}_{20} & (\tilde{S}_{21} - h_2) & \tilde{S}_{22} & \ldots & \tilde{S}_{2,M-1} \\ \tilde{S}_{30} & (\tilde{S}_{31} - h_2) & (\tilde{S}_{32} - h_3) & \ldots & \tilde{S}_{3,M-1} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \tilde{S}_{M-1,0} & (\tilde{S}_{M-1,1} - h_2) & (\tilde{S}_{M-1,2} - h_3) & \ldots & \tilde{S}_{M-1,M-1} \end{bmatrix},$$

where the terms

$$\tilde{S}_{ij} = \begin{cases} S_{ij} & i = 1, \quad j = 0, \ldots, M, \\ S_{ij} + S_{i-1,j} & i = 2, \ldots, M, \quad j = 0, \ldots, M, \end{cases}$$

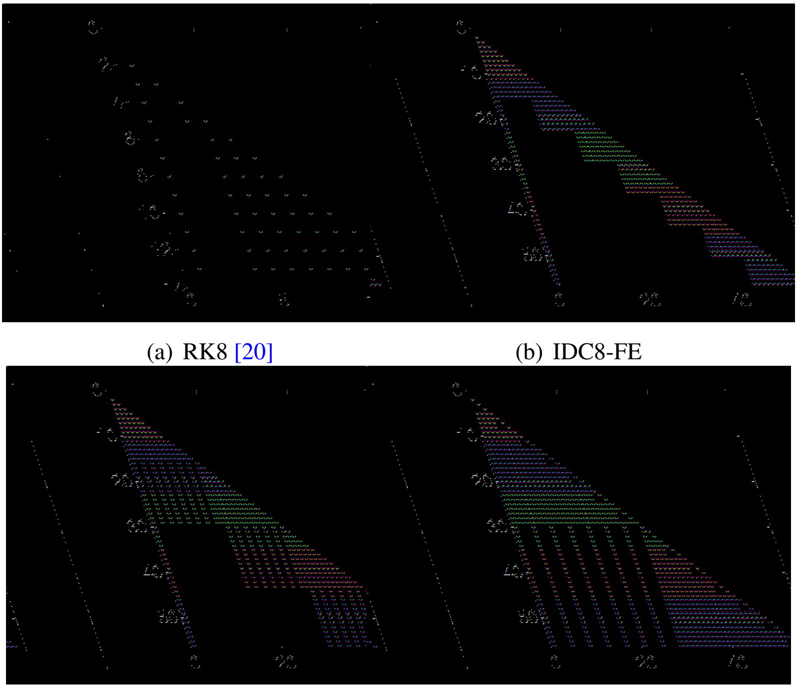
are the sums of the integration matrix defined in (2-12),

$$A_2 = \begin{bmatrix} 0 & 0 & 0 & 0 & \ldots & 0 \\ \tilde{S}_{1M} & 0 & 0 & 0 & \ldots & 0 \\ \tilde{S}_{2M} & h_2 & 0 & 0 & \ldots & 0 \\ \tilde{S}_{3M} & h_2 & h_3 & 0 & \ldots & 0 \\ \vdots & \vdots & \vdots & \ddots & \ddots & 0 \\ \tilde{S}_{M-1,M} & h_2 & h_3 & \ldots & h_{M-1} & 0 \end{bmatrix},$$

and

$$\vec{d}_1 = \left[\tilde{S}_{M0}, (\tilde{S}_{M1} - h_2), (\tilde{S}_{M2} - h_3), \ldots, (\tilde{S}_{M,M-1} - h_M)\right]^T,$$
$$\vec{b}_2 = \left[\tilde{S}_{MM}, h_2, h_3, \ldots, h_M\right]^T.$$

Subsequent correction steps can be added into a Butcher tableau format in a similar fashion. This results in a distinct block structure, since at the k-th correction loop,



(a) RK8 [20]

(b) IDC8-FE

(c) IDC8-RK2

(d) IDC8-RK4

Figure 1. The sparsity structure of an RK8 [20] Butcher tableau and various IDC8 Butcher tableaus. The block structure for the IDC weights are evident; for example, IDC8-RK2 shows the prediction loop, and the subsequent three correction loops.

only the initial value $f(t, y_0)$, the previous approximations $f(t, \eta^{[k-1]})$, and the current iterates $f(t, \eta^{[k]})$ are used.

This block structure is more easily seen in Figure 1, where uniformly spaced quadrature nodes are used to construct various IDC schemes. For future reference, we adopt the following notation to denote our IDC schemes: IDCn-RKp denotes an n-th order IDC scheme constructed using p-th order RK integrators. To construct an n-th order IDC scheme, either $(n+1)$ uniformly distributed quadrature nodes, or $(\lceil \frac{n}{2} \rceil + 1)$ Gauss–Lobatto nodes are used. The only time we distinguish between using uniformly distributed and Gauss–Lobatto nodes is when forward Euler integrators are used to construct the IDC scheme; in all other cases, we use uniform nodes to achieve the order of accuracy desired. In Figure 1, IDC8-FE denotes an eighth-order IDC scheme constructed using forward Euler updates, IDC8-RK2 denotes an eighth-order IDC scheme constructed using a trapezoidal RK2 scheme for the prediction and correction steps, and IDC8-RK4 denotes an eighth-order

IDC scheme constructed using an RK4 predicition and correction loop. Several observations can be made presently. The number of stages for the IDC methods are consistent with Proposition 4.1. Specifically, IDC8-FE has $M = 7, s = 2, k_l = 7$, resulting in 56 overall stages. IDC8-RK2 has $M = 7, s = 2, k_l = 3$, and IDC8-RK4 has $M = 7, s = 4, k_l = 1$, both resulting in 56 overall stages. The sparsity structure of the Butcher tableau which arises from the prediction and correction steps is also evident. For example, IDC8-RK2 shows the prediction step and three subsequent correction steps.

It is important to note that the stage weights, a_{ij}, can be computed *exactly* using a symbolic manipulator such as Maple or Mathematica. This contrasts with most other optimization schemes for generating RK methods, where the coefficients have to be approximated numerically. For eighth- or lower-order schemes, computing the stage weights to double precision is sufficient. From numerical experiments, it seems that ninth-order IDC schemes (or higher) require quad precision or better.

4.2. ***Efficiency comparison.*** In order to compare how various p-th order RK methods stack up against each other, a quantitative measure is the so-called efficiency: how much computational effort is required to obtain a certain error tolerance. To make this measurement, we need to review the computational effort of IDC/RK methods, as well as bound the local truncation error (LTE).

In solving IVP (2-1), the evaluation of $f(t, y)$ is usually the most computationally expensive component. Hence, we will use the number of function evaluations, n_{fe}, (or equivalently, the number of stages of an RK method), as a measure of the computational effort. Recall that an IDC method constructed using $(M + 1)$ quadrature nodes, k_l correction loops, and an s-stage RK integrator requires $((M-1)\cdot(k_l+1)\cdot s)$ function evaluations (stages). For a p-th order IDC method, this corresponds to at least $p\,(p-1)$ function evaluations when p uniformly spaced quadrature nodes are used, and at least $(\frac{p}{2} - 1)\,p$ function evaluations when $\frac{p}{2}$ Gaussian nodes are used. Compared to classically known p-th order RK methods which involve s_p stages, IDC methods require significantly more function evaluations per iteration. This is offset, however, by the smaller LTE that arises from IDC methods.

The LTE which arises from solving $y' = f(t, y)$ can be computed by taking the appropriate Taylor expansions of the scheme. For a p-th order method [10] the LTE can be expressed as

$$\text{LTE} = \sum_{i=p+1}^{\infty} h^i \left(\sum_{j=1}^{\lambda_i} \alpha_{ij} D_{ij} \right).$$

Here, h is the interval size, D_{ij} are the elementary differentials (sums of products of partial derivatives of $f(t, y)$), α_{ij} are the truncation error coefficients, and λ_i denotes the number of elementary differentials of order $\mathcal{O}(h^i)$. Consequently, a very

Method	# Stages	LTE	Efficiency
RK4 (classical)	4	1.0417E-2	1
IDC4-FE (unif)	12	7.716 E-4	1.78
IDC4-FE (gauss)	8	3.906 E-3	1.64
IDC4-RK2 (unif)	12	5.144 E-4	1.64
SDC4-RK2 (gauss)	8	2.6042E-3	1.52
RK6 [20]	9	8.4369E-7	1
RK6 [17]	7	1.455 E-2	3.13
IDC6-FE (unif)	30	1.0000E-6	3.42
IDC6-FE (gauss)	18	5.5014E-5	3.63
IDC6-RK2 (unif)	30	8.8889E-7	3.36
IDC6-RK3 (unif)	30	4.4444E-7	3.04
RK8 [20]	13	3.8872E-6	1
RK8 [3]	11	2.1957E-5	1.03
IDC8-FE (unif)	56	6.776 E-10	1.65
IDC8-FE (gauss)	32	1.181 E-7	1.67
IDC8-RK2 (unif)	56	5.0193E-10	1.59
IDC8-RK4 (unif)	56	3.0689E-11	1.17

Table 5. A comparison of classical RK methods and IDC methods. The second column lists the effective number of stages, the third column lists a bound on the LTE coefficients, and the last column is the computed efficiency between the respective orders. Eighth-order IDC methods are almost as efficient as an RK8 method. The LTE for twelfth-order methods is not presented due to machine precision restrictions. Also, since three Gauss–Lobatto nodes are in fact uniformly spaced, $x = \{0, 0.5, 1\}$, we are able to generate a fourth-order IDC scheme using three Gauss–Lobatto nodes and RK2 integrators for the prediction and correction loops. Note than an efficiency close to 1 is optimal.

crude bound for the LTE, if h is sufficiently small, is

$$\text{LTE} \le h^{p+1} \cdot \lambda_{p+1} \cdot \|\alpha_{p+1,j}\|_\infty \cdot \|D_{p+1,j}\|_\infty.$$

We note that this local error estimate gives a bound on the global error [6], proving the convergence of IDC-RK methods.

Now, consider the LTE for two p-th order RK methods,

$$(\text{LTE})_1 = c_1 h^{p+1} \cdot (\lambda_{p+1} \|D_{p+1,j}\|_\infty),$$
$$(\text{LTE})_2 = c_2 h^{p+1} \cdot (\lambda_{p+1} \|D_{p+1,j}\|_\infty).$$

If both LTEs are bounded by the same tolerance ϵ, then the largest step size that will satisfy this tolerance for both methods is

$$h_1 = \left(\frac{\epsilon}{\beta c_1}\right)^{1/(p+1)}, \quad h_2 = \left(\frac{\epsilon}{\beta c_2}\right)^{1/(p+1)},$$

respectively, where $\beta = (\lambda_{p+1} \| D_{p+1,j} \|_\infty)$. If method one is computed in s_1 stages and method two is computed in s_2 stages, then the total amount of work done by each methods is s_i/h_i, since $1/h_i$ is the number of iterations required, and s_i is the cost per iteration. A measure of efficiency is then given by the ratio of the amount of work done:

$$\text{efficiency} = \frac{s_2/h_2}{s_1/h_1} = \frac{s_2}{s_1}\left(\frac{c_2}{c_1}\right)^{1/(p+1)}. \tag{4-1}$$

Using (4-1), the efficiencies for various IDC methods are computed and compared to classically known RK methods. In Table 5, we list the number of stages for each method, a bound on the LTE (using a code provided in [10]), and the computed efficiency. An efficiency close to 1 is optimal while an efficiency of 1.5 means it takes 50% more work to achieve the same error tolerance. We compute the LTEs for eighth- and lower-order schemes to avoid machine precision issues. (As mentioned in the previous section, the accuracy increase is lost when the nodes are nonuniformly spaced; thus, IDC schemes constructed using Gaussian nodes and high-order integrators are in general less efficient than IDC schemes constructed using uniformly spaced nodes and high-order integrators. Consequently, the efficiency analysis for other IDC schemes using Gaussian nodes is not presented.)

Two observations are in order: first, that the efficiency of IDC schemes improves as the order of the embedded integrator is increased; and second, that IDC8 schemes are comparable, in terms of efficiency, to RK8 schemes. Although we are unable to accurately compute the LTE for higher than eighth-order schemes, we show that twelfth-order IDC schemes are comparable in terms of efficiency to known RK-12 schemes by generating their accuracy regions, defined in Section 4.3.

4.3. ***Accuracy region.*** A more visual way to compare these IDC methods is to plot the accuracy region for each method. Specifically, the following IVP,

$$y'(t) = \lambda y(t), \quad y(0) = 1, \tag{4-2}$$

is solved for various λ's in the complex plane. A contour plot of the resulting error at $T = 1$ is called the accuracy region. Figures 2–5 show the accuracy regions for classical RK and IDC methods. Consistent with the efficiency analysis, the IDC4 and IDC6 schemes perform poorly in contrast with classical RK methods. IDC8-RK4 has a comparable accuracy region with RK8. The accuracy regions for IDC12 methods are plotted, even though the efficiency is not computed in the previous section. It appears that IDC12-RK3 and IDC12-RK4 might be more efficient than classically known RK12 schemes. One should also note that the accuracy regions for IDC methods increase in area as the order of the embedded integrator is increased.

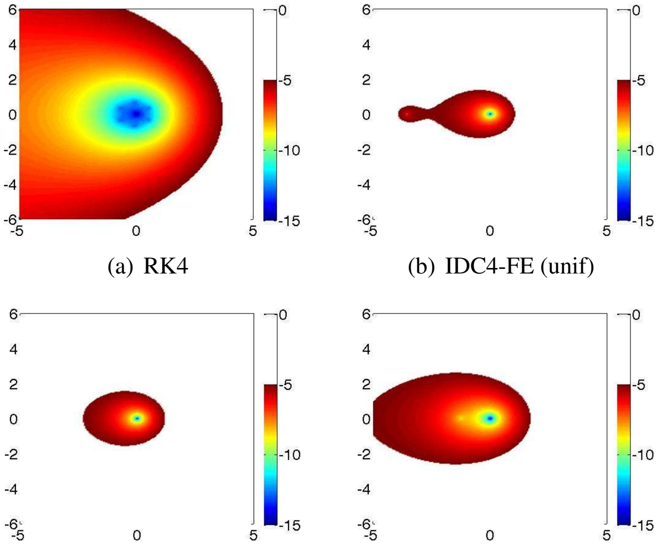
4.4. ***Stability region.*** Another way to quantitatively compare RK and IDC methods is to perform a linear stability analysis of the methods, and identify restrictions on the possible time steps. The linear stability region, S, is the subset of the complex plane, $\mathbb{C}$, satisfying

$$S = \{\lambda : \mathrm{Am}(\lambda) \leq 1\},$$

where $\mathrm{Am}(\lambda)$, the amplification factor for a numerical method, can be interpreted as the numerical solution of IVP (4-2)

$$y'(t) = \lambda y(t), \quad y(0) = 1,$$

after a time step of size one. To quantify the size of these linear stability regions, we measure the *linear stability radius*, the real interval $\{z : \mathrm{Re}(z) \in S\}$, and the maximum imaginary value, $\sup |\mathrm{Im}(z)|$, $z \in S$.



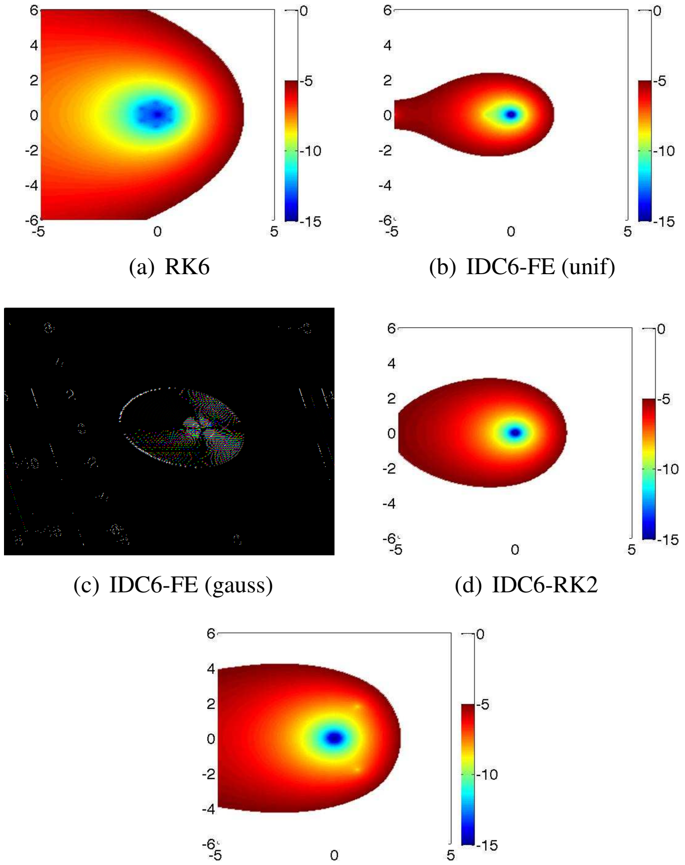
(a) RK4

(b) IDC4-FE (unif)

(c) IDC4-FE (gauss)

(d) IDC4-RK2

Figure 2. Accuracy plots for various fourth-order RK and IDC methods. Each plot was generated after 48 function evaluations. The RK4 method is vastly superior to the IDC methods.



(a) RK6

(b) IDC6-FE (unif)

(c) IDC6-FE (gauss)

(d) IDC6-RK2

(e) IDC6-RK3

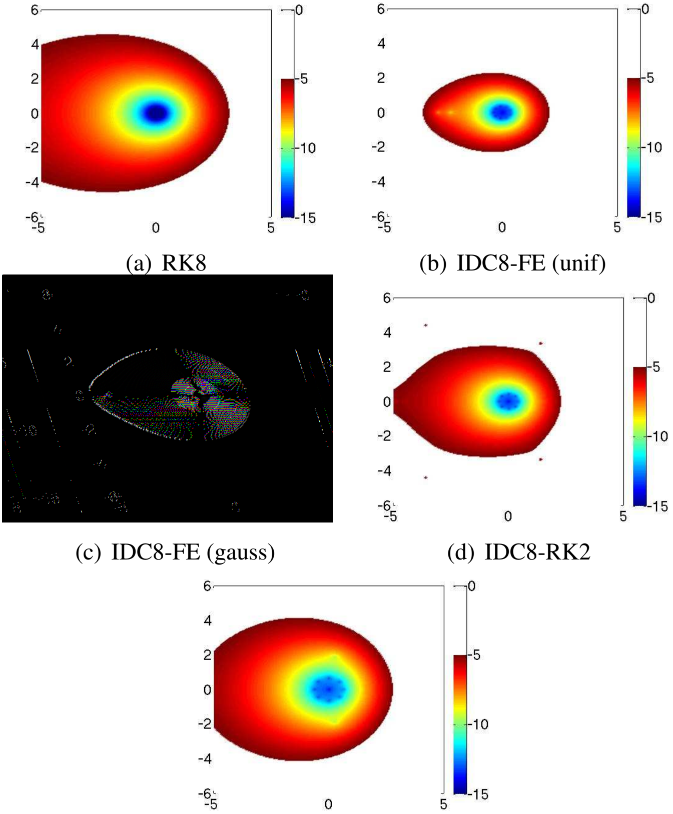
Figure 3. Accuracy plots for various sixth-order RK and IDC methods are generated using ≈ 60 function evaluations. The RK6 method has a larger accuracy region. Also, observe that the accuracy regions for the IDC methods increase with the order of the embedded integrator.

Definition 4.2. The linear stability radius is defined to be the radius of the largest disc that can fit inside the stability region,

$$\rho = \sup \{r : D(r) \in S\},$$

where $D(r)$ is the disc $D(r) = \{z \in \mathbb{C} : |z + r| \leq r\}$.

This measure of the stability region is argued to be a good compromise between stretching the stability region in the real and in the imaginary directions [13; 19].



(a) RK8

(b) IDC8-FE (unif)

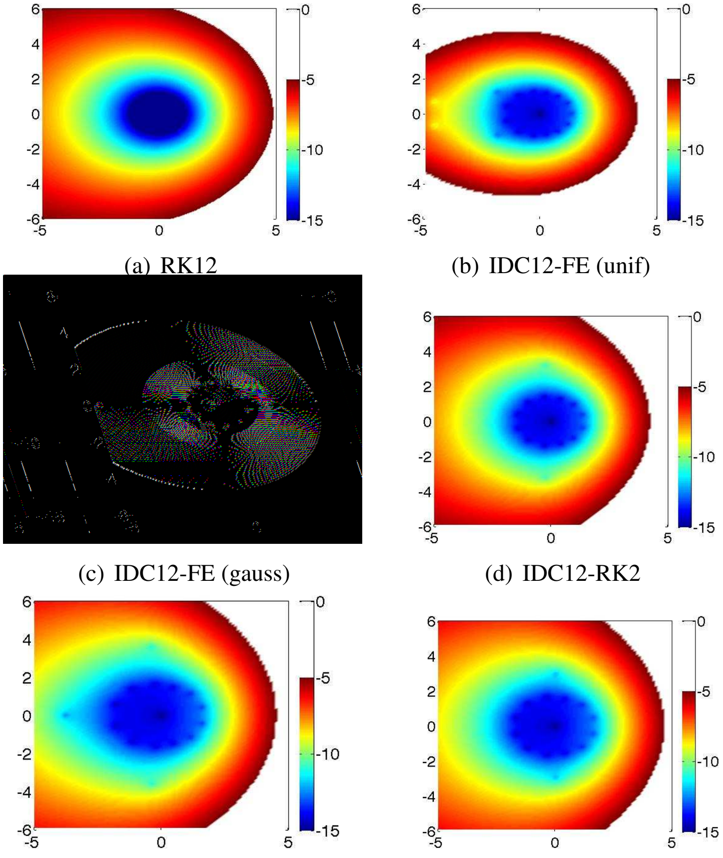
(c) IDC8-FE (gauss)

(d) IDC8-RK2

(e) IDC8-RK4

Figure 4. The accuracy plots for various eighth-order RK and IDC methods are generated after ≈ 56 function evaluations. Notice that accuracy regions for IDC methods get larger as the order of the low-order integrator is increased. The accuracy region for IDC8-RK4 is comparable to the accuracy region for RK8, which is consistent with the efficiency analysis.

We plot the stability regions for various IDC and RK methods in Figure 6. In all cases, p-th order IDC methods offer a larger stability region in contrast with classically derived p-th order RK methods. Additionally, the stability regions of IDC methods increase with the order of the integrator used to construct the scheme; for example, IDC8-RK4 has a larger stability region than IDC8-RK2. Quantitative comparison of the stability regions are given in Table 6.



(a) RK12

(b) IDC12-FE (unif)

(c) IDC12-FE (gauss)

(d) IDC12-RK2

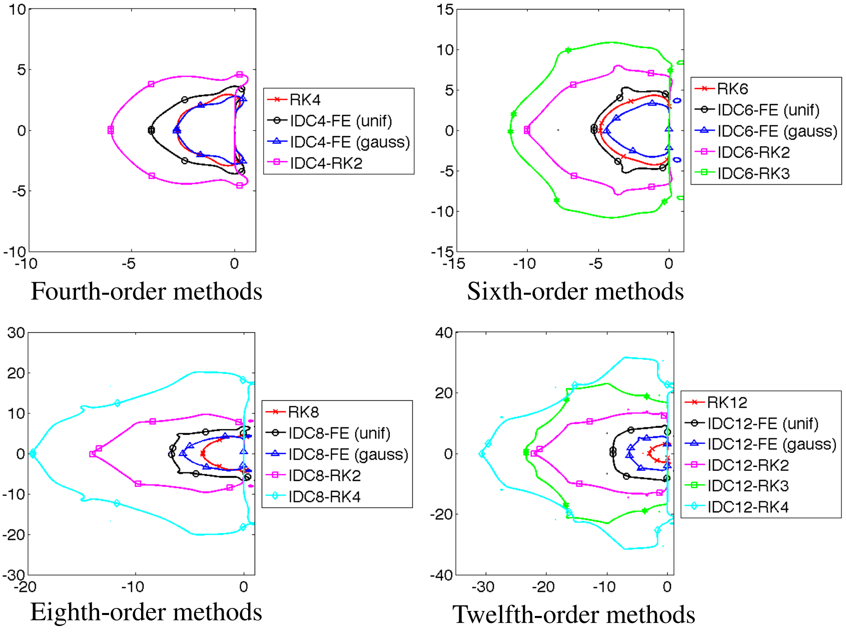
(e) IDC12-RK3

(f) IDC12-RK4

Figure 5. The accuracy plots for various twelfth-order RK and IDC methods are generated after ≈ 132 function evaluations. The accuracy region for IDC12-RK3 and IDC12-RK4 appear larger than the classically known RK12 scheme.

5. Conclusion

In this paper, we studied a class of novel correction methods, IDC methods, constructed using high-order integrators within the prediction and correction loops. It was also shown that the accuracy of an IDC method is closely related to the smoothness of its error vector. Unlike IDC methods constructed with uniform quadrature points, the order of accuracy for IDC methods constructed with a general nonuniform distribution of quadrature nodes does not increase by r orders if an r-th order RK correction step is applied; for multistep methods, the accuracy of an IDC method depends heavily on the starting method. Finally, IDC methods are



Fourth-order methods

Sixth-order methods

Eighth-order methods

Twelfth-order methods

Figure 6. Stability regions for fourth-, sixth-, eighth- and twelfth-order IDC methods. The stability regions of IDC methods are larger than that of the RK method. Additionally, the stability region of the IDC methods increase with the order of the integrator used to construct the scheme.

viewed as a means for generating high-order RK methods. The efficiency, stability, and accuracy of IDC methods are compared with RK methods. As a family of methods, these IDC schemes are capable of matching the efficiency of optimized high-order RK methods. Additionally, superior regions of absolute stability are observed for IDC methods constructed using high order integrators.

Present studies and analyses are being conducted on IDC methods constructed using diagonally implicit Runge–Kutta integrators and IDC methods constructed using additive Runge–Kutta integrators.

References

[1] Claudio Canuto, M. Yousuff Hussaini, Alfio Quarteroni, and Thomas A. Zang, *Spectral methods in fluid dynamics*, Springer Series in Computational Physics, Springer, New York, 1988. MR 89m:76004 Zbl 0658.76001

[2] Andrew Christlieb, Benjamin Ong, and Jing-Mei Qiu, *Integral deferred correction methods constructed with high order Runge–Kutta integrators*, Math. Comp., accepted.

[3] A. R. Curtis, *An eighth order Runge–Kutta process with eleven function evaluations per step*, Numer. Math. **16** (1970), 268–277. MR 42 #5444 Zbl 0194.18902

Method	$\bar{\rho}$	Real Interval	Max Im
RK4 (classical)	1.39	$[-2.78, 0.24]$	2.93
IDC4-FE (unif)	2.00	$[-4.05, 0.43]$	3.60
IDC4-FE (gauss)	1.40	$[-2.81, 0.41]$	2.79
IDC4-RK2 (unif)	3.00	$[-6.00, 0.63]$	4.57
RK6 [20]	2.43	$[-4.85, 0.00]$	4.30
IDC6-FE (unif)	2.66	$[-5.32, 0.01]$	5.27
IDC6-FE (gauss)	2.14	$[-4.42, 0.00]$	3.32
IDC6-RK2 (unif)	4.76	$[-10.00, 0.17]$	7.98
IDC6-RK3 (unif)	5.59	$[-11.18, 0.16]$	10.84
RK8 [20]	1.90	$[-3.80, 0.33]$	4.66
IDC8-FE (unif)	3.33	$[-6.65, 0.54]$	6.66
IDC8-FE (gauss)	2.78	$[-6.92, 0.00]$	4.23
IDC8-RK2 (unif)	6.58	$[-14.0, \ \ 0.83]$	9.64
IDC8-RK4 (unif)	9.61	$[-19.49, 1.14]$	20.09
RK12	1.51	$[-3.00, 0.00]$	2.75
IDC12-FE (unif)	4.60	$[-9.01, 0.00]$	9.09
IDC12-FE (gauss)	3.34	$[-7.20, 0.25]$	4.97
IDC12-RK2 (unif)	9.94	$[-23.00, 0.00]$	13.47
IDC12-RK3 (unif)	11.45	$[-23.25, 0.00]$	23.02
IDC12-RK4 (unif)	14.92	$[-30.63, 1.05]$	31.61

Table 6. A table quantifying the stability regions of RK and IDC methods. For completeness, we list the linear stability radius ρ, the real interval, and the maximum imaginary value.

[4] Alok Dutt, Leslie Greengard, and Vladimir Rokhlin, *Spectral deferred correction methods for ordinary differential equations*, BIT **40** (2000), 241–266. MR 2001e:65104 Zbl 0959.65084

[5] Thomas Hagstrom and Ruhai Zhou, *On the spectral deferred correction of splitting methods for initial value problems*, Commun. Appl. Math. Comput. Sci. **1** (2006), 169–205. MR2008k:65131

[6] E. Hairer, S. P. Nørsett, and G. Wanner, *Solving ordinary differential equations, i: nonstiff problems*, 2nd ed., Springer Series in Computational Mathematics, no. 8, Springer, Berlin, 1993. MR 94c:65005 Zbl 0789.65048

[7] E. Hairer and G. Wanner, *Solving ordinary differential equations, ii: stiff and differential-algebraic problems*, 2nd ed., Springer Series in Computational Mathematics, no. 14, Springer, Berlin, 1996. MR 97m:65007 Zbl 0859.65067

[8] A. Hansen and J. Strain, *On the order of deferred correction*, preprint.

[9] ———, *Convergence theory for spectral deferred correction*, preprint, 2005.

[10] M. E. Hosea, *A new recurrence for computing Runge–Kutta truncation error coefficients*, SIAM J. Numer. Anal. **32** (1995), no. 6, 1989–2001. MR 96m:65073 Zbl 0841.65071

[11] Jingfang Huang, Jun Jia, and Michael Minion, *Accelerating the convergence of spectral deferred correction methods*, J. Comput. Phys. **214** (2006), no. 2, 633–656. MR 2006k:65173 Zbl 1094.65066

[12] ———, *Arbitrary order Krylov deferred correction methods for differential algebraic equations*, J. Comput. Phys. **221** (2007), no. 2, 739–760. MR 2008a:65134 Zbl 1110.65076

[13] Rolf Jeltsch and Olavi Nevanlinna, *Largest disk of stability of explicit Runge–Kutta methods*, BIT **18** (1978), no. 4, 500–502. MR 80b:65099 Zbl 0399.65051

[14] Anita T. Layton, *On the choice of correctors for semi-implicit Picard deferred correction methods*, Appl. Numer. Math. **58** (2008), no. 6, 845–858. MR 2420621 Zbl 1143.65057

[15] Anita T. Layton and Michael L. Minion, *Implications of the choice of quadrature nodes for Picard integral deferred corrections methods for ordinary differential equations*, BIT **45** (2005), no. 2, 341–373. MR 2006h:65087 Zbl 1078.65552

[16] ———, *Implications of the choice of predictors for semi-implicit Picard integral deferred correction methods*, Commun. Appl. Math. Comput. Sci. **2** (2007), 1–34. MR 2008e:65252 Zbl 1131.65059

[17] H. A. Luther, *An explicit sixth-order Runge–Kutta formula*, Math. Comp. **22** (1968), no. 102, 434–436. Zbl 0155.20402

[18] Michael L. Minion, *Semi-implicit spectral deferred correction methods for ordinary differential equations*, Commun. Math. Sci. **1** (2003), no. 3, 471–500. MR 2005f:65085 Zbl 1088.65556

[19] Brynjulf Owren and Kristian Seip, *Some stability results for explicit Runge–Kutta methods*, BIT **30** (1990), no. 4, 700–706. MR 91m:65203 Zbl 0718.65061

[20] James Verner, *High order Runge–Kutta methods*.

[21] Yinhua Xia, Yan Xu, and Chi-Wang Shu, *Efficient time discretization for local discontinuous Galerkin methods*, Discrete Contin. Dyn. Syst. Ser. B **8** (2007), 677–693. MR 2008e:65307 Zbl 1141.65076

Received November 13, 2008. Revised January 20, 2009.

ANDREW CHRISTLIEB: christlieb@math.msu.edu
Michigan State University, Department of Mathematics, D304 Wells Hall,
East Lansing, MI 48824-1027, United States

BENJAMIN ONG: bwo@math.msu.edu
Michigan State University, Department of Mathematics, D304 Wells Hall,
East Lansing, MI 48824-1027, United States

JING-MEI QIU: jingqiu@mines.edu
Mathematical and Computer Science, Colorado School of Mines, Golden, CO, 80401, United States

COMM. APP. MATH. AND COMP. SCI.
Vol. 4, No. 1, 2009

A HIGHER-ORDER UPWIND METHOD FOR VISCOELASTIC FLOW

ANDREW NONAKA, DAVID TREBOTICH, GREGORY MILLER,
DANIEL GRAVES AND PHILLIP COLELLA

We present a conservative finite difference method designed to capture elastic wave propagation in viscoelastic fluids in two dimensions. We model the incompressible Navier–Stokes equations with an extra viscoelastic stress described by the Oldroyd-B constitutive equations. The equations are cast into a hybrid conservation form which is amenable to the use of a second-order Godunov method for the hyperbolic part of the equations, including a new exact Riemann solver. A numerical stress splitting technique provides a well-posed discretization for the entire range of Newtonian and elastic fluids. Incompressibility is enforced through a projection method and a partitioning of variables that suppresses compressive waves. Irregular geometry is treated with an embedded boundary/volume-of-fluid approach. The method is stable for time steps governed by the advective Courant–Friedrichs–Lewy (CFL) condition. We present second-order convergence results in L^1 for a range of Oldroyd-B fluids.

1. Introduction

The governing equations for viscoelastic flow of an Oldroyd-B fluid are the incompressible Navier–Stokes equations plus an extra viscoelastic stress described by the Oldroyd-B constitutive equations:

$$\frac{\partial \boldsymbol{u}}{\partial t} + (\boldsymbol{u}\cdot\nabla)\boldsymbol{u} - \frac{1}{\rho}\nabla\cdot\boldsymbol{\tau} = -\frac{1}{\rho}\nabla p + \frac{\mu_s}{\rho}\Delta\boldsymbol{u}, \tag{1}$$

$$\nabla\cdot\boldsymbol{u} = 0, \tag{2}$$

$$\frac{\partial \boldsymbol{\tau}}{\partial t} + (\boldsymbol{u}\cdot\nabla)\boldsymbol{\tau} - (\nabla\boldsymbol{u})\boldsymbol{\tau} - \boldsymbol{\tau}(\nabla\boldsymbol{u})^T = \frac{\mu_p}{\lambda}2\boldsymbol{D} - \frac{1}{\lambda}\boldsymbol{\tau}, \tag{3}$$

where $\boldsymbol{u}$ is the fluid velocity, $\boldsymbol{\tau}$ is the polymeric stress tensor, p is the isotropic pressure, and $\boldsymbol{D} = [\nabla\boldsymbol{u} + (\nabla\boldsymbol{u})^T]/2$ is the rate-of-strain tensor. The parameters that describe the fluid are the density, ρ, relaxation time, λ, and the solvent and polymeric

MSC2000: 65N06, 76D05.

Keywords: viscoelasticity, Oldroyd-B fluid, Godunov method, Riemann solver, projection method, embedded boundaries.

contributions to the total viscosity, $\mu = \mu_s + \mu_p$. The dimensionless parameters that characterize these types of flows are the Reynolds number, $\mathrm{Re} = \rho U L/\mu$, and the Weissenberg number, $\mathrm{We} = \lambda U/L$, where U and L are the characteristic velocity and length.

Though the Reynolds number and the Weissenberg number independently characterize viscoelastic flows, it is the elastic Mach number, $\mathrm{Ma} = \sqrt{\mathrm{Re} \cdot \mathrm{We}}$, that is the critical parameter in determining well-posedness of the system. In particular, the system of equations exhibits a change in type from parabolic to hyperbolic when the elastic Mach number becomes supercritical ($\mathrm{Ma} > 1$), admitting propagation of discontinuities. This mathematical behavior was alluded to in the experimental results of Ultman and Denn [33] and formally noted in [7; 18]. Joseph suggested that a method suitable for transonic flows may be needed to capture the transition to supercritical flows in viscoelasticity [17]. The analysis described in [30] capitalized on this concept in the design of a numerical algorithm that resolves unsteady elastic wave behavior in viscoelastic fluids.

In this paper, we extend the previous numerical algorithm [30] by leveraging the conservative hyperbolic formulation described therein to design a suitable higher resolution upstream method for the hyperbolics. In the original algorithm the Oldroyd-B equation is recast into a well-posed hyperbolic form with source terms using a stress-splitting technique; a Lax–Wendroff method is used to discretize the quasilinear form of the hyperbolic part in the context of a predictor-corrector projection method. (Projection methods are an approach to enforcing the constraint in incompressible flows [3; 2] and have proven to be successful in treating unsteady viscoelastic flows [19; 30].) Our new method uses a second-order Godunov method [5; 6], instead of Lax–Wendroff as in [30], to discretize the hyperbolic part of the equations, resulting in two immediate advantages. First, the maximum time step is increased by a factor of four to allow an advective CFL number restriction of $0 < \mathrm{CFL} < 1$. Second, we can apply second-order conservative finite volume techniques which have been developed for hyperbolic conservation laws [6], elliptic equations [16], and parabolic equations [22] in an embedded boundary (EB) framework for irregular geometry. Our results are consistent with the modified equation analysis in these methods, and we obtain second-order solution error convergence in L^1 for a range of Oldroyd-B fluids.

2. Hyperbolic analysis

Through the introduction of the inverse deformation tensor, $\boldsymbol{g}$, which links material (Lagrangian) coordinates, $\boldsymbol{X}$, and spatial (Eulerian) coordinates, $\boldsymbol{x}$, as in

$$g_{\alpha\beta} = \frac{\partial X_\alpha}{\partial x_\beta}, \tag{4}$$

the advective part of the PDE for viscoelastic stress (3) may be put in conservation form. The quantity $\boldsymbol{M}$ is conserved:

$$\boldsymbol{M} \equiv \boldsymbol{g}\left(\boldsymbol{\tau} + \rho a^2 \boldsymbol{I}\right)\boldsymbol{g}^T, \tag{5}$$

$$\frac{\partial \boldsymbol{M}}{\partial t} + \nabla\cdot(\boldsymbol{u}\otimes\boldsymbol{M}) = \boldsymbol{g}\left[-\frac{1}{\lambda}\boldsymbol{\tau} + \left(\frac{\mu_p}{\lambda} - \rho a^2\right)2\boldsymbol{D}\right]\boldsymbol{g}^T, \tag{6}$$

$$\frac{\partial \boldsymbol{g}\boldsymbol{e}_d}{\partial t} + \frac{\partial}{\partial x_d}\boldsymbol{g}\boldsymbol{u} = \left[\boldsymbol{u}\times(\nabla\times\boldsymbol{g}^T)\right]^T\boldsymbol{e}_d. \tag{7}$$

The PDE for $\boldsymbol{g}$ and its right hand side are described in detail in [23]. Here a is an arbitrary constant with dimensions of velocity. As developed in [30], this fictitious wave speed may be treated as a parameter that affects the partitioning of hyperbolic and elliptic terms. Through proper choice of that parameter, the CFL limiting time step of the hyperbolic partition can be improved by several orders of magnitude in the Newtonian limit ($\lambda \to 0$). Here, for purposes of analysis, a need only satisfy $\min_d(\rho a^2 + \tau_{dd}) > 0$.

All together, the coupled PDEs (1)-(3) may be written in the form

$$\frac{\partial \boldsymbol{U}}{\partial t} + \frac{\partial \boldsymbol{F}_\alpha}{\partial x_\alpha} = \boldsymbol{S}_h(\boldsymbol{U}) + \boldsymbol{S}_i(\boldsymbol{U}, \nabla\boldsymbol{U}, \Delta\boldsymbol{U}), \tag{8}$$

where the left hand side is a system of conservation laws, and the right hand side contains proper hyperbolic source terms, $\boldsymbol{S}_h$, and improper (elliptic) source terms, $\boldsymbol{S}_i$. $\boldsymbol{U}$ is the vector of conserved quantities:

$$\boldsymbol{U} = \left(\boldsymbol{u},\ \boldsymbol{M},\ \boldsymbol{g}\boldsymbol{e}_0, \ldots, \boldsymbol{g}\boldsymbol{e}_{D-1}\right)^T, \tag{9}$$

$$\boldsymbol{F}_d = \left(u_d\boldsymbol{u} - \frac{1}{\rho}\boldsymbol{\tau}\boldsymbol{e}_d,\ u_d\boldsymbol{M},\ \boldsymbol{g}\boldsymbol{u}\delta_{0d}, \ldots, \boldsymbol{g}\boldsymbol{u}\delta_{D-1,d}\right)^T, \tag{10}$$

$$\boldsymbol{S}_h = \left(-\frac{1}{\rho}\nabla p,\ -\frac{1}{\lambda}\boldsymbol{g}\boldsymbol{\tau}\boldsymbol{g}^T,\ 0, \ldots, 0\right)^T, \tag{11}$$

$$\begin{aligned}\boldsymbol{S}_i = \Bigg(&\nu_s\Delta\boldsymbol{u},\ 2\Big(\frac{\mu_p}{\lambda} - \rho a^2\Big)\boldsymbol{g}\boldsymbol{D}\boldsymbol{g}^T,\\ &\left[\boldsymbol{u}\times(\nabla\times\boldsymbol{g}^T)\right]^T\boldsymbol{e}_0, \ldots, \left[\boldsymbol{u}\times(\nabla\times\boldsymbol{g}^T)\right]^T\boldsymbol{e}_{D-1}\Bigg)^T,\end{aligned} \tag{12}$$

where $D = 2$ is the dimensionality of the problem and $\nu = \mu/\rho$.

We analyze the hyperbolic subsystem in primitive variables, $\boldsymbol{W}$. The linearization of (8) in primitive variables gives matrices whose eigenvalues are wave speeds, and whose eigenvectors determine the characteristics. If, in the 1D analysis of these linearized equations for direction d, primitive variable u_d is included, then wave speeds and characteristics describing compressive wave motion are observed. Yet, omission of u_d and its corresponding stress τ_{dd} is also inaccurate [9] since variation in these quantities is permitted by the multidimensional equations. The approach to this dilemma, after [9; 8] is to block partition the primitive equations, treating dependence on gradients of the variables u_d and τ_{dd} as source terms from the point of view of the remaining variables. We will refer to the variable partition $(u_d,\ \tau_{dd})$ as *inactive* (subscript I), and the remaining variable partition as *active* (subscript A). For $d = 0$,

$$\boldsymbol{W}_0^T = \left(\boldsymbol{W}_{A,0}^T \mid \boldsymbol{W}_{I,0}^T\right) = (u_1,\ \tau_{10},\ \tau_{11},\ g_{00},\ g_{10},\ g_{01},\ g_{11} \,|\, u_0, \tau_{00}). \tag{13}$$

The primitive variable τ_{01} is omitted because $\boldsymbol{\tau}$ is symmetric. In these variables, the linearized homogeneous advection equation in direction $d = 0$ is

$$\frac{\partial \boldsymbol{W}_0}{\partial t} + \boldsymbol{A}_0 \frac{\partial \boldsymbol{W}_0}{\partial x_0} = 0, \tag{14}$$

$$\boldsymbol{A}_0 = \left[\begin{array}{c|c} \boldsymbol{A}_{AA,0} & \boldsymbol{A}_{AI,0} \\ \hline \boldsymbol{A}_{IA,0} & \boldsymbol{A}_{II,0} \end{array}\right]$$
$$= \left[\begin{array}{ccccccc|cc} u_0 & -1/\rho & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -\rho c_0^2 & u_0 & 0 & 0 & 0 & 0 & 0 & -\tau_{10} & 0 \\ -2\tau_{10} & 0 & u_0 & 0 & 0 & 0 & 0 & 0 & 0 \\ g_{01} & 0 & 0 & u_0 & 0 & 0 & 0 & g_{00} & 0 \\ g_{11} & 0 & 0 & 0 & u_0 & 0 & 0 & g_{10} & 0 \\ 0 & 0 & 0 & 0 & 0 & u_0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & u_0 & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & 0 & 0 & 0 & u_0 & -1/\rho \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -2\rho c_0^2 & u_0 \end{array}\right], \tag{15}$$

with $c_d = \sqrt{a^2 + \tau_{dd}/\rho}$. The diagonal matrix of eigenvalues of partition $\boldsymbol{A}_{AA,0}$ is

$$\boldsymbol{\Lambda}_0 = \operatorname{diag}\left(u_0 - c_0,\ u_0,\ u_0,\ u_0,\ u_0,\ u_0,\ u_0 + c_0\right)^T. \tag{16}$$

The corresponding right eigenvectors are given by the columns of

$$\boldsymbol{R}_0 = \begin{bmatrix} -c_0 & 0 & 0 & 0 & 0 & 0 & c_0 \\ -\rho c_0^2 & 0 & 0 & 0 & 0 & 0 & -\rho c_0^2 \\ -2\tau_{10} & 1 & 0 & 0 & 0 & 0 & -2\tau_{10} \\ g_{01} & 0 & 1 & 0 & 0 & 0 & g_{01} \\ g_{11} & 0 & 0 & 1 & 0 & 0 & g_{11} \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}. \tag{17}$$

2.0.1. *An exact Riemann solver.* For the incompressible Euler equations, Bell et al. [2] construct edge-centered time-centered predictor states using Taylor series with upwind derivatives. For those equations, their approach is identical to using a higher-order Godunov predictor because *upwinding* solves exactly the associated

Riemann problem. In the present system of equations, the wave structure is more complex, but there are no genuinely nonlinear waves, that is,

$$\left(\nabla_{\boldsymbol{W}_A} \Lambda_{kk}\right) \cdot \boldsymbol{Re}_k = 0, \tag{18}$$

for each of the 7 waves k associated with block A_{AA} of (15). This condition is guaranteed by the fact that the complete solution for the inactive variables is taken to be the average of the input left and right states [8], and therefore the eigenvalues are constant with respect to each component of $\boldsymbol{W}_A$.

By analysis of the generalized Riemann invariants,

$$\frac{\partial(\boldsymbol{W}_A)_0}{\boldsymbol{e}_0^T \boldsymbol{Re}_k} = \frac{\partial(\boldsymbol{W}_A)_1}{\boldsymbol{e}_1^T \boldsymbol{Re}_k} = \cdots \frac{\partial(\boldsymbol{W}_A)_6}{\boldsymbol{e}_6^T \boldsymbol{Re}_k}, \tag{19}$$

for each wave k, it may be concluded (assuming for convenience $d = 0$) that

(i) u_1 and τ_{10} are constant across the 5 contact (speed u_0) waves;

(ii) g_{01} and g_{11} are constant across the fast $u_0 \pm c_0$ waves;

(iii) the generalized Riemann invariants for the ± fast waves include the identities

$$\frac{\partial u_1}{\pm c_0} = \frac{\partial \tau_{10}}{-\rho c_0^2} = \frac{\partial g_{00}}{g_{01}} = \frac{\partial g_{10}}{g_{11}}, \tag{20}$$

where the denominators of each term are constant across the wave. Thus, across each fast wave the change in u_1 is proportional to c_0, etc.;

(iv) across the fast waves, the generalized Riemann invariants contain also

$$\frac{\partial \tau_{11}}{-2\tau_{10}}. \tag{21}$$

So, given the change of τ_{10} across the given wave, the change in τ_{11} is determined.

Let the constant states in the Riemann fan be labeled $\boldsymbol{W}_L$, $\boldsymbol{W}_{L^*}$, $\boldsymbol{W}_{R^*}$, and $\boldsymbol{W}_R$ in sequence, and let Ψ_L (Ψ_R) measure the strength of the left (right) fast waves. From observation (iii) one has

$$\begin{aligned} \begin{pmatrix} u_1 \\ \tau_{10} \end{pmatrix}_{L^*} &= \begin{pmatrix} u_1 \\ \tau_{10} \end{pmatrix}_L - \Psi_L \begin{pmatrix} c_0 \\ \rho c_0^2 \end{pmatrix}, \\ \begin{pmatrix} u_1 \\ \tau_{10} \end{pmatrix}_{R^*} &= \begin{pmatrix} u_1 \\ \tau_{10} \end{pmatrix}_R + \Psi_R \begin{pmatrix} c_0 \\ -\rho c_0^2 \end{pmatrix}, \end{aligned} \tag{22}$$

and from observation (i) one has $(u_1,\ \tau_{10})_{L^*} = (u_1,\ \tau_{10})_{R^*}$, which couples the fast waves enabling their strength to be simply determined from

$$\begin{pmatrix} c_0 & c_0 \\ \rho c_0^2 & -\rho c_0^2 \end{pmatrix} \begin{pmatrix} \Psi_L \\ \Psi_R \end{pmatrix} = \begin{pmatrix} u_1 \\ \tau_{10} \end{pmatrix}_L - \begin{pmatrix} u_1 \\ \tau_{10} \end{pmatrix}_R . \tag{23}$$

With τ_{10} determined across the wave fan, observation (iv) determines τ_{11}:

$$\int_{(\tau_{11})_L}^{(\tau_{11})_{L^*}} d\tau_{11} = \frac{2}{\rho c_0^2} \int_{(\tau_{10})_L}^{(\tau_{10})_{L^*}} \tau_{10} d\tau_{10}, \tag{24}$$

$$(\tau_{11})_{L^*} = (\tau_{11})_L + \frac{1}{\rho c_0^2} \left[(\tau_{10})_{L^*}^2 - (\tau_{10})_L^2 \right]. \tag{25}$$

The same equation holds across the right fast wave. The determination of other variables is then trivial by application of observation (iii). For example, from (20),

$$\frac{(g_{00})_{R^*} - (g_{00})_R}{(g_{01})_{R^*}} = \frac{(u_1)_{R^*} - (u_1)_R}{c_0}. \tag{26}$$

The active variable solution to our Riemann problem is given by the constant state (L, L^*, R^*, or R) containing the zero wave speed characteristic.

3. Predictor-corrector formulation

We discretize time in steps Δt, with $t^{n+1} = t^n + \Delta t^n$. Space is discretized in square cells of length h, and $\boldsymbol{x} = h\boldsymbol{i}$ is the lower left corner of cell $\boldsymbol{i}$. Variables $\boldsymbol{U}_{\boldsymbol{i}}^n$ are cell-centered.

For each time step n, the artificial wave speed a is a global constant determined by the heuristic model:

$$a^2 = \min\left\{ \chi(\lambda) a_\infty^2 + [1 - \chi(\lambda)]\, a_0^2,\ \frac{\nu_p}{\lambda} \right\}, \tag{27}$$

$$a_\infty^2 = \frac{\nu_p}{\lambda}, \tag{28}$$

$$a_0^2 = \frac{2}{\rho} \min_{\boldsymbol{i},d} |(\tau_{dd})_{\boldsymbol{i}}|, \tag{29}$$

$$\chi(\lambda) = \frac{\lambda}{t_{\text{adv}}} \left[1 - e^{-\lambda/(2t_{\text{adv}})} \right] \left(1 - e^{-t_{\text{adv}}/\lambda} \right), \tag{30}$$

$$t_{\text{adv}} = \frac{h}{\max_{\boldsymbol{i}} |\boldsymbol{u}|}, \tag{31}$$

with limiting values $a^2 = a_\infty^2$ as $\lambda \to \infty$, and $a^2 = a_0^2$ as $\lambda \to 0$. Note that the conserved quantity $\boldsymbol{M}$ depends on a, so a reevaluation of a necessitates a rescaling of $\boldsymbol{M}$ throughout the domain.

The predictor step of the method uses well-established higher-order Godunov approaches [5; 6] to estimate time-centered edge-centered solution values. These predictor states are made discrete divergence-free ($\nabla \cdot \boldsymbol{u}^{n+\frac{1}{2}} = 0$) on a marker-and-cell (MAC) stencil [15].

Fluxes $\boldsymbol{F}^{n+\frac{1}{2}}_{\boldsymbol{i}\pm\boldsymbol{e}/2} = \boldsymbol{F}\big(\boldsymbol{U}^{n+\frac{1}{2}}_{\boldsymbol{i}\pm\boldsymbol{e}/2}\big)$ computed from these predictor states enter a conservative update:

$$\tilde{\boldsymbol{U}}^{n+1}_{\boldsymbol{i}} = \boldsymbol{U}^{n}_{\boldsymbol{i}} - \frac{\Delta t}{h}\sum_{d=0}^{D-1}\Big[\big(\boldsymbol{F}_d\big)^{n+\frac{1}{2}}_{\boldsymbol{i}+\boldsymbol{e}_d/2} - \big(\boldsymbol{F}_d\big)^{n+\frac{1}{2}}_{\boldsymbol{i}-\boldsymbol{e}_d/2}\Big]. \tag{32}$$

The corrector computes $\boldsymbol{U}^{n+1}$ by adding to $\tilde{\boldsymbol{U}}^{n+1}$ implicit and explicit source term contributions, and by use of an approximate cell-centered projection to make $\boldsymbol{u}^{n+1}$ discrete divergence-free.

3.1. ***Predictor.*** The predictor in our predictor-corrector method consists of the calculation of time-centered edge-centered states, $\boldsymbol{W}^{n+1/2}_{\boldsymbol{i}+\boldsymbol{e}_d/2}$, which are discrete divergence-free. The predictor state is computed in four steps.

First, the one-dimensional primitive equations are used to estimate time-centered edge-centered states. For the active partition, characteristic tracing and slope limiting occur as in higher-order Godunov methods. For the inactive partition, Taylor series in space and time with centered differences are used. This first step uses strictly one-dimensional equations with no transverse coupling.

Second, the edge states so obtained are double-valued, and we resolve these with the Riemann solver described above in Section 2.0.1.

Third, the transverse coupling omitted in the first step is incorporated using cell-centered gradients of the edge-centered states computed by the Riemann solution. The transverse flux correction is described in [5; 28], but we include the transverse terms in terms of primitive variable differences rather than conservative fluxes. The corrected states so-obtained are again double-valued, and another Riemann problem gives a single final result.

Fourth, time-centered edge-centered velocity data is made discrete divergence-free, i.e.,

$$\boldsymbol{u} := \boldsymbol{u} - \nabla\left[\Delta^{-1}(\nabla\cdot\boldsymbol{u})\right]. \tag{33}$$

With $\boldsymbol{u}$ edge-centered, $\nabla\cdot\boldsymbol{u}$ is cell-centered. This projection is exact, in the sense that $\Delta_h = (\nabla\cdot)_h\nabla_h$, with the discrete Laplacian reducing to the standard 5-point stencil in two dimensions away from boundaries. Then, $\Delta^{-1}(\nabla\cdot\boldsymbol{u})$ is also cell-centered. The discrete gradient operator uses centered divided differences to give edge-centered corrections. The normal and tangential velocity components are updated at each face even though only the normal velocity contributes to the divergence.

The details of the first step is now given. With the *active–inactive* partitioning introduced in (13), upwind characteristic tracing for the active primitive variables takes the form

$$\begin{aligned}
\big(\tilde{\boldsymbol{W}}_{A,d}\big)^{n+\frac{1}{2}}_{\boldsymbol{i}+\boldsymbol{e}_d/2,L} &= \big(\boldsymbol{W}_{A,d}\big)^n_{\boldsymbol{i}} - \boldsymbol{R}_d \mathscr{P}_+\left(\frac{\Delta t}{2}\boldsymbol{\Lambda}_d - \frac{h}{2}\boldsymbol{I}\right)\boldsymbol{R}_d^{-1}\left(\frac{\partial \boldsymbol{W}_{A,d}}{\partial x_d}\right)^n_{\boldsymbol{i}} \\
&\qquad\qquad - \frac{\Delta t}{2h}\boldsymbol{A}_{AI,d}\left(\frac{\partial \boldsymbol{W}_{I,d}}{\partial x_d}\right)^n_{\boldsymbol{i}}, \\
\big(\tilde{\boldsymbol{W}}_{A,d}\big)^{n+\frac{1}{2}}_{\boldsymbol{i}+\boldsymbol{e}_d/2,R} &= \big(\boldsymbol{W}_{A,d}\big)^n_{\boldsymbol{i}+\boldsymbol{e}_d} - \boldsymbol{R}_d \mathscr{P}_-\left(\frac{\Delta t}{2}\boldsymbol{\Lambda}_d + \frac{h}{2}\boldsymbol{I}\right)\boldsymbol{R}_d^{-1}\left(\frac{\partial \boldsymbol{W}_{A,d}}{\partial x_d}\right)^n_{\boldsymbol{i}+\boldsymbol{e}_d} \\
&\qquad\qquad - \frac{\Delta t}{2h}\boldsymbol{A}_{AI,d}\left(\frac{\partial \boldsymbol{W}_{I,d}}{\partial x_d}\right)^n_{\boldsymbol{i}+\boldsymbol{e}_d},
\end{aligned} \tag{34}$$

where $\mathscr{P}_\pm(\mathscr{D}) = \mathrm{diag}(\mathscr{D}_{ii} \text{ if } \Lambda_{ii} \gtrless 0,\ 0 \text{ otherwise})$ is a projection that sets to zero those terms of the diagonal argument matrix corresponding with eigenvalues whose sign is negative/positive, respectively. The subscript L (R) indicates that the result is traced to the left (right) side of the edge $\boldsymbol{i} + \boldsymbol{e}_d/2$. Where the stencils support it, the derivatives $\partial \boldsymbol{W}_A/\partial x$ use van Leer limited [34] fourth-order accurate derivatives [4]. The derivatives $\partial \boldsymbol{W}_I/\partial x$ use second-order centered divided differences. The tilde denotes that source terms have not yet been accounted for. The inactive variables are extrapolated in time using

$$\begin{aligned}
\big(\tilde{\boldsymbol{W}}_{I,d}\big)^{n+\frac{1}{2}}_{\boldsymbol{i}+\boldsymbol{e}_d/2,L} &= \big(\tilde{\boldsymbol{W}}_{I,d}\big)^n_{\boldsymbol{i}} - \left(\frac{\Delta t}{2}\boldsymbol{A}_{II,d} - \frac{h}{2}\boldsymbol{I}\right)\left(\frac{\partial \boldsymbol{W}_{I,d}}{\partial x_d}\right)^n_{\boldsymbol{i}} \\
&\qquad\qquad - \frac{\Delta t}{2}\boldsymbol{A}_{IA,d}\left(\frac{\partial \boldsymbol{W}_{A,d}}{\partial x_d}\right)^n_{\boldsymbol{i}}, \\
\big(\tilde{\boldsymbol{W}}_{I,d}\big)^{n+\frac{1}{2}}_{\boldsymbol{i}+\boldsymbol{e}_d/2,R} &= \big(\tilde{\boldsymbol{W}}_{I,d}\big)^n_{\boldsymbol{i}+1} - \left(\frac{\Delta t}{2}\boldsymbol{A}_{II,d} + \frac{h}{2}\boldsymbol{I}\right)\left(\frac{\partial \boldsymbol{W}_{I,d}}{\partial x_d}\right)^n_{\boldsymbol{i}+\boldsymbol{e}_d} \\
&\qquad\qquad - \frac{\Delta t}{2}\boldsymbol{A}_{IA,d}\left(\frac{\partial \boldsymbol{W}_{A,d}}{\partial x_d}\right)^n_{\boldsymbol{i}+\boldsymbol{e}_d}.
\end{aligned} \tag{35}$$

The velocity source is computed explicitly via

$$\boldsymbol{u}^{n+\frac{1}{2}}_{\boldsymbol{i}+\boldsymbol{e}_d/2,L} = \tilde{\boldsymbol{u}}^{n+\frac{1}{2}}_{\boldsymbol{i}+\boldsymbol{e}_d/2,L} + \frac{\Delta t}{2}\Big(-\frac{1}{\rho}\nabla p^{n-\frac{1}{2}}_{\boldsymbol{i}} + \nu_s(\Delta_h \boldsymbol{u}^n)_{\boldsymbol{i}}\Big), \tag{36}$$

where Δ_h the discrete 5-point Laplacian in regular domains. The time-centered pressure is taken from the previous time step. The calculation of $\nabla p^{n+\frac{1}{2}}$ occurs as the last step of the corrector, (43).

The source term for viscoelastic stress is computed implicitly to properly recover the Newtonian limit ($\boldsymbol{\tau} \to 2\mu_p \boldsymbol{D}$ as $\lambda \to 0$):

$$\boldsymbol{\tau}^{n+\frac{1}{2}}_{\boldsymbol{i}+\boldsymbol{e}_d/2,L} = \tilde{\boldsymbol{\tau}}^{n+\frac{1}{2}}_{\boldsymbol{i}+\boldsymbol{e}_d/2,L} + \frac{\Delta t}{2}\left[-\frac{1}{\lambda}\boldsymbol{\tau}^{n+\frac{1}{2}}_{\boldsymbol{i}+\boldsymbol{e}_d/2,L} + \Big(\frac{\mu_p}{\lambda} - \rho a^2\Big)2\boldsymbol{D}^n_{\boldsymbol{i}}\right]. \tag{37}$$

The rate of strain tensor, $\boldsymbol{D}$, is calculated with centered differences.

The source terms for $\boldsymbol{g}$ are omitted for the following reason. The material reference frame $\boldsymbol{X}$ can be defined, at the start of each time step, to be equal to $\boldsymbol{x}$, i.e., $\boldsymbol{g} = \boldsymbol{I}$ identically at the start of each time step. With this choice, the source terms for $\boldsymbol{g}$ are zero if evaluated at t^n. Resetting $\boldsymbol{g}$ to $\boldsymbol{I}$ necessitates renormalizing $\boldsymbol{M}$ from time step to time step.

3.2. ***Corrector.*** The corrector generates time $n+1$ cell-centered states that are discrete divergence-free. The basic idea is to generate cell-centered time t^{n+1} estimates, $\tilde{\boldsymbol{U}}^{n+1}$, using the flux differencing quadrature (32). To these estimates source terms are added, as described below, to obtain $\boldsymbol{U}^{n+1}$.

The corrector step for the velocity field is more complicated. We would like to use the following update equation (the superscript $*$ indicates that the velocity field is not yet divergence-free):

$$\frac{\boldsymbol{u}^{n+1,*} - \boldsymbol{u}^n}{\Delta t} = \left[-\nabla \cdot \left(\boldsymbol{u} \otimes \boldsymbol{u} - \frac{1}{\rho} \boldsymbol{\tau} \right)^{n+\frac{1}{2}} \right] + \left(-\frac{1}{\rho} \nabla p^{n-\frac{1}{2}} + \nu_s \Delta \boldsymbol{u} \right). \tag{38}$$

However, as in [30], we would like for the velocity update equation to properly capture the Newtonian and elastic limits. We modify the predictor step by not including the source terms for $\boldsymbol{\tau}$ in the edge state prediction to instead obtain $\tilde{\boldsymbol{\tau}}$ at edges. However, extra care must be taken since the transverse correction term is still computed with edge states that have been constructed with the $\boldsymbol{\tau}$ sources.

Combining an equation of the form (37) with (38), we arrive at our new update equation for velocity:

$$\frac{\boldsymbol{u}^{n+1,*} - \boldsymbol{u}^n}{\Delta t} = \left[\nu_s + \frac{\Delta t (\nu_p - \lambda a^2)}{2\lambda + \Delta t} \right] \Delta \boldsymbol{u} + \left[-\nabla \cdot \left(\boldsymbol{u} \otimes \boldsymbol{u} - \frac{2\lambda}{2\lambda + \Delta t} \frac{\tilde{\boldsymbol{\tau}}}{\rho} \right)^{n+\frac{1}{2}} - \frac{1}{\rho} \nabla p^{n-\frac{1}{2}} \right]. \tag{39}$$

These equations are expressible as D scalar discrete Helmholtz equations. This discretization is chosen in order to capture the Newtonian and elastic limits, that is, in the Newtonian limit ($\lambda \to 0$) we recover

$$\frac{\boldsymbol{u}^{n+1,*} - \boldsymbol{u}^n}{\Delta t} = [\nu_s + \nu_p] \Delta \boldsymbol{u} + \left[-\nabla \cdot (\boldsymbol{u} \otimes \boldsymbol{u})^{n+\frac{1}{2}} - \frac{1}{\rho} \nabla p^{n-\frac{1}{2}} \right], \tag{40}$$

and in the elastic limit ($\lambda \to \infty$), where a^2 is given by (28), we recover

$$\frac{\boldsymbol{u}^{n+1,*} - \boldsymbol{u}^n}{\Delta t} = \nu_s \Delta \boldsymbol{u} + \left[-\nabla \cdot \left(\boldsymbol{u} \otimes \boldsymbol{u} - \frac{\tilde{\boldsymbol{\tau}}}{\rho} \right)^{n+\frac{1}{2}} - \frac{1}{\rho} \nabla p^{n-\frac{1}{2}} \right]. \tag{41}$$

The Helmholtz equations (39) are solved using the Runge–Kutta technique of [32], which yields an l_0 stable solution in regular and irregular domains. That method specifies the time centering of the Laplacian term.

The last step of the velocity corrector removes the divergence of $\boldsymbol{u}^*$ and calculates the pressure whose gradient will affect the subsequent time step using a pressure-projection formulation [31]. First, a potential ϕ is calculated on cell centers with the discrete Laplacian:

$$\Delta\phi = \left[\nabla \cdot \mathrm{Avg}\left(\boldsymbol{u}^{n+1,*} + \frac{\Delta t}{\rho}\nabla p^{n-\frac{1}{2}}\right)\right], \tag{42}$$

where Avg is an operator that computes face-centered values by averaging neighboring cell-centered values. Pressure is proportional to ϕ, and

$$\nabla p^{n+\frac{1}{2}} = \frac{\rho}{\Delta t}\nabla\phi. \tag{43}$$

With this gradient, the discrete-divergence-free velocity is

$$\boldsymbol{u}^{n+1} = \boldsymbol{u}^{n+1,*} + \frac{\Delta t}{\rho}\left(\nabla p^{n-\frac{1}{2}} - \nabla p^{n+\frac{1}{2}}\right). \tag{44}$$

This projection is approximate, in the sense that $\Delta_h \neq (\nabla\cdot)_h\nabla_h$. As noted by Lai [20], the approximate projection does not remove certain nonphysical oscillatory modes. These are damped by application of a filter

$$\boldsymbol{u} := \boldsymbol{u} + \zeta\nabla(\nabla\cdot\boldsymbol{u}), \tag{45}$$

using a divergence stencil other than the centered divided difference used in (42). We use $\zeta = h^2/5$ in two dimensions which is stable while always damping monopole modes in the experience of [8; 30].

The corrector step for $\boldsymbol{g}$ and $\boldsymbol{M}$ simply follows the flux differencing quadrature (32) followed by a source term update. The source term for $\boldsymbol{g}$ is computed as in [23] using edge — and time — centered values from the predictor. The viscoelastic stress source term is discretized using Crank–Nicholson:

$$\boldsymbol{M}^{n+1} = \tilde{\boldsymbol{M}}^{n+1} + \frac{\Delta t}{2}\left(\boldsymbol{g}\left[\left(\frac{\mu_p}{\lambda} - \rho a^2\right)2\boldsymbol{D} - \frac{1}{\lambda}\boldsymbol{\tau}\right]\boldsymbol{g}^T\right)^n + \frac{\Delta t}{2}\left(\boldsymbol{g}\left[\left(\frac{\mu_p}{\lambda} - \rho a^2\right)2\boldsymbol{D} - \frac{1}{\lambda}\boldsymbol{\tau}\right]\boldsymbol{g}^T\right)^{n+1},$$

rearranged in the form

$$\begin{aligned}\boldsymbol{M}^{n+1} = \frac{2\lambda}{2\lambda+\Delta t}\tilde{\boldsymbol{M}}^{n+1} - \frac{\Delta t}{2\lambda+\Delta t}\boldsymbol{M}^n &+ \frac{\Delta t}{2\lambda+\Delta t}\left(\boldsymbol{g}\left[(\mu_p - \rho a^2\lambda)\,2\boldsymbol{D} + \rho a^2\,\boldsymbol{I}\right]\boldsymbol{g}^T\right)^n \\ &+ \frac{\Delta t}{2\lambda+\Delta t}\left(\boldsymbol{g}\left[(\mu_p - \rho a^2\lambda)\,2\boldsymbol{D} + \rho a^2\,\boldsymbol{I}\right]\boldsymbol{g}^T\right)^{n+1},\end{aligned} \tag{46}$$

which is evaluated pointwise.

4. Irregular domains

We use a Cartesian grid embedded boundary method to discretize the fluid equations in the presence of irregular boundaries [6]. In this approach, the irregular domain is discretized as a collection of control volumes formed by the intersection of the problem domain with the square Cartesian grid cells as in a "cookie cutter". The various operators — the discrete divergence $\nabla\cdot$, discrete gradient ∇, and discrete Laplacian Δ — are approximated using finite volume differences on the irregular control volumes. Cells are classified as regular if they do not intersect embedded boundaries, irregular if they intersect boundaries, or covered if they have zero fluid volume fraction. Faces are classified in an analogous way. In problems containing irregular domains, the finite volume treatment of the regular cells follows the description of Section 3.

Throughout, time t^n data ($\boldsymbol{U}$) will be centered at cell centers, even if that point lies outside the fluid domain. Time $t^{n+\frac{1}{2}}$ data (fluxes $\boldsymbol{F}$) are centered at the centroid of faces,

$$\hat{\boldsymbol{x}}_{\boldsymbol{i}\pm\boldsymbol{e}_d/2} = \frac{1}{\alpha_{\boldsymbol{i}\pm\boldsymbol{e}_d/2} h^{D-1}} \int_{A_{\boldsymbol{i}\pm\boldsymbol{e}_d/2}} \boldsymbol{x}\, dA, \tag{47}$$

where $\alpha_{\boldsymbol{i}\pm\boldsymbol{e}_d/2}$ is the area fraction of a cell edge $\boldsymbol{i}\pm\boldsymbol{e}_d/2$ not covered by the embedded boundary, or

$$\alpha_{\boldsymbol{i}\pm\boldsymbol{e}_d/2} = \frac{A_{\boldsymbol{i}\pm\boldsymbol{e}_d/2}}{h^{D-1}}, \tag{48}$$

with $A_{\boldsymbol{i}\pm\boldsymbol{e}_d}$ the area of cell $\boldsymbol{i}$ on side $\pm d$ in contact with the fluid. Other geometric quantities used are the volume fraction, defined as

$$\kappa_{\boldsymbol{i}} = \frac{V_{\boldsymbol{i}}}{h^D}, \tag{49}$$

the area fraction of the domain boundary intersected with cell $\boldsymbol{i}$, $A_{\boldsymbol{i}}^{\mathrm{EB}}$, and its associated area fraction, defined as

$$\alpha_{\boldsymbol{i}}^{\mathrm{EB}} = \frac{A_{\boldsymbol{i}}^{\mathrm{EB}}}{h^{D-1}}, \tag{50}$$

and the outward-directed vector normal to the embedded boundary interface in cell $\boldsymbol{i}$, given by

$$\boldsymbol{n}_{\boldsymbol{i}} = \frac{1}{\alpha_{\boldsymbol{i}}^{\mathrm{EB}} h^{D-1}} \int_{A_{\boldsymbol{i}}^{\mathrm{EB}}} \boldsymbol{n}\, dA. \tag{51}$$

In irregular cells, the quadrature (32) is not appropriate [6]. A stable but nonconservative update is

$$\tilde{\boldsymbol{U}}_{\boldsymbol{i}}^{n+1} = \boldsymbol{U}_{\boldsymbol{i}}^{n} - \Delta t\big[\kappa_{\boldsymbol{i}}(\nabla\cdot\boldsymbol{F})_{\boldsymbol{i}}^{\mathrm{C}} + (1-\kappa_{\boldsymbol{i}})(\nabla\cdot\boldsymbol{F})_{\boldsymbol{i}}^{\mathrm{NC}}\big]^{n+\frac{1}{2}}, \tag{52}$$

with conservative and nonconservative flux differences given by

$$(\nabla\cdot\boldsymbol{F})_{\boldsymbol{i}}^{\mathrm{NC}} = \frac{1}{h}\sum_{d=0}^{D-1}\big[(\boldsymbol{F}_d)_{\boldsymbol{i}+\boldsymbol{e}_d/2} - (\boldsymbol{F}_d)_{\boldsymbol{i}-\boldsymbol{e}_d/2}\big], \tag{53}$$

$$\begin{aligned}(\nabla\cdot\boldsymbol{F})_{\boldsymbol{i}}^{\mathrm{C}} &= \frac{1}{V_{\boldsymbol{i}}}\int_{V_i}(\nabla\cdot\boldsymbol{F})\,dV \\ &\approx \frac{1}{\kappa_{\boldsymbol{i}}h}\left[\sum_{d=0}^{D-1}\sum_{\pm}\big[\pm\alpha_{\boldsymbol{i}\pm\boldsymbol{e}_d/2}\boldsymbol{F}_d(\hat{\boldsymbol{x}}_{\boldsymbol{i}\pm\boldsymbol{e}_d/2})\big] + \alpha_{\boldsymbol{i}}^{\mathrm{EB}}(\boldsymbol{n}_{\boldsymbol{i}}\cdot\boldsymbol{F}_{\boldsymbol{i}}^{\mathrm{EB}})\right],\end{aligned} \tag{54}$$

respectively.

Conservation violation is expressed locally by the generalized mass deficit $\delta\boldsymbol{m}$,

$$\delta\boldsymbol{m}_{\boldsymbol{i}} = \Delta t(1-\kappa_{\boldsymbol{i}})\kappa_{\boldsymbol{i}}\,\big[(\nabla\cdot\boldsymbol{F})^{\mathrm{NC}} - (\nabla\cdot\boldsymbol{F})^{\mathrm{C}}\big]_{\boldsymbol{i}}, \tag{55}$$

which is redistributed in a volume-weighted manner according to

$$\tilde{\boldsymbol{U}}_{\boldsymbol{i}}^{n+1} := \tilde{\boldsymbol{U}}_{\boldsymbol{i}}^{n+1} + \sum_{\boldsymbol{j}=\mathrm{neighbor}(\boldsymbol{i})}^{3^D}\frac{\delta\boldsymbol{m}_{\boldsymbol{j}}}{w_{\boldsymbol{j}}}, \tag{56}$$

$$w_{\boldsymbol{i}} = \sum_{\boldsymbol{j}=\mathrm{neighbor}(\boldsymbol{i})}^{3^D}\kappa_{\boldsymbol{j}}. \tag{57}$$

The calculation of fluxes on covered faces, and stencils used to re-center fluxes to centroids, are described in [6; 22; 29]. Additional details are given in [24]. Here we describe differences between the regular and irregular domain calculations that are specific to the present algorithm.

We compute the Poisson equation in divergence form, $\Delta\phi \approx \nabla^h\cdot(\nabla^h\phi) = f$, with discrete divergence given by the conservative form (54). This means that $\kappa\Delta^h\phi$ is directly accessible, and division by κ can be unstable. For the Laplacian appearing in the velocity source term (36) we use $\kappa\Delta^h\phi$ in place of $\Delta^h\phi$, which formally introduces an $\mathbb{O}(\Delta t)$ discretization error. However, the results obtained by this approximation are stable and appear to not affect the global error.

In irregular cells the discretization of the divergence term in (39) is computed as follows. Define a velocity flux to be

$$\boldsymbol{F}_{\boldsymbol{u}} = \boldsymbol{u}\otimes\boldsymbol{u} - \frac{2\lambda}{2\lambda+\Delta t}\frac{\tilde{\boldsymbol{\tau}}}{\rho}. \tag{58}$$

Then, compute the divergence of $\boldsymbol{F_u}$ using (52) and redistribute according to (56).

Covered face values needed in the nonconservative divergence are obtained by extrapolation from face-centered time-centered values, as described in [6]. Unlike [6], we take this extrapolated edge state to represent the unique face value, so no further Riemann problem is solved.

5. Boundary conditions

In the hyperbolic treatment, boundary conditions enter in two ways:

(1) on embedded boundaries, e.g., the computation of $\boldsymbol{F}^{EB}$ in (54); and

(2) where the Cartesian cells abut the problem domain.

The conservative flux divergence (54) includes the flux derived from data centered at the centroid of the embedded boundary. Such states are derived from cell-centered data using Taylor series, without upwind projection. If $\hat{\boldsymbol{x}}_{\boldsymbol{i}}^{\mathrm{EB}}$ is the centroid relative to the cell center,

$$\begin{aligned}\boldsymbol{W}_{\boldsymbol{i}}^{n+\frac{1}{2},\mathrm{EB}} &= \boldsymbol{W}_{\boldsymbol{i}}^{n} + \hat{\boldsymbol{x}}_{\boldsymbol{i}}^{\mathrm{EB}} \cdot (\nabla \boldsymbol{W}^{n})_{\boldsymbol{i}} + \frac{\Delta t}{2}\left(\frac{\partial \boldsymbol{W}_{\boldsymbol{i}}}{\partial t}\right)^{n} \\ &= \boldsymbol{W}_{\boldsymbol{i}}^{n} + \sum_{d}\left[\left(\hat{\boldsymbol{x}}_{\boldsymbol{i}}^{\mathrm{EB}}\right)_{d}\boldsymbol{I} - \frac{\Delta t}{2}\boldsymbol{A}_{d}\right]\left(\frac{\partial \boldsymbol{W}^{n}}{\partial x_{d}}\right)_{\boldsymbol{i}} + \frac{\Delta t}{2}\boldsymbol{S}_{\boldsymbol{i}}^{n}. \end{aligned} \tag{59}$$

This extrapolation is implemented without partitioning of $\boldsymbol{W}$ or $\boldsymbol{A}$. The source terms are implemented as with the predictor Section 3.1. The discrete gradient $\nabla\boldsymbol{W}$ uses central differences where possible, or one-sided differences where necessary.

This one-sided boundary value may be incompatible with physical boundary conditions. The approach to boundary conditions uses the ideas of Ghidaglia and Pascal [10]. Let $\boldsymbol{W}_P$ be an extrapolated edge state, as calculated by (59), and let $\boldsymbol{W}_S$ be the final value used to construct the edge flux. In appropriately rotated coordinates, we are interested in the eigenstructure of the matrix $\boldsymbol{A}_{AA}(\boldsymbol{W}_S)$. For each characteristic pointing into the domain, one degree of freedom at the boundary must be specified. For each characteristic pointing out of the domain, a characteristic condition must be met. Specifically, if characteristic k points out of the domain, a sufficient characteristic condition is

$$\boldsymbol{l}_k \cdot (\boldsymbol{W}_P - \boldsymbol{W}_S) = 0. \tag{60}$$

For solid wall boundaries, including the embedded boundaries, this construction is straightforward. We derive $\boldsymbol{A}_{AA}$ on the boundary using active variables taken from $\boldsymbol{W}_P$, and selecting inactive variables on physical grounds. In the present application, the embedded boundaries are stationary surfaces subject to no-flow conditions. Accordingly, the inactive variable u_n is uniquely determined, $u_n = 0$

(here subscript n denotes the interface normal direction; subscript t will denote the tangential direction). There is no a priori reason for τ_{nn} to be affected by boundary conditions, thus we take τ_{nn} in state $\boldsymbol{W}_S$ equal to its extrapolated value in $\boldsymbol{W}_P$. With these choices, exactly one characteristic of $\boldsymbol{A}_{AA}$ enters the domain, leaving one degree of freedom to be specified. We use the no-slip boundary condition to zero the tangential velocity component. For the characteristic that points out of the domain, the the characteristic condition $\boldsymbol{l}_{u_n-c} \cdot (\boldsymbol{W}_P - \boldsymbol{W}_S) = 0$ (if the wall normal is positive) or $\boldsymbol{l}_{u_n+c} \cdot (\boldsymbol{W}_P - \boldsymbol{W}_S) = 0$ (if the wall normal is negative) uniquely determines the shear stress τ_{nt} component of $\boldsymbol{W}_S$. Thus, for solid wall boundaries, $\boldsymbol{W}_S = \boldsymbol{W}_P$, except for variables $\boldsymbol{u}$ which are taken to be zero on physical grounds and the shear stress which is determined by the characteristic condition.

For inflow and outflow boundaries this procedure is more involved. Let $\boldsymbol{n}$ point out of the domain, so for inflow we have $u_n < 0$. Inflow conditions are either supersonic, $u_n + c < 0$, or not, with u_n and c given by inactive variables taken from the specified inflow condition. When supersonic, all characteristics flow into the domain, and the state $\boldsymbol{W}_S$ is given exclusively by imposed conditions. If not supersonic, only the characteristic $u_n + c$ flows out of the domain, so only one constraint on $\boldsymbol{W}_S$ comes from $\boldsymbol{W}_P$. In this case we determine the shear stress τ_{nt} component of $\boldsymbol{W}_S$ by solving $\boldsymbol{l}_{u_n+c} \cdot (\boldsymbol{W}_P - \boldsymbol{W}_S) = 0$, with all other components of $\boldsymbol{W}_S$ being prescribed by the inflow condition.

On outflow, we take the inactive variables from $\boldsymbol{W}_P$, and $u_n > 0$. If supersonic, $u_n - c > 0$, no characteristics flow into the fluid domain, and we take $\boldsymbol{W}_S = \boldsymbol{W}_P$. If subsonic, one degree of freedom of $\boldsymbol{W}_S$ is specified by external conditions. In that case, we choose $u_t = 0$ and determine the remaining values of $\boldsymbol{W}_S$ from $\boldsymbol{l}_k \cdot (\boldsymbol{W}_S - \boldsymbol{W}_P) = 0$, for all $k \neq u_n - c$.

Boundary conditions are also required for the Helmholtz velocity correctors, (39), and the divergence-cleaning projections (33) and (42) . The implicit velocity equations (39) use homogeneous Dirichlet conditions on solid wall boundaries, inhomogeneous Dirichlet conditions on inflow boundaries (using prescribed far-field values), and homogeneous Neumann conditions on outflow. The discrete Laplacian operator encountered in divergence-cleaning projections uses homogeneous Dirichlet on outflow, and homogeneous Neumann on inflow and solid walls.

6. Results

Results are presented for three fluids: a Maxwell (highly elastic) fluid, characterized by having no solvent viscosity, a nonzero polymeric viscosity, and a nonzero relaxation time; a Newtonian fluid, characterized by having a nonzero solvent viscosity, no polymeric viscosity, and relaxation time of zero; and a hybrid fluid [30] — a Maxwell fluid with an added solvent viscosity. Two geometries are used

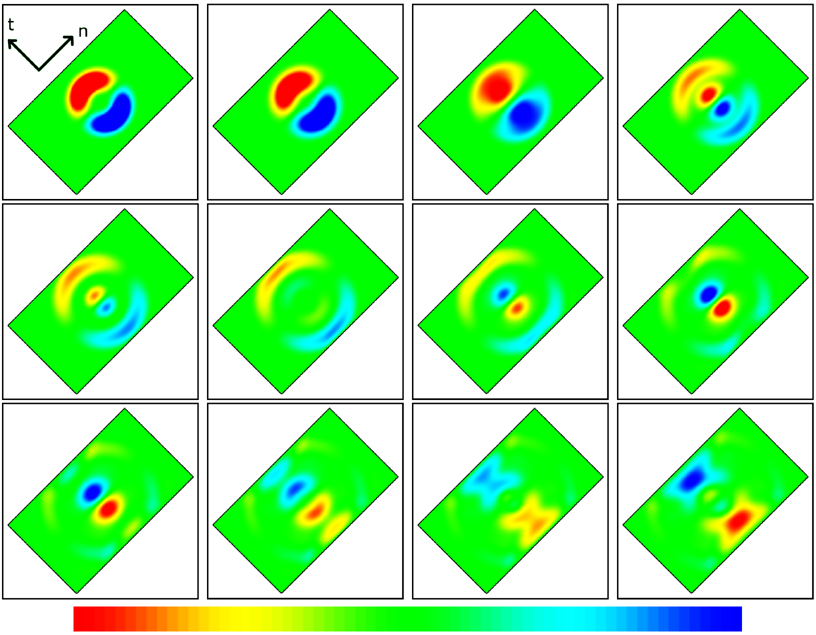


Figure 1. Time-dependent u_n profiles of a Maxwell fluid with a vortex initial condition in a rectangle. The domain has 256 × 256 cells with 24 time step increments using $\Delta t = 1.6 \times 10^{-3}$. The range is from −0.5 (red) to 0.5 (blue).

that are nonconforming with Cartesian grids; a rotated rectangular geometry, and a circular domain.

For the rectangular geometry, the computational domain has $l = w = 2.0$. The rectangular box has dimensions $l = 1.7$, and $w = 1.0$, and has been rotated 45° to maximize the amount of fluid in the computational domain. The coarse domain has 128 × 128 cells. We have chosen an initial vortex velocity profile that is sufficiently smooth at the vortex edge, given by the function

$$u_\theta(r) = 2.56[(r/0.45)(1 - r/0.45)]^4 H(0.45 - r),$$

where r is the distance to the center of the box and H is the Heaviside step function. This gives a maximum initial speed of $|\boldsymbol{u}| = 1.0$ at $r = 0.225$ (see Figure 1, top left).

For all images corresponding to the angled box geometry, we have rotated the output so the variables are seen with respect to the normal (lengthwise) and transverse (widthwise) directions. The initial pressure is set to zero. We define the characteristic speed, U, as the maximum initial velocity and the characteristic length, L, as the width of the box.

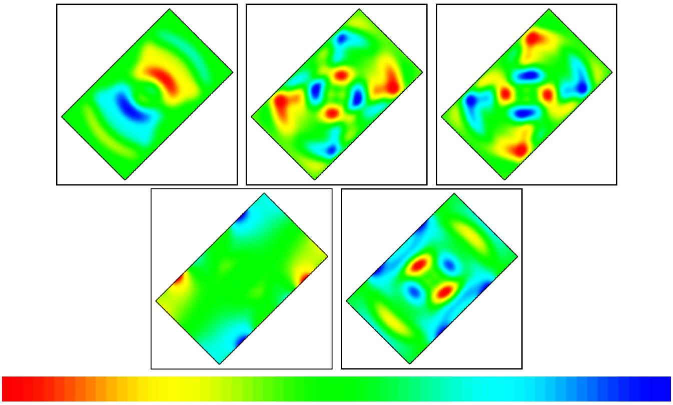


Figure 2. Profiles for a Maxwell fluid in a rectangle at $t = 0.4224$ (last image in Figure 1). Clockwise from top left: u_t, -0.5 (red) to 0.5 (blue); normal stress τ_{nn}, -0.21 (red) to 0.31 (blue); normal stress τ_{tt}, -0.21 (red) to 0.28 (blue); shear stress τ_{tn}, -0.46 (red) to 0.33 (blue); hydrostatic pressure p, 0 (red) to 0.656 (blue).

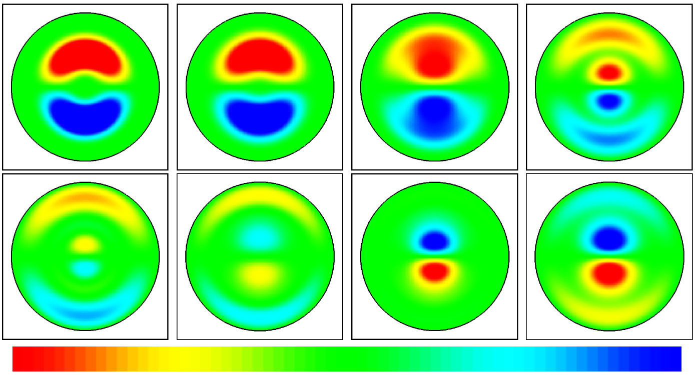


Figure 3. Time-dependent u_0 profiles of a Maxwell fluid with a vortex initial condition in a disk. The domain has 128×128 cells with 24 time step increments using $\Delta t = 1.6 \times 10^{-3}$. The range is from -0.50 (red) to 0.50 (blue).

For the circular geometry, the computational domain has $l = w = 1.0$ and the circle has radius $r = 0.45$ to maximize the amount of fluid in the computational domain. The coarse domain has 64×64 cells. The initial velocity profile is $u_\theta(r) = 2.56[(r/0.4)(1-r/0.4)]^4 H(0.4-r)$, which gives a maximum initial speed of $|\boldsymbol{u}| = 1.0$ at $r = 0.2$ (see Figure 3, top left). The initial pressure is set to zero. We define the characteristic speed, U, as the maximum initial velocity and the characteristic length, L, as the diameter of the circle.

norm	Variable	Coarse Error	Fine Error	Order
L_1	u_0	9.90e-04	2.69e-04	1.88
	u_1	9.62e-04	2.63e-04	1.87
	τ_{00}	1.24e-03	3.06e-04	2.02
	τ_{10}	1.38e-03	3.40e-04	2.02
	τ_{11}	1.37e-03	3.39e-04	2.01
	p	1.04e-03	2.68e-04	1.96
L_2	u_0	1.65e-03	4.34e-04	1.93
	u_1	1.66e-03	4.23e-04	1.97
	τ_{00}	1.89e-03	4.78e-04	1.98
	τ_{10}	3.06e-03	6.93e-04	2.14
	τ_{11}	3.27e-03	8.36e-04	1.97
	p	2.52e-03	4.78e-04	2.40
L_∞	u_0	4.08e-02	6.32e-03	2.69
	u_1	4.15e-02	6.98e-03	2.57
	τ_{00}	5.11e-02	1.09e-02	2.23
	τ_{10}	8.36e-02	2.77e-02	1.59
	τ_{11}	1.45e-01	3.73e-02	1.95
	p	7.76e-02	1.15e-02	2.75

Table 1. Solution error convergence rates for a Maxwell fluid with a vortex initial condition in a rectangle. Data correspond to Figures 1 and 2.

6.1. ***Maxwell fluid.*** For the Maxwell fluid, the rheological parameters are $\mu_s = 0$, $\mu_p = 1.0$, $\lambda = 1.0$, and $\rho = 1.0$. This gives the dimensionless parameters $\mathrm{Re} = 1.0$, $\mathrm{We} = 1.0$, and $\mathrm{Ma} = 1.0$ for the rectangular box geometry. The coarse time step for each geometry is 3.2×10^{-3}, corresponding to CFL ≈ 0.5. The time-dependent normal velocity is shown in Figure 1. The elastic wave propagation and reflection off the walls is clearly visible. The transverse velocity, stress, and pressure corresponding to the final image of normal velocity are shown in Figure 2. The solution error convergence after 400 fine time steps is given in Table 1. We use the same rheological parameters for the circular geometry, leading to dimensionless parameters $\mathrm{Re} = 0.9$, $\mathrm{We} = 0.9$, and $\mathrm{Ma} = 1.0$. The time-dependent u_0 profiles are shown in Figure 3. Again, the elastic wave propagation and reflection off the walls is easily visible. The u_1 component of velocity, stress, and pressure corresponding to the final image of u_0 are shown in Figure 4. The solution error convergence after 400 fine time steps is given in Table 2.

For Maxwell fluids, we have observed that additional cell-centered filtering steps (45) are required to prevent the buildup of divergent modes near cells with small

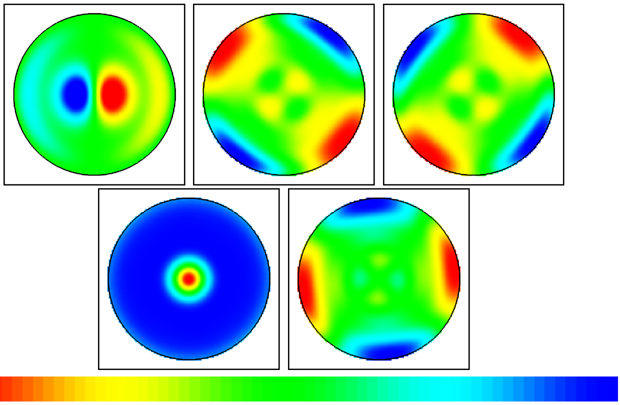


Figure 4. Profiles for a Maxwell fluid in a disk at $t = 0.2688$ (last image in Figure 3). Clockwise from top left: u_1, -0.50 (red) to 0.50 (blue); normal stress τ_{00}, -0.38 (red) to 0.67 (blue); normal stress τ_{11}, -0.38 (red) to 0.67 (blue); shear stress τ_{10}, -0.53 (red) to 0.53 (blue); hydrostatic pressure p, 0 (red) to 0.55 (blue).

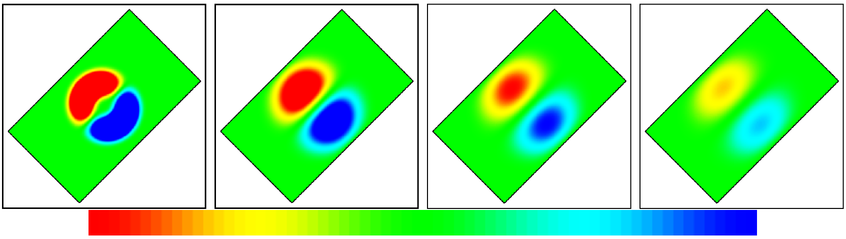


Figure 5. Time-dependent u_n profiles of a Newtonian fluid with a vortex initial condition in a rectangle. The domain has 256×256 cells with 2 time step increments using $\Delta t = 3.75 \times 10^{-3}$. The range is from -0.25 (red) to 0.25 (blue).

volume fractions. In the other flow regimes, the nonzero solvent viscosity in the diffusion equation solver smooths the velocity and helps eliminate the divergent modes and additional filtering steps are not required. The approach taken here to stabilize the method is to perform 1 filter iteration per time step at the coarse resolution, 2 iterations at the medium resolution, and 4 iterations at the fine resolution. The additional filter steps are not required for the other flow regimes, but are included for consistency.

6.2. ***Newtonian fluid.*** For the Newtonian fluid, the rheological parameters are $\mu_s = 1.0$, $\mu_p = 0.0$, $\lambda = 1.0 \times 10^{-11}$, and $\rho = 1.0$ leading to dimensionless parameters $\mathrm{Re} = 1.0$ and $\mathrm{We} = 0.0$ for the rectangular box geometry. Since $\mu_p = 0$,

norm	Variable	Coarse Error	Fine Error	Order
L_1	u_0	2.00e-03	5.70e-04	1.81
	u_1	2.05e-03	6.14e-04	1.74
	τ_{00}	2.01e-03	6.87e-04	1.55
	τ_{10}	1.62e-03	6.87e-04	1.39
	τ_{11}	2.03e-03	6.88e-04	1.56
	p	1.49e-03	5.62e-04	1.40
L_2	u_0	3.06e-03	1.01e-03	1.59
	u_1	3.15e-03	1.08e-03	1.55
	τ_{00}	3.09e-03	1.02e-03	1.60
	τ_{10}	2.33e-03	8.78e-04	1.41
	τ_{11}	3.00e-03	1.00e-03	1.58
	p	2.19e-03	8.60e-04	1.35
L_∞	u_0	3.11e-02	1.66e-02	0.91
	u_1	3.31e-02	1.64e-02	1.01
	τ_{00}	4.07e-02	2.24e-02	0.86
	τ_{10}	4.15e-02	1.94e-02	1.09
	τ_{11}	3.68e-02	2.31e-02	0.67
	p	3.04e-02	2.16e-02	0.49

Table 2. Solution error convergence rates for a Maxwell fluid with a vortex initial condition in a disk. Data correspond to Figures 3 and 4.

the polymeric stress remains zero at all times. The coarse time step for each geometry is 7.5×10^{-3}, corresponding to CFL ≈ 0.5. The time-dependent normal velocity is shown in Figure 5, in which the vortex spreads out to fill the box and decays over time. The transverse velocity and pressure corresponding to the final image of normal velocity are shown in Figure 6. The solution error convergence after 40 fine time steps is given in Table 3. Only a small number of time steps are used because after 40 the fluid velocity has already decayed to less than two percent of its initial value.

We use the same rheological parameters for the circular geometry, leading to dimensionless parameters Re $= 0.9$ and We $= 0$. The time-dependent normal velocity is shown in Figure 7. As with the rectangular box case, the vortex spreads out to fill the circle and decays over time. The transverse velocity and pressure corresponding to the final image of normal velocity are shown in Figure 8. The solution error convergence after 20 fine time steps is given in Table 4.

6.3. ***Hybrid fluid.*** For the hybrid fluid, the rheological parameters are $\mu_s = 0.1$, $\mu_p = 0.9$, $\lambda = 1.0$, and $\rho = 1.0$ leading to dimensionless parameters Re $= 1.0$, We $=$

1.0, and Ma = 1.05 for the rectangular box geometry. The initial stress is set to zero. The coarse time step is 3.2×10^{-3} for each geometry, corresponding to CFL ≈ 0.5. The time-dependent normal velocity is shown in Figure 9. As is the case with the Newtonian fluid, the vortex spreads out and decays over time, with a different shape than the Newtonian case. The transverse velocity, stress, and pressure corresponding

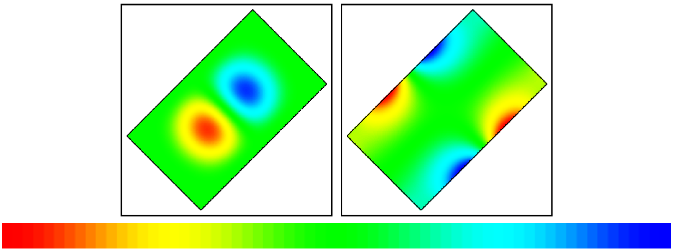


Figure 6. Profiles for a Newtonian fluid in a rectangle at $t = 2.25 \times 10^{-2}$ (last image in Figure 5). Left: u_t, −0.15 (red) to 0.15 (blue); right: hydrostatic pressure p, 0 (red) to 1.96 (blue).

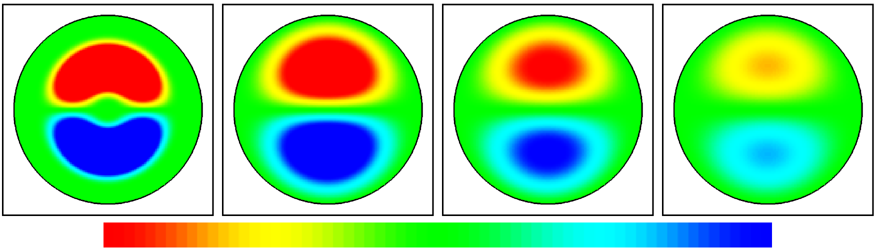


Figure 7. Time-dependent u_0 velocity profiles of a Newtonian fluid with a vortex initial condition in a disk. The domain has 128×128 cells with 2 time step increments using $\Delta t = 3.75 \times 10^{-3}$. The range is from −0.25 (red) to 0.25 (blue).

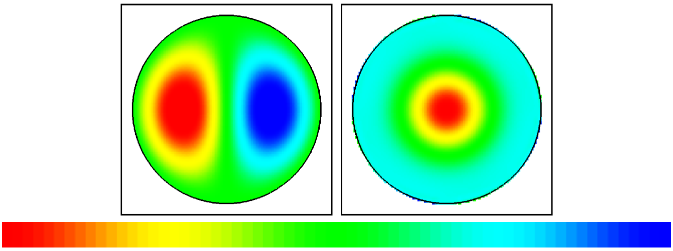


Figure 8. Profiles for a Newtonian fluid in a disk at $t = 2.25 \times 10^{-2}$ (last image in Figure 7). Left: u_1, −0.15 (red) to 0.15 (blue); right: hydrostatic pressure p, 0 (red) to 0.032 (blue).

norm	Variable	Coarse Error	Fine Error	Order
L_1	u_0	1.68e-04	3.53e-05	2.25
	u_1	1.68e-04	3.54e-05	2.25
	p	3.15e-03	1.16e-03	1.44
L_2	u_0	2.26e-04	4.63e-05	2.28
	u_1	2.26e-04	4.64e-05	2.28
	p	6.63e-03	2.00e-03	1.72
L_∞	u_0	2.36e-03	5.46e-04	2.11
	u_1	2.36e-03	5.46e-04	2.11
	p	4.12e-02	1.59e-02	1.37

Table 3. Solution error convergence rates for a Newtonian fluid with a vortex initial condition in a rectangle. Data correspond to Figures 5 and 6.

norm	Variable	Coarse Error	Fine Error	Order
L_1	u_0	4.06e-04	9.44e-05	2.10
	u_1	4.06e-04	9.44e-05	2.10
	p	5.12e-03	1.58e-04	5.02
L_2	u_0	4.88e-04	1.16e-04	2.07
	u_1	4.88e-04	1.16e-04	2.07
	p	1.04e-02	2.48e-04	5.40
L_∞	u_0	1.06e-03	5.51e-04	0.95
	u_1	1.06e-03	5.51e-04	0.95
	p	7.77e-02	1.13e-03	6.11

Table 4. Solution error convergence rates for a Newtonian fluid with a vortex initial condition in a disk. Data correspond to Figures 7 and 8.

to the final image of normal velocity are shown in Figure 10. The solution error convergence after 200 fine time steps is given in Table 5.

We use the same rheological parameters for the circular geometry, leading to dimensionless parameters $\mathrm{Re} = 0.9$, $\mathrm{We} = 0.9$, and $\mathrm{Ma} = 1.05$. The time-dependent u_0 component of velocity is shown in Figure 11. As with the rectangular box case, the vortex spreads out to fill the circle and decays over time. The transverse velocity, stress, and pressure corresponding to the final image of u_0 are shown in Figure 12. The solution error convergence after 200 fine time steps is given in Table 6.

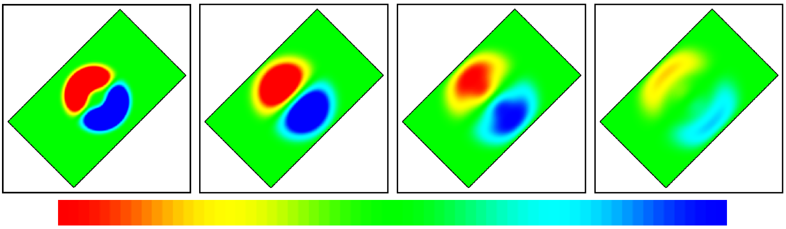


Figure 9. Time-dependent u_n profiles of a hybrid fluid with a vortex initial condition in a rectangle. The domain has 256 × 256 cells with 30 time step increments using $\Delta t = 1.6 \times 10^{-3}$. The range is from -0.25 (red) to 0.25 (blue).

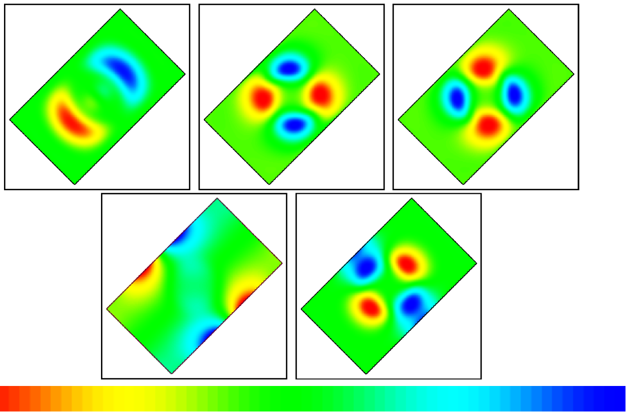


Figure 10. Profiles for a hybrid fluid in a rectangle at $t = 0.144$ (last image in Figure 9). Clockwise from top left: u_t, -0.15 (red) to 0.15 (blue); normal stress τ_{nn}, -0.25 (red) to 0.37 (blue); normal stress τ_{tt}, -0.25 (red) to 0.37 (blue); shear stress τ_{tn}, -0.30 (red) to 0.29 (blue); hydrostatic pressure p, 0 (red) to 0.55 (blue).

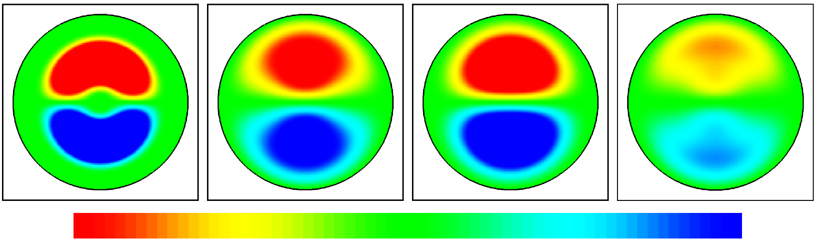


Figure 11. Time-dependent u_0 profiles of a hybrid fluid with a vortex initial condition in a disk. The domain has 128 × 128 cells with 20 time step increments using $\Delta t = 1.6 \times 10^{-3}$. The range is from -0.25 (red) to 0.25 (blue).

norm	Variable	Coarse Error	Fine Error	Order
L_1	u_0	8.74e-05	2.62e-05	1.74
	u_1	8.99e-05	2.93e-05	1.62
	τ_{00}	1.31e-04	3.78e-05	1.80
	τ_{10}	1.74e-04	5.81e-05	1.58
	τ_{11}	1.33e-04	3.96e-05	1.75
	p	2.49e-04	7.95e-05	1.64
L_2	u_0	1.98e-04	7.53e-05	1.39
	u_1	2.02e-04	7.75e-05	1.39
	τ_{00}	2.47e-04	1.00e-04	1.30
	τ_{10}	3.00e-04	1.28e-04	1.22
	τ_{11}	2.90e-04	1.40e-04	1.05
	p	3.82e-04	1.28e-04	1.58
L_∞	u_0	6.75e-03	3.26e-03	1.05
	u_1	6.82e-03	3.29e-03	1.05
	τ_{00}	2.67e-03	3.45e-03	-0.37
	τ_{10}	8.52e-03	4.40e-03	0.95
	τ_{11}	4.72e-03	6.15e-03	-0.38
	p	6.58e-03	3.16e-03	1.06

Table 5. Solution error convergence rates for a hybrid fluid with a vortex initial condition in a rectangle. Data correspond to Figures 9 and 10.

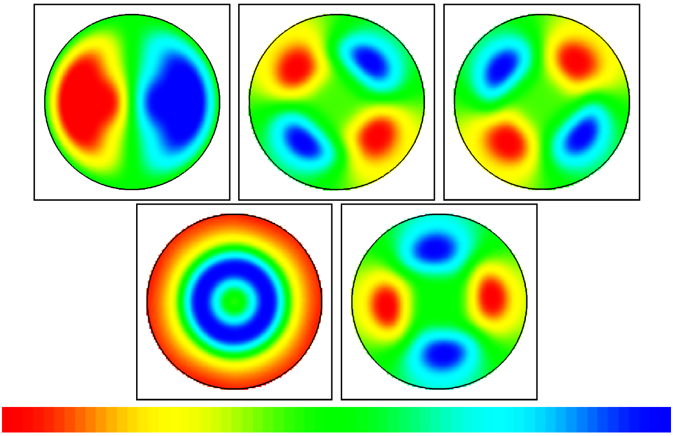


Figure 12. Profiles for a hybrid fluid in a disk at $t = 0.096$ (last image in Figure 11). Clockwise from top left: u_1, -0.15 (red) to 0.15 (blue); normal stress τ_{00}, -0.25 (red) to 0.35 (blue); normal stress τ_{11}, -0.25 (red) to 0.35 (blue); shear stress τ_{10}, -0.30 (red) to 0.30 (blue); hydrostatic pressure p, 0 (red) to 0.040 (blue).

norm	Variable	Coarse Error	Fine Error	Order
L_1	u_0	1.21e-04	3.05e-05	1.99
	u_1	1.27e-04	3.40e-05	1.91
	τ_{00}	2.66e-04	6.62e-05	2.01
	τ_{10}	3.95e-04	1.25e-04	1.66
	τ_{11}	2.54e-04	6.39e-05	1.99
	p	3.94e-04	1.40e-04	1.50
L_2	u_0	2.23e-04	6.86e-05	1.70
	u_1	2.27e-04	7.12e-05	1.67
	τ_{00}	3.53e-04	9.91e-05	1.83
	τ_{10}	4.94e-04	1.62e-04	1.60
	τ_{11}	3.44e-04	9.73e-05	1.82
	p	4.82e-04	1.72e-04	1.49
L_∞	u_0	2.31e-03	1.15e-03	1.00
	u_1	2.30e-03	1.15e-03	1.00
	τ_{00}	4.47e-03	2.13e-03	1.07
	τ_{10}	3.78e-03	1.64e-03	1.21
	τ_{11}	4.75e-03	2.28e-03	1.06
	p	2.19e-03	1.90e-03	0.20

Table 6. Solution error convergence rates for a hybrid fluid with a vortex initial condition in a disk. Data correspond to Figures 11 and 12.

7. Conclusions

For each of the test problems, we demonstrate second-order convergence of the solution error in L^1 and first-order in L^∞ for velocity and stress with an advective CFL time step constraint of CFL ≈ 0.5, as expected. This is an improvement over [30], in which less than second-order convergence was obtained with a smaller time step, and the algorithm did not support arbitrary smooth geometries. The algorithm also exhibits at least first-order convergence in L^1 for pressure, as expected. In some cases, such as the Maxwell fluid in the rectangular geometry, the convergence rates in L^∞ exceed first-order. This is due to the fact that given the position and shape of the expanded vortex, the largest magnitude errors occur in the interior of the domain, where the algorithm is second-order.

A feature calling for further study is the apparent need for additional projection filters (45) to smooth out the divergence in the velocity field of Maxwell fluids in irregular cells. Approaches include different filtering stencils, or different covered face state extrapolation algorithms.

The first obvious extension to this work is a three-dimensional discretization of the equations. The upwind method [6] and discretizations for Poisson's equation and the heat equation [29] in a three-dimensional embedded boundary framework have already been developed, so the extension is straightforward. The methods in this paper have been developed under the assumption that the geometry is sufficiently smooth. Additional studies are required to determine the robustness of the algorithm in the presence of discontinuous geometries, such as abrupt contractions. This will enable comparisons to standard benchmark problems [1; 26; 27; 30], such as the flow of elastic liquids in hard-cornered planar and axisymmetric contractions. Additional studies are also required to examine the robustness of this algorithm under higher values of We and Ma, and for a variety of operating conditions for experimental comparison [12; 13; 14]. In addition, adaptive numerical algorithms for the incompressible Navier–Stokes equations, in which the grid is locally refined in regions of interest, are being developed [21]. Adaptive techniques have already been used with success for hyperbolic conservation laws [6], so these two methods can be combined to develop a new adaptive projection method for incompressible viscoelasticity. Finally, another possible extension is the discretization of more advanced constitutive models, such as the PTT [25], White–Metzner [35] and Giesekus [11] models. The methods in this paper provide a framework for including the additional terms present in these models.

Acknowledgments

This work was performed under the auspices of the U. S. Department of Energy by the University of California, Lawrence Livermore National Laboratory (LLNL) under contract no. W-7405-Eng-48. Work at the Lawrence Berkeley National Laboratory was supported by the U. S. DOE Mathematical, Information, and Computer Sciences (MICS) Division under contract number DE-AC02-05CH11231. Work at the University of California, Davis was partially supported by the U. S. DOE MICS Division under contract number DE-FG02-03ER25579. A. Nonaka was supported by the Lawrence Livermore National Laboratory through the Student Employee Graduate Research Fellowship Program. At the time of this work, D. Trebotich was affiliated with LLNL's Center for Applied Scientific Computing.

References

[1] M. A. Alves, P. J. Oliveira, and F. T. Pinho, *Benchmark solutions for the flow of Oldroyd-B and PTT fluids in planar contractions*, J. Non-Newtonian Fluid Mech. **110** (2003), 45–75.

[2] J. B. Bell, P. Colella, and H. M. Glaz, *A second-order projection method for the incompressible Navier–Stokes equations*, J. Comput. Phys. **85** (1989), no. 2, 257–283. MR 90i:76002 Zbl 0681.76030

[3] A. J. Chorin, *Numerical solution of the Navier–Stokes equations*, Math. Comp. **22** (1968), 745–762. MR 39 #3723 Zbl 0198.50103

[4] P. Colella, *A direct Eulerian MUSCL scheme for gas dynamics*, SIAM J. Sci. Statist. Comput. **6** (1985), no. 1, 104–117. MR 86g:65156 Zbl 0562.76072

[5] ———, *Multidimensional upwind methods for hyperbolic conservation laws*, J. Comput. Phys. **87** (1990), no. 1, 171–200. MR 91c:76087 Zbl 0694.65041

[6] P. Colella, D. T. Graves, B. J. Keen, and D. Modiano, *A Cartesian grid embedded boundary method for hyperbolic conservation laws*, J. Comput. Phys. **211** (2006), no. 1, 347–366. MR 2006i:65142 Zbl 1120.65324

[7] M. J. Crochet, A. R. Davies, and K. Walters, *Numerical simulation of non-Newtonian flow*, Rheology Series, no. 1, Elsevier Scientific Publishing Co., Amsterdam, 1984. MR 86m:76002 Zbl 0583.76002

[8] R. K. Crockett, P. Colella, R. T. Fisher, R. J. Klein, and C. I. McKee, *An unsplit, cell-centered Godunov method for ideal MHD*, J. Comput. Phys. **203** (2005), no. 2, 422–448. MR 2005j:76065 Zbl 1143.76599

[9] T. A. Gardiner and J. M. Stone, *An unsplit Godunov method for ideal MHD via constrained transport*, J. Comput. Phys. **205** (2005), no. 2, 509–539. MR 2006m:65158 Zbl 1087.76536

[10] J.-M. Ghidaglia and F. Pascal, *The normal flux method at the boundary for multidimensional finite volume approximations in CFD*, Eur. J. Mech. B Fluids **24** (2005), no. 1, 1–17. MR 2005m:76121 Zbl 1060.76076

[11] H. Giesekus, *A simple constitutive equation for polymer fluids based on the concept of deformation-dependent tensorial mobility*, J. Non-Newtonian Fluid Mech. **11** (1982), 69–109.

[12] S. Gulati, *Effects of abrupt changes in microfluidic geometry on complex biological fluid flows*, Ph.D. thesis, University of California, Berkeley, 2007.

[13] S. Gulati, S. J. Muller, and D. Liepmann, *Direct measurements of viscoelastic flows in micro-contractions*, Proceedings of the 3rd International Conference on Microchannels and Minichannels (Toronto, Canada), June 2005.

[14] ———, *Quantifying viscoelastic behavior of DNA-laden flows in microfluidic systems*, Proceedings of the 3rd Annual International IEEE EMBS Special Topic Conference on Microtechnologies in Medicine and Biology (Kahuku, Oahu, HI), May 2005.

[15] F. H. Harlow and J. E. Welch, *Numerical calculation of time-dependent viscous incompressible flow of fluid with free surface*, Physics of Fluids **8:12** (1965), 2182–2189.

[16] H. Johansen and P. Colella, *A Cartesian grid embedded boundary method for Poisson's equation on irregular domains*, J. Comput. Phys. **147** (1998), no. 1, 60–85. MR 99m:65231 Zbl 0923.65079

[17] D. D. Joseph, *Fluid dynamics of viscoelastic liquids*, Applied Mathematical Sciences, no. 84, Springer, New York, 1990. MR 91d:76003 Zbl 0698.76002

[18] D. D. Joseph, M. Renardy, and J.-C. Saut, *Hyperbolicity and change of type in the flow of viscoelastic fluids*, Arch. Rational Mech. Anal. **87** (1985), no. 3, 213–251. MR 86c:76003 Zbl 0572.76011

[19] R. Kupferman, *Simulation of viscoelastic fluids: Couette–Taylor flow*, J. Comput. Phys. **147** (1998), no. 1, 22–59. MR 99h:76067 Zbl 0935.76058

[20] M. F. Lai, *A projection method for reacting flow in the zero mach number limit*, Ph.D. thesis, University of California, Berkeley, 1994.

[21] D. F. Martin and P. Colella, *A cell-centered adaptive projection method for the incompressible Euler equations*, J. Comp. Phys. **163** (2000), 311–333.

[22] P. McCorquodale, P. Colella, and H. Johansen, *A Cartesian grid embedded boundary method for the heat equation on irregular domains*, J. Comput. Phys. **173** (2001), no. 2, 620–635. MR 2002h:80009 Zbl 0991.65099

[23] G. H. Miller and P. Colella, *A high-order Eulerian Godunov method for elastic-plastic flow in solids*, J. Comp. Phys. **167** (2001), 131–176.

[24] A. J. Nonaka, *A higher-order upwind method for viscoelastic flow*, Ph.D. thesis, University of California, Davis, 2007.

[25] N. Phan-Thien and R. I. Tanner, *A new constitutive equation based derived from network theory*, J. Non-Newtonian Fluid Mech. **2** (1977), 353–365.

[26] T. N. Phillips and A. J. Williams, *Viscoelastic flow through a planar contraction using a semi-Lagrangian finite volume method*, J. Non-Newtonian Fluid Mech. **87** (1999), 215–246.

[27] ———, *Comparison of creeping and inertial flow of an Oldroyd B fluid through planar and axisymmetric contractions*, J. Non-Newtonian Fluid Mech. **108** (2002), 25–47.

[28] J. Saltzman, *An unsplit* 3*D upwind method for hyperbolic conservation laws*, J. Comput. Phys. **115** (1994), no. 1, 153–168. MR MR1300337 Zbl 0813.65111

[29] P. Schwartz, M. Barad, P. Colella, and T. Ligocki, *A Cartesian grid embedded boundary method for the heat equation and Poisson's equation in three dimensions*, J. Comput. Phys. **211** (2006), no. 2, 531–550. MR 2006e:65194 Zbl 1086.65532

[30] D. Trebotich, P. Colella, and G. H. Miller, *A stable and convergent scheme for viscoelastic flow in contraction channels*, J. Comput. Phys. **205** (2005), no. 1, 315–342. MR MR2132311 Zbl 1087.76005

[31] D. P. Trebotich and P. Colella, *A projection method for incompressible viscous flow on moving quadrilateral grids*, J. Comput. Phys. **166** (2001), no. 2, 191–217. MR 2001m:76076 Zbl 1030.76044

[32] E. H. Twizell, A. B. Gumel, and M. A. Arigu, *Second-order, L_0-stable methods for the heat equation with time-dependent boundary conditions*, Adv. Comput. Math. **6** (1996), no. 3-4, 333–352. MR 97m:65164 Zbl 0872.65084

[33] J. S. Ultman and M. M. Denn, *Anomalous heat transfer and a wave phenomenon in dilute polymer solutions*, Trans. Soc. Rheology **14** (1970), 307–317.

[34] B. van Leer, *Towards the ultimate conservative difference scheme, V: A second-order sequel to Godunov's method*, J. Comp. Phys. **32** (1979), 101–136.

[35] J. L. White and A. B. Metzner, *Development of constitutive equations for polymeric melts and solutions*, J. Appl. Polymer Sci. **7** (1963), 1867–1889.

Received August 7, 2008. Revised May 22, 2009.

ANDREW NONAKA: AJNonaka@lbl.gov
Center for Computational Sciences and Engineering, Lawrence Berkeley National Laboratory,
Mail Stop 50A-1148, 1 Cyclotron Road, Berkeley, CA 94720-8142, United States
https://seesar.lbl.gov/ccse/index.html

DAVID TREBOTICH: DPTrebotich@lbl.gov
Applied Numerical Algorithms Group, Lawrence Berkeley National Laboratory, Mail Stop 50A-1148,
1 Cyclotron Road, Berkeley, CA 94720-8142, United States

GREGORY MILLER: grgmiller@ucdavis.edu
Department of Applied Science, University of California, Davis, 1 Shields Ave.,
Davis, CA 95616-8254, United States

DANIEL GRAVES: DTGraves@lbl.gov
Applied Numerical Algorithms Group, Lawrence Berkeley National Laboratory, Mail Stop 50A-1148,
1 Cyclotron Road, Berkeley, CA 94720-8142, United States

PHILLIP COLELLA: PColella@lbl.gov
Applied Numerical Algorithms Group, Lawrence Berkeley National Laboratory, Mail Stop 50A-1148,
1 Cyclotron Road, Berkeley, CA 94720-8142, United States

COMM. APP. MATH. AND COMP. SCI.
Vol. 4, No. 1, 2009

A NUMERICAL METHOD FOR CELLULAR ELECTROPHYSIOLOGY BASED ON THE ELECTRODIFFUSION EQUATIONS WITH INTERNAL BOUNDARY CONDITIONS AT MEMBRANES

YOICHIRO MORI AND CHARLES S. PESKIN

We present a numerical method for solving the system of equations of a model of cellular electrical activity that takes into account both geometrical effects and ionic concentration dynamics. A challenge in constructing a numerical scheme for this model is that its equations are stiff: There is a time scale associated with "diffusion" of the membrane potential that is much faster than the time scale associated with the physical diffusion of ions. We use an implicit discretization in time and a finite volume discretization in space. We present convergence studies of the numerical method for cylindrical and two-dimensional geometries for several cases of physiological interest.

1. Introduction

Cellular electrical activity is central to cellular physiology [1], and it has been an area in which mathematical modeling has seen great success [16; 18]. Most models of cellular electrical activity are based on the cable model, in which an ohmic current continuity relation results in a one-dimensional reaction diffusion system [16; 18].

In the derivation of the cable model, one assumes that the ionic concentrations do not change appreciably over the time of interest, and that a one-dimensional picture of cell geometry is adequate for purposes of describing cellular electrical activity. In [25; 27], we presented a three-dimensional model of cellular electrical activity that takes into account both ionic concentration and geometrical effects on electrophysiology. The resulting system of partial differential equations has the virtue of being more general in its physiological applicability, but has the difficulty of being far more complicated to study either analytically or numerically.

In this paper, we develop an efficient numerical method to solve this system of equations in two spatial dimensions. In Section 2, we give a short presentation of

MSC2000: 65M12, 92C30, 92C50.

Keywords: three-dimensional cellular electrophysiology, electrodiffusion, ephaptic transmission, finite volume method.

the model equations and in Section 3 we discuss the time and space scales that are relevant to the behavior of the model. We shall see that the model equations have two time scales of interest, the ionic diffusion time scale and the membrane potential time scale. The membrane potential time scale is associated with the "diffusion" of the membrane potential, which is closely related to the spread of the membrane potential in the cable model. In Section 4, we discuss spatial discretization. We use a finite volume scheme and develop a numerical scheme for cylindrical geometry and a related scheme for arbitrary two-dimensional geometry. In Section 5, we discuss time discretization. We use an operator splitting approach. Each time step is split into two substeps, one in which the gating variables are updated and the other in which the electrostatic potential and the ionic concentrations are updated. For the latter substep, a the electrostatic potential and ionic concentrations are treated implicitly to deal with the disparity of time scales mentioned above. We then discuss the iterative numerical solution of the nonlinear algebraic equations which result from the discretization. We conclude with convergence studies using several examples of biophysical relevance: the Hodgkin–Huxley axon, ephaptic transmission between cardiac cells, and three model geometries at length scales typically found in the central nervous system.

2. Model equations

We consider biological tissue to be a three-dimensional space partitioned into the intracellular and extracellular spaces by membranes. In these regions, we track the ionic concentrations as well as the electrostatic potential. Let the biological tissue of interest be divided into membrane bound subregions $\Omega^{(k)}$, indexed by k. We denote the membrane separating the regions $\Omega^{(k)}$ and $\Omega^{(l)}$ by $\Gamma^{(kl)}$ (Figure 1).

In any of the subregions $\Omega^{(k)}$, both the extracellular and intracellular, the equations satisfied by the ionic concentrations c_i and the electrostatic potential ϕ are

$$\frac{\partial c_i}{\partial t} = -\nabla \cdot \boldsymbol{f}_i, \qquad \text{(ion conservation)} \qquad (1)$$

$$\boldsymbol{f}_i = -D_i\Big(\nabla c_i + \frac{qz_ic_i}{k_BT}\nabla\phi\Big), \qquad \text{(drift-diffusion flux)} \qquad (2)$$

$$0 = \rho_0 + \sum_{i=1}^{N} qz_ic_i, \qquad \text{(electroneutrality condition).} \qquad (3)$$

Here, $\boldsymbol{f}_i$ denotes the flux of the i-th species of ion. This quantity is expressed as a sum of a diffusion term and a drift term. D_i is the diffusion coefficient of the i-th ion, qz_i is the amount of charge on the i-th ion, where q is the proton charge. Thus $qD_i/(k_BT)$ is the mobility of the ion species (Einstein relation), where k_B is the

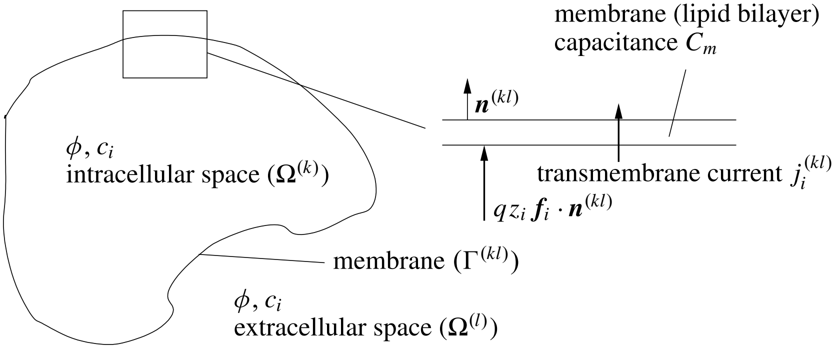

membrane (lipid bilayer)
capacitance C_m
$\boldsymbol{n}^{(kl)}$
ϕ, c_i
intracellular space ($\Omega^{(k)}$)
transmembrane current $j_i^{(kl)}$
$qz_i \boldsymbol{f}_i \cdot \boldsymbol{n}^{(kl)}$
membrane ($\Gamma^{(kl)}$)
ϕ, c_i
extracellular space ($\Omega^{(l)}$)

Figure 1. The variables ϕ, c_i are defined in the regions $\Omega^{(k)}$ and $\Omega^{(l)}$, which we have identified as intracellular and extracellular regions in the above. The membrane acts primarily as a capacitor, but possesses ionic channels through which transmembrane current can flow.

Boltzmann constant and T the absolute temperature. The fixed background charge density (if any) is given by ρ_0.

The electrostatic potential ϕ is determined implicitly by the electroneutrality condition (3). We can obtain an equation that is satisfied by ϕ by taking the derivative of (3) with respect to time t:

$$\sum_{i=1}^{N} qz_i \frac{\partial c_i}{\partial t} = \sum_{i=1}^{N} qz_i \nabla \cdot \boldsymbol{f}_i = \nabla \cdot (a\nabla\phi + \nabla b) = 0, \tag{4}$$

where

$$a(\boldsymbol{x}, t) = \sum_{i=1}^{N} \frac{(qz_i)^2 D_i}{k_B T} c_i(\boldsymbol{x}, t), \quad b(\boldsymbol{x}, t) = \sum_{i=1}^{N} qz_i D_i c_i(\boldsymbol{x}, t). \tag{5}$$

Thus, ϕ satisfies an elliptic constraint such that electroneutrality is satisfied at each instant of time.

We now turn to the boundary conditions, satisfied at both the intracellular and extracellular sides of the cell membrane. Biological membranes consist largely of a lipid bilayer. In this cell membrane are embedded ionic channels and transporters through which certain ionic species may pass.

Across the cell membrane, a jump in electrostatic potential (*membrane potential*) is maintained, and the cell membrane acts as a capacitor. There is, therefore, a thin layer (*space charge layer*) on both sides of the membrane where electric charge accumulates. The thickness of this layer is on the order of the *Debye length* which measures approximately 1 nm in physiological systems. In (3), we have taken the electroneutrality condition to hold inside and outside the cell, and this implies that we must treat the effect of having Debye layers in the form of boundary conditions.

The strength of the ionic current through an open ionic channel is determined by the membrane potential and ionic concentrations on either side of the membrane. Ionic channels may open or close, and the dynamics of this *gating* is also determined in large part by the membrane potential and ionic concentrations [10].

The boundary conditions satisfied on the $\Omega^{(k)}$ face of the membrane $\Gamma^{(kl)}$ are

$$\frac{\partial \sigma_i^{(k)}}{\partial t}(\boldsymbol{x}_m, t) + j_i^{(kl)}(\boldsymbol{x}_m, t) = q z_i \boldsymbol{f}_i^{(k)} \cdot \boldsymbol{n}^{(kl)}(\boldsymbol{x}_m). \tag{6}$$

All variables are defined on the boundary, and a spatial location on the boundary is denoted by $\boldsymbol{x}_m$. The term $j_i^{(kl)}$ denotes the transmembrane current per unit area from region $\Omega^{(k)}$ into $\Omega^{(l)}$. We note by definition that $j_i^{(kl)} = -j_i^{(lk)}$. The variable $\sigma_i^{(k)}$ denotes the contribution of the i-th species of ion to surface charge per unit area. The above boundary condition states that the current that flows onto the membrane either goes across the membrane through ionic channels, or contributes to change in surface charge.

In order to make (6) into a useful boundary condition, we must be able to write $j_i^{(kl)}$ and $\sigma_i^{(k)}$ in terms of c_i and ϕ. The surface charge density $\sigma_i^{(k)}$ is expressed as

$$\sigma_i^{(k)} = \lambda_i^{(k)}(\boldsymbol{x}_m, t)\sigma^{(k)}(\boldsymbol{x}_m, t), \qquad \sigma^{(k)} = C_{\mathrm{m}}\phi^{(kl)}, \tag{7}$$

$$\frac{\partial \lambda_i^{(k)}}{\partial t} = \frac{\tilde{\lambda}_i^{(k)} - \lambda_i^{(k)}}{\tau}, \qquad \tilde{\lambda}_i^{(k)}(\boldsymbol{x}_m, t) = \frac{z_i^2 c_i^{(k)}}{\sum_{i'=1}^{N} z_{i'}^2 c_{i'}^{(k)}}. \tag{8}$$

Here, $c_i^{(k)}$ and $\phi^{(k)}$ denote limiting values of c_i and ϕ as one approaches the membrane from the $\Omega^{(k)}$ side of the membrane $\Gamma^{(kl)}$ and $\phi^{(kl)} = \phi^{(k)} - \phi^{(l)}$ is the membrane potential. τ is a relaxation time constant which we shall discuss shortly. $\sigma^{(k)}$ is the total charge on the $\Omega^{(k)}$ side of the membrane surface and is the product of C_{m}, the capacitance of the membrane and the membrane potential $\phi^{(kl)}$. Since $\phi^{(kl)} = -\phi^{(lk)}$, (7) implies that $\sigma^{(k)} = -\sigma^{(l)}$ at each point of the membrane. Thus, like a capacitor, each patch of membrane is electrically neutral, since the charge stored on one side of the membrane balances the charge stored on the other side.

Note that $\lambda_i^{(k)}$ is the fractional contribution of the i-th species of ion to the surface charge density on face k of the membrane (7). The quantity $\lambda_i^{(k)}$ relaxes to $\tilde{\lambda}_i^{(k)}$ with time constant $\tau = r_d^2/D_0 = 1$ ns, the diffusive time scale within the Debye layer (r_d is the aforementioned Debye length and D_0 is a representative diffusive constant for ions). This relaxation time is introduced to avoid an instability that occurs if we formally take the limit $\tau \to 0$ and set $\lambda_i^{(k)} = \tilde{\lambda}_i^{(k)}$; see the Appendix for further discussion and [25; 24] for details. The choice $\tau = 1$ ns is large enough to avoid this instability and yet small enough that $\lambda_i^{(k)}$ remains close to $\tilde{\lambda}_i^{(k)}$ at all times in any practical application.

The derivation of specific formula for $\tilde{\lambda}_i^{(k)}$ given in (8) requires a closer look at the ionic composition of the space charge layer. A derivation by physical reasoning in [27] and by matched asymptotic analysis in [25] and [24]. A quick derivation is given in the Appendix for convenience of the reader. The expression for $\tilde{\lambda}_i^{(k)}$ states that the fractional contribution of each species ion to the surface charge on one face of the membrane is given by the concentration of that ion species in the bulk solution near that face of the membrane weighted by the *square* of the charge carried by that species of ion. This result is closely related to the concept of *ionic strength* in electrochemistry [5], which is defined as $\frac{1}{2}z_i^2 c_i$. Note the implication that ions of either sign can contribute, for example, to a positive space charge layer. Such a layer involves an increased concentration of positive ions and a reduced concentration of negative ions in comparison to the concentrations of these ions in the electroneutral bulk solution outside of the space charge layer.

The interpretation of $\lambda_i^{(k)}$ as the fractional contribution of the i-th ion species to the surface charge on face k of the membrane requires that $\sum_i \lambda_i^{(k)} = 1$ be satisfied identically, at all membrane locations for all time. To verify this condition, sum both parts of (8) from $i = 1, \ldots, N$. The second part gives $\sum_i \tilde{\lambda}_i^{(k)} = 1$, and the first part therefore shows that $\sum_i \lambda_i^{(k)}$ relaxes to 1, and indeed is identically equal to 1 if it is equal to 1 initially. We assume in the sequel that the initial values of λ_i^k have this property.

We now discuss j_i, the transmembrane currents. Biophysically, these are currents that flow through ion channels, transporters, or pumps that are located within the cell membrane [1; 10; 15]. We use the formalism of Hodgkin and Huxley for ion channel currents [11; 16; 18], generalized to allow for nonlinear instantaneous current-voltage relations and ion concentration effects.

$$j_i^{(kl)}(\boldsymbol{x}_m, t) = J_i^{(kl)}(\boldsymbol{x}_m, s^{(kl)}, \phi^{(kl)}, c^{(k)}, c^{(l)}). \tag{9}$$

The transmembrane current density $J_i^{(kl)}$ is a function characteristic of the channels (possibly of more than one type) that carry the i-th species of ion across the membrane separating $\Omega^{(k)}$ from $\Omega^{(l)}$. The explicit dependence of $J_i^{(kl)}$ on $\boldsymbol{x}$ reflects the possible inhomogeneity of the membrane: the density of channels may vary from one location to another. The other arguments of $J_i^{(kl)}$ are as follows.

First, there is a vector of gating variables $s^{(kl)}(\boldsymbol{x}_m, t) = (s_1^{(kl)}, \ldots, s_G^{(kl)})$ where G is the total number of gating variables in all of the channel types that arise in our system. (Only some of these influence the channels that conduct ions of species i.) The individual components $s_g^{(kl)}$ of $s^{(kl)}$ are dimensionless variables as introduced by Hodgkin and Huxley [11] that take values in the interval [0, 1] and satisfy ordinary differential equations of the form,

$$\frac{\partial s_g^{(kl)}}{\partial t} = \alpha_g^{(kl)}(\phi^{(kl)})(1 - s_g^{(kl)}) - \beta_g^{(kl)}(\phi^{(kl)})s_g^{(kl)} \tag{10}$$

for $g = 1, \dots, G$ where $\alpha_g^{(kl)}$ and $\beta_g^{(kl)}$ are positive, empirically defined functions of the transmembrane potential. In general, the gating variables obey a more complicated ordinary differential equation:

$$\frac{\partial s_g^{(kl)}}{\partial t} = f_g^{(kl)}(s^{(kl)}, \phi^{(kl)}, c^{(k)}, c^{(l)}). \tag{11}$$

We note the conditions $j_i^{(kl)} = -j_i^{(lk)}$ and $\phi^{(kl)} = -\phi^{(lk)}$ impose the following constraints on the form of the functions $\alpha_g^{(kl)}$, $\beta_g^{(kl)}$ and $f_g^{(kl)}$:

$$\alpha_g^{(kl)}(\phi^{(kl)}) = \alpha_g^{(lk)}(\phi^{(lk)}), \quad \beta_g^{(kl)}(\phi^{(kl)}) = \beta_g^{(lk)}(\phi^{(lk)}), \tag{12}$$

$$f_g^{(kl)}(s^{(kl)}, \phi^{(kl)}, c^{(k)}, c^{(l)}) = f_g^{(lk)}(s^{(lk)}, \phi^{(lk)}, c^{(l)}, c^{(k)}). \tag{13}$$

The next argument of $J_i^{(kl)}$ is again the transmembrane potential $\phi^{(kl)}$. Holding the other arguments fixed in $J_i^{(kl)}$, and letting only $\phi^{(kl)}$ vary, we get the instantaneous current-voltage relationship for current carried by the i-th ion from $\Omega^{(k)}$ to $\Omega^{(l)}$ at point $\boldsymbol{x}$ at time t.

The last two arguments of $J_i^{(kl)}$ are the vectors of ion concentrations on the two sides of the membrane: $c^{(k)} = (c_1^{(k)}, \dots, c_N^{(k)})$ and similarly for $c^{(l)}$. By including the whole vector of ion concentrations, we allow for the possibility that the current carried by the i-th species of ion is influenced by the concentrations of other ionic species on the two sides of the membrane. This, for example, is the case with calcium gated potassium channels ($\mathrm{K_{Ca}}$ channels) whose potassium conductance is controlled by the intracellular calcium concentration [10].

Equations (1)–(3) with the boundary condition (6) is the model we consider in this paper. We shall call this model the *electroneutral model*.

3. Cable model and multiple timescales

The above electroneutral model provides a more detailed description of cellular electrophysiology than the more familiar one-dimensional cable model (see (20), below). In this section, we sketch the derivation of the cable model from the electroneutral model. We do so in part to confirm that the electroneutral model contains the cable model as a limiting case, but also to bring out the different time scales that will complicate the numerical solution of the equations of the electroneutral model. For a more complete exposition of the derivation sketched here, see [25].

First, recall from (4) that ϕ satisfies an elliptic equation. The boundary conditions for this equation can be obtained by the sum over i in (6):

$$C_{\mathrm{m}}\frac{\partial\phi^{(kl)}}{\partial t}+I^{(kl)}(\boldsymbol{x}_m,t)=-(a\nabla\phi+\nabla b)\cdot\boldsymbol{n}, \tag{14}$$

where we have used $\sum_{i=1}^{N}\sigma_i=C_{\mathrm{m}}\phi^{(kl)}$ (7) and $I^{(kl)}\equiv\sum_{i=1}^{N}j_i^{(kl)}$. The coefficients a and b were given in (5). Thus, the electrostatic potential satisfies an elliptic problem with an evolutionary boundary condition satisfied at the membrane.

Suppose now that the ionic concentrations inside and outside the cell do not change appreciably in the time of biophysical interest, and that the ionic concentrations gradients are negligible. Then, we have only to track the evolution of the electrostatic potential and the coefficient a and b of (4) is constant within each of the domains separated by the membrane. Thus, (4) and (14) can now be written as

$$\Delta\phi=0 \qquad \text{in } \Omega^{(k)},\Omega^{(l)}, \tag{15}$$

$$C_{\mathrm{m}}\frac{\partial\phi^{(kl)}}{\partial t}+I^{(kl)}=-a^{(k)}\frac{\partial\phi}{\partial\boldsymbol{n}^{(kl)}}=-a^{(l)}\frac{\partial\phi}{\partial\boldsymbol{n}^{(kl)}} \qquad \text{on } \Gamma^{(kl)}. \tag{16}$$

where $a^{(k)}$ and $a^{(l)}$ are now constants defined within each domain. The gradient of b disappears from the equations because we have assumed that we do not have a concentration gradient. We see that the evolution of the electrostatic potential is completely specified by what happens at the boundary. We note that Equations (15) and (16) have been used to model cellular electrophysiology and is also the basis for the bidomain model used in tissue level electrophysiology [8; 3; 28; 16].

Consider a simple situation in which we have just two regions, one intracellular and the other extracellular. We take the intracellular region to be a bounded simply connected set whereas the extracellular space its complement in $\mathbb{R}^3$. For simplicity, suppose that $a^{\mathrm{int}}=a^{\mathrm{ext}}$. Consider the following boundary value problem for ϕ.

$$\Delta\phi=0 \qquad \text{in } \Omega^{\mathrm{ext}},\Omega^{\mathrm{int}}, \tag{17}$$

$$\phi_{\mathrm{m}}\equiv\phi^{\mathrm{int}}-\phi^{\mathrm{ext}}=f,\quad \frac{\partial\phi^{\mathrm{int}}}{\partial\boldsymbol{n}}=\frac{\partial\phi^{\mathrm{ext}}}{\partial\boldsymbol{n}} \qquad \text{on } \Gamma. \tag{18}$$

where f is some function given on the membrane Γ and $\boldsymbol{n}$ is the unit normal pointing from the intracellular to extracellular side of the cell. We require that ϕ decays to 0 at infinity. The above boundary value problem defines a map from $\phi_{\mathrm{m}}=f$ to $\partial\phi^{\mathrm{int}}/\partial\boldsymbol{n}=\partial\phi^{\mathrm{ext}}/\partial\boldsymbol{n}$. This is similar to the usual Dirichlet-to-Neumann map on a single domain, except that we are here solving a Laplace problem on both sides of the membrane interface, and the input we are given is the *jump* in the electrostatic potential. We denote this map as $\mathscr{L}$. Using this map, and the simplification $a=a^{\mathrm{int}}=a^{\mathrm{ext}}$, we can write (15) and (16) as

$$C_{\mathrm{m}}\frac{\partial \phi_{\mathrm{m}}}{\partial t}+I=-a\mathscr{L}\phi_{\mathrm{m}}. \tag{19}$$

We now clearly see that the evolution of ϕ is confined to the boundary. It is straightforward to show that $\mathscr{L}$ can be extended to a nonnegative self-adjoint operator on square integrable functions on Γ. This tells us that ϕ_{m} evolves according to an evolutionary equation similar to a reaction-diffusion equation, where the Laplacian is replaced with -1 times $\mathscr{L}$. Thus, there is a "diffusive" process that takes place on the two dimensional membrane surface. We shall call this *membrane potential diffusion*.

We would like to compare the speeds of the two dissipative processes at play: ionic diffusion and membrane potential diffusion. The "diffusion" coefficient a/C_{m} in front of the operator $-\mathscr{L}$ in (19) has dimensions of length/time. Therefore, it cannot be compared directly with the ionic diffusion coefficient D_i which has dimensions length^2/time. However, if there is a natural characteristic length scale L associated with the geometry of the system, the combination $D_\phi \equiv aL/C_{\mathrm{m}}$ may be used as a value to be compared with D_i.

Suppose the cell is cylindrical in shape. Assuming that the membrane potential varies slowly on the length scale defined by the radius of the cylinder, (19) can be further reduced to the following one-dimensional reaction diffusion equation.

$$C_{\mathrm{m}}\frac{\partial \phi_{\mathrm{m}}}{\partial t}+I=\frac{aR}{2}\frac{\partial^2 \phi_{\mathrm{m}}}{\partial z^2}, \tag{20}$$

where R is the radius of the cylinder and z is the axial coordinate. This is nothing other than the cable model. A quick derivation of this is given in the Appendix. The factor $R/2$ comes from the ratio of the cylindrical cross-sectional area to the circumference: $\pi R^2/(2\pi R)$. In (20), $L=R/2$ emerges as the natural characteristic length scale, and $D_\phi = aR/(2C_{\mathrm{m}})$. Let us examine the ratio between D_ϕ and D_i.

$$D_\phi=\frac{aR}{2C_{\mathrm{m}}}=\sum_{i=1}^{N}\frac{L(qz_i)^2c_i}{C_{\mathrm{m}}k_BT}D_i=\sum_{i=1}^{N}\frac{Lqc_i}{C_{\mathrm{m}}(k_BT/q)}\frac{1}{2}z_i^2D_i, \tag{21}$$

where we used (5) in the second equality. Given that z_i^2 is an order 1 quantity,

$$\frac{D_\phi}{D_i}\approx\frac{Lqc_0}{C_{\mathrm{m}}(k_BT/q)}=10^4\sim 10^6, \tag{22}$$

where c_0 is the typical ionic concentration. The above is a ratio of the absolute amount of charge in the electrolyte solution to the membrane surface charge, which turns out to be 10^4 to 10^6 in physiological systems. This illustrates the presence of two disparate time scales in the problem.

That membrane potential diffusion is fast and dissipative has important implications for time stepping, to be discussed in Section 5.

4. Spatial discretization

4.1. ***Finite volume method.*** We shall use a finite volume discretization in space [21]. Take any finite volume Ω_{fv} contained in $\Omega^{(k)}$ and suppose the boundary of this region is comprised of two components, the Γ_{el} component that faces the electrolyte solution, and the Γ_{m} component that faces the membrane. It may be the case that either Γ_{el} or Γ_{m} is empty. For each ionic species, we have the following conservation relation in integral form:

$$\begin{aligned}\int_{\Omega_{\text{fv}}} \frac{\partial c_i}{\partial t} dV &= \int_{\Gamma_{\text{el}}\cup\Gamma_{\text{m}}} \boldsymbol{f}_i \cdot \boldsymbol{n} dA \\ &= -\int_{\Gamma_{\text{el}}} D_i \big(\nabla c_i + \frac{qz_i c_i}{k_B T}\nabla\phi\big)\cdot \boldsymbol{n} dA \\ &\qquad + \int_{\Gamma_{\text{m}}} \frac{1}{qz_i}\big(C_{\text{m}} \frac{\partial(\lambda_i^{(k)}\phi^{(kl)})}{\partial t} + j_i^{(kl)}\big) dA. \end{aligned} \tag{23}$$

The electroneutrality condition is equivalent to saying that

$$\rho_0 + \sum_{i=1}^N qz_i c_i = 0 \text{ at } t = 0, \qquad \sum_{i=1}^N qz_i \frac{\partial c_i}{\partial t} = 0, \qquad \text{for } t > 0. \tag{24}$$

As long as the electroneutrality condition is satisfied at $t = 0$, we have only to consider the time derivative of the electroneutrality condition for time $t > 0$. Therefore, we can obtain the electroneutrality condition expressed in integral form by taking (23), multiplying by qz_i and summing in i.

$$\begin{aligned} 0 = &-\int_{\Gamma_{\text{el}}} \Bigg(\sum_{i=1}^N qz_i D_i \Big(\nabla c_i + \frac{qz_i c_i}{k_B T}\nabla\phi\Big)\cdot \boldsymbol{n}\Bigg) dA \\ &\qquad + \int_{\Gamma_{\text{m}}} \Bigg(C_{\text{m}} \frac{\partial \phi^{(kl)}}{\partial t} + \sum_{i=1}^N j_i^{(kl)}\Bigg) dA. \end{aligned} \tag{25}$$

Note that (23) and (25) are equivalent to the differential equations since the finite volume Ω_{fv} is arbitrary.

In the finite volume discretization, we partition the spatial region into a finite number of finite volumes (FVs), and apply (23) and (25) on each FV. We then approximate the volume and surface integrals that appear in the integral conservation relations.

For simplicity, consider a two-dimensional situation. Discretize space into polygonal finite volumes. For each FV we designate a representative location $\boldsymbol{x}_c$ where we define the value of the physical variables. Equation (1) for ion conservation can be discretized using a finite volume approach in the following fashion.

$$\begin{aligned}\left.\frac{\partial c_i}{\partial t}\right|_{\boldsymbol{x}=\boldsymbol{x}_c} &\approx \frac{1}{V}\int_{\text{finite volume}} \frac{\partial c_i}{\partial t}\,dV = -\frac{1}{V}\int_{\text{faces of FV}} \boldsymbol{f}\cdot\boldsymbol{n}\,dA \\ &\approx -\frac{1}{V}\sum_q e_q F^q, \end{aligned} \tag{26}$$

where V is the volume (in two dimensions the area) of the FV, ℓ labels the faces (polygonal sides), and e_ℓ is the area (in two dimensions the length) of the face ℓ. F^ℓ is an approximation to the true flux $\boldsymbol{f}\cdot\boldsymbol{n}$ evaluated on face ℓ. The discrete evolution equation is thus

$$\frac{\partial c_i}{\partial t} = -\frac{1}{V}\sum_\ell e_\ell F^\ell. \tag{27}$$

Label the FVs by p and write the flux density approximation from FV p to p' as $F^{(p,p')}$. As long as $F^{(p,p')} = -F^{(p',p)}$, we have discrete conservation of c_i. Thus, it is straightforward with the finite volume method construct a conservative numerical scheme. In our case, however, there is the unusual complication that we have to account not only for ions in the interiors of the FVs but also for the ions in the space charge layers. Because of this, it will not always be the case that $F^{(p,p')} = -F^{(p',p)}$, but our scheme will be conservative anyway, as explained below.

As we shall see, the discretization of the electroneutrality condition will be obtained by multiplying (27) by qz_i and summing in i. This can be seen as a discretization of the integral charge conservation relation (25).

4.2. ***Cylindrical geometry.*** We have developed finite volume schemes adapted to two types of simulations, one for arbitrary two-dimensional membrane geometry, and the other for cylindrical geometry. We first discuss the finite volume discretization for cylindrical geometry.

Take a cylindrical coordinate system with the axial coordinate z and the radial coordinate r. We seek solutions that are axisymmetric. We discretize in r and z. We have a series of FVs whose shape is a torus with a rectangular cross-section. Each FV will generically have four faces at which it touches other FVs. FVs are indexed by p and the associated quantities of the FV p are labeled with the subscript or superscript p. Consider an FV p. Let the width of this FV in the r direction be h_p^r and that in the z direction be h_p^z. We let $h_p^r < Kh$ and $h_p^z < Kh$ for some constant K uniformly for all p and take $h \to 0$. To each FV we apply the divergence theorem and its approximation, as we did in (26) and (27).

For cylindrical geometry, we require that the membrane conform to the FV boundaries. That is to say, the membrane patches can be described by $z = \text{const}$ or $r = \text{const}$. FV faces that coincide with the membrane will be referred to as *membrane faces*. Non-membrane faces will be referred to as *ordinary faces*. To each ordered pair of FVs p and p' we associate two quantities $e^{(p,p')}$ and $\gamma^{(p,p')}$. If FVs p and p' are adjacent to each other through an ordinary face, we let $e^{(p,p')}$ be the area of this membrane face. Otherwise we set $e^{(p,p')} = 0$. Likewise, if FVs p and p' are adjacent through a membrane face, we set $\gamma^{(p,p')}$ to be its area and 0 otherwise. By definition, $e^{(p,p')} = e^{(p',p)}$ and likewise for $\gamma^{(p,p')}$. If $e^{(p,p')} \neq 0$ or $\gamma^{(p,p')} \neq 0$, we let $F^{(p,p')}$ and $G^{(p,p')}$ denote the flux density from FV p to FV p' respectively. Otherwise, we set $F^{(p,p')} = 0$ or $G^{(p,p')} = 0$. The specific forms of $F^{(p,p')}$ and $G^{(p,p')}$ will be discussed shortly. If $\gamma^{(p,p')} \neq 0$, we must define membrane associated quantities that correspond to this ordered pair. They include the gating variables $s_g^{(p,p')}$ and the membrane charge fraction $\lambda_i^{(p,p')}$. The former satisfy $s_g^{(p,p')} = s_g^{(p',p)}$ and their associated evolution equations satisfy symmetry conditions that correspond to (12) and (13). We let $\lambda_i^{(p,p')}$ denote the membrane charge fraction of the membrane patch (p, p') found on the membrane surface facing FV p. There is no symmetry relation between $\lambda_i^{(p,p')}$ and $\lambda_i^{(p',p)}$ since they are different physical entities. FVs with one or more membrane faces will be called *membrane FVs*, whereas FVs without membrane faces will be called *ordinary FVs*.

Consider FV p whose coordinates are given by $z_{p0} < z < z_{p1}$ and $r_{p0} < r < r_{p1}$. The discrete evolution equation for ionic concentrations c_i^pin FV p are

$$\frac{\partial c_i^p}{\partial t} = -\frac{1}{V_p} \sum_{p' \neq p} (e^{(p,p')} F_i^{(p,p')} + \gamma^{(p,p')} G_i^{(p,p')}), \tag{28}$$

where V_p is the volume of the FV p. We think of c_i^p and ϕ^p, the physical variables associated with the FV p, as being defined at the center of the axial cross-section of FV p. That is to say, the representative point $\boldsymbol{x}_c$ in (26) is taken to be at $r = (r_{p0} + r_{p1})/2$, and $z = (z_{p0} + z_{p1})/2$. For an ordinary FV, the second sum in (28) is 0.

All we need now in (28) are the approximate flux density expressions $F_i^{(p,p')}$ and $G_i^{(p,p')}$. For discretization of the flux density in the axial direction, we take

$$F_i^{(p,p')} = D_i \left(\frac{c_i^p - c_i^{p'}}{(h_{p'}^z + h_p^z)/2} + \frac{q z_i}{k_B T} \frac{h_{p'}^z c_i^p + h_p^z c_i^{p'}}{h_{p'}^z + h_p^z} \frac{\phi^p - \phi^{p'}}{(h_{p'}^z + h_p^z)/2} \right). \tag{29}$$

Note that this expression changes sign if p and p' are exchanged, making this a conservative discretization. We take care in constructing our mesh that the mesh width in the z direction changes smoothly as a function of the z coordinate of the representative point of the FV. For fluxes in the radial direction, we discretize in exactly the same fashion.

We now turn to the approximation to the membrane fluxes $G_i^{(p,p')}$.

$$qz_i G_i^{(p,p')} = C_{\mathrm{m}} \frac{\partial(\lambda_i^{(p,p')}\phi_{\mathrm{m}}^{(p,p')})}{\partial t} + j_i^{(p,p')}(s^m, \phi_{\mathrm{m}}^{(p,p')}, c^p, c^{p'}), \tag{30}$$

$$\phi_{\mathrm{m}}^{(\mathrm{p,p'})} = \phi^p - \phi^{p'}. \tag{31}$$

We evaluate the membrane quantities c (vector of ionic concentrations $(c_1, \dots, c_N)$) and ϕ_{m} using values at the representative points $\boldsymbol{x}_c$ of FVs p and p'. The function $j_i^{(p,p')}$ satisfies symmetry conditions equivalent to (13) so that $j_i^{(p,p')} = -j_i^{(p',p)}$. Note that in general $G_i^{(p,p')} \neq -G_i^{(p',p)}$, because of the presence of the surface charge

$$\sigma_i^{(p,p')} \equiv \lambda_i^{(p,p')} C_{\mathrm{m}} \phi_{\mathrm{m}}^{(p,p')}.$$

Our discretization is conservative nevertheless in the following sense. The quantity

$$V_p c_i^p + \frac{1}{qz_i} \sum_{p' \neq p} \gamma^{(p,p')} \sigma_i^{(p,p')} \tag{32}$$

will be conserved thanks to the symmetry conditions satisfied by $F_i^{(p,p')}$ and $j_i^{(p,p')}$. Expression (32) is the total ionic content in the FV p, taking into account the amount of ion that resides within the space charge layer.

We finally note that the discretization of the electroneutrality condition for each FV can be obtained by multiplying the discrete evolution equation (28) by qz_i and summing them over i (under the assumption that the initial configuration satisfies the electroneutrality condition):

$$0 = \sum_{i=1}^{N} qz_i \sum_{p' \neq p} (e^{(p,p')} F_i^{(p,p')} + \gamma^{(p,p')} G_i^{(p,p')}). \tag{33}$$

This is precisely the discretization of (25).

By expanding these flux approximations in Taylor series and substituting into (26), one can easily obtain the local truncation error for each FV. The local truncation error for ordinary FVs is $\mathbb{O}(h)$ and is $\mathbb{O}(1)$ for membrane FVs. We shall nonetheless observe approximate second order convergence in space in the cylindrical geometry case, as we shall see in Section 7.

As for boundary conditions at the outer rim of the computational domain, we shall make use a no-flux boundary condition.

4.3. ***Arbitrary two-dimensional membrane geometry.*** We have developed code that handles two-dimensional arbitrary membrane geometry. The ideas are the same as for the cylindrical case. We shall use an *embedded boundary method*, where a uniform Cartesian grid is used over most of the computational domain, except where the grid is cut by the membrane [13; 7; 23].

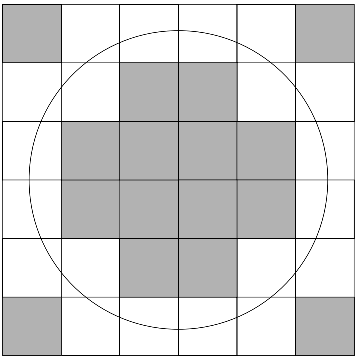
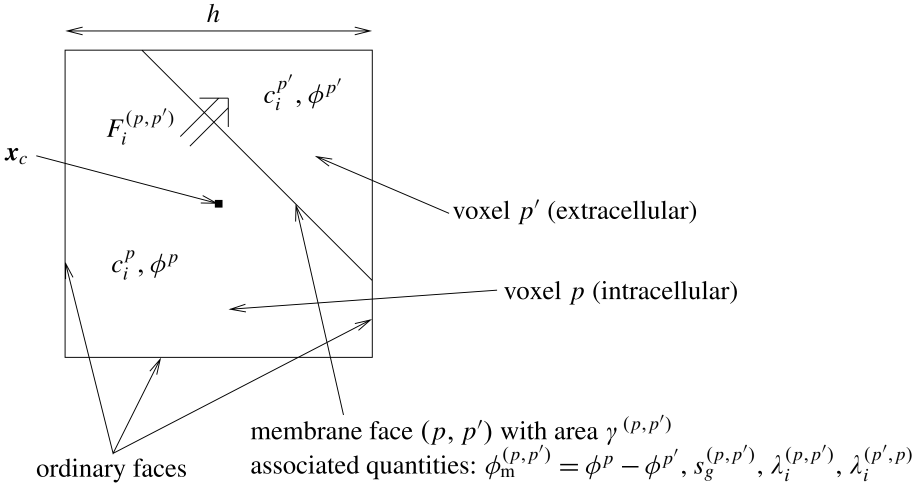


Figure 2. Grid for two-dimensional simulation. The shaded FVs are the ordinary FVs whereas the nonshaded FVs are the membrane FVs. To each membrane patch, there are two membrane FVs. In the above figure, there are 36 meshes, 16 ordinary FVs, and 40 membrane FVs.

Henceforth, we shall use the word *mesh* to denote the uniform Cartesian grid laid on the computational domain, and also to denote the square patches that result from this grid (Figure 2). We shall take the mesh sides to have length h. An FV is the same thing as a mesh if the membrane does not cut through this mesh. If the membrane does cut through the mesh, we approximate this membrane cut by a straight line, and the resulting two polygons will be our FVs that correspond to this mesh. We shall call an uncut Cartesian mesh an *ordinary* FV. An FV that is cut out of a Cartesian mesh by the membrane will be called a *membrane* FV. We shall label our FVs and their associated quantities with subscript or superscript p.

When a membrane cuts through a mesh, two FVs will be generated. These two FVs share a common membrane patch. These are the *membrane faces*, whereas other faces of the FV are the *ordinary faces*. Each face is flanked by two FVs p and p', and the faces are labeled by the ordered pair (p, p'). As in the cylindrical case, we associate with each membrane patch (p, p') its attendant physical quantities.

In the case of an ordinary FV, we shall take the representative point $\boldsymbol{x}_c$, with which the values of the physical variables are associated, to be the center of the FV. For a membrane FV, we shall take $\boldsymbol{x}_c$ to be the center of the Cartesian mesh from which the FV was cut. Thus, there will be cases in which $\boldsymbol{x}_c$ geometrically lies outside the FV. Conceptually, this involves the smooth extrapolation of a function defined on one side of the membrane to the other side of the membrane. For each membrane FV, there is another membrane FV that shares the same membrane patch which was therefore cut out of the same mesh. These two membrane FVs have representative points $\boldsymbol{x}_c$ that coincide geometrically but are computationally distinguished.



h
$F_i^{(p,p')}$
$c_i^{p'}, \phi^{p'}$
$\boldsymbol{x}_c$
voxel p' (extracellular)
c_i^p, ϕ^p
voxel p (intracellular)
membrane face (p, p') with area $\gamma^{(p,p')}$
associated quantities: $\phi_{\mathrm{m}}^{(p,p')} = \phi^p - \phi^{p'}$, $s_g^{(p,p')}$, $\lambda_i^{(p,p')}$, $\lambda_i^{(p',p)}$
ordinary faces

Figure 3. A membrane FV and its associated quantities.

The structure of the discretization exactly parallels that for the cylindrical case. The evolution equation for the concentration discretized in space will be the same as (28). We now briefly discuss approximation of the flux terms $F_i^{(p,p')}$ and $G_i^{(p,p')}$. For $F_i^{(p,p')}$, regardless of whether $e^{(p,p')} = h$ or otherwise, we shall use

$$F_i^{(p,p')} = D_i\Big(\frac{c_i^p - c_i^{p'}}{h} + \frac{qz_i(c_i^p + c_i^{p'})}{2k_BT}\frac{\phi^p - \phi^{p'}}{h}\Big). \tag{34}$$

For the membrane flux, we use the same expression as (30).

As in the cylindrical case, one can determine the local truncation error of the above scheme. For ordinary FVs, the truncation error is $\mathbb{O}(h^2)$ whereas for membrane FVs, the error is $\mathbb{O}(1)$. Nonetheless, we shall see in Section 8 that we generally observe obtain linear to supralinear convergence in space.

To handle arbitrary membrane geometry, we must generate the necessary geometry data at the membrane where the mesh is cut. We have written a custom mesh generator to perform this task. It takes the characteristic function of a region as input to generate the necessary data. The mesh generator approximates a cut by the membrane as a straight line, and cannot handle nongeneric cases of degenerate geometry. When the volume of a membrane FV is less than 10^{-5} times the volume of an ordinary mesh, this FV is ignored.

5. Temporal discretization

For our algorithm overall, we adopt an operator splitting approach. We split each time evolution step into the *gating substep* in which we update the gating variables s_g defined on the membrane followed by the *potential/concentration substep* in which we update the electrostatic potential and ionic concentrations. This splitting allows us to significantly reduce computational cost and makes the code modular by making it easy to supply the PDE system with different gating variable kinetics, which varies widely depending on the biophysical system of interest. Since the splitting error is first order, the time-stepping error should be first order overall so long as we use a first order method for the gating part and the potential/concentration part of the time stepping.

In the gating substep, we treat the gating variables implicitly while we treat the electrostatic potential and concentrations implicitly. In the potential/concentration substep, we treat the electrostatic potential and concentrations implicitly while treating the gating variables explicitly. We now discuss this latter substep in greater detail.

Recall from the discussion of Section 3 that the system of equations has two diffusive time scales. For parameter ranges of biophysical interest, the dissipative nature of membrane potential "diffusion" in particular makes it prohibitively expensive to use an explicit time-stepping scheme, rather than the ionic diffusion, as we shall see below. We saw in Section 3 that the evolution of the electrostatic potential is governed by (19) under the approximation of constant concentration:

$$\frac{\partial \phi_{\mathrm{m}}}{\partial t} + \frac{I(\phi_{\mathrm{m}})}{C_{\mathrm{m}}} = -\frac{a}{C_{\mathrm{m}}} \mathscr{L} \phi_{\mathrm{m}}, \tag{35}$$

where we have explicitly noted the dependence of I on ϕ_{m}. The behavior of $\mathscr{L}$ can be gleaned by looking at how $\mathscr{L}$ acts in the special case when Ω^{int} and Ω^{ext} are the upper and lower half spaces of $\mathbb{R}^3$ respectively. By employing Fourier analysis, we can see that the component with wave number $\boldsymbol{k}$ on the membrane is multiplied in amplitude by a factor proportional to $|\boldsymbol{k}|$. This can also be inferred by looking at the "diffusion" coefficient a/C_{m}, which has dimensions length/time. Note that this is different from the diffusion operator where the amplitude is multiplied by $|\boldsymbol{k}|^2$. This implies that as the mesh width on the membrane is made smaller, one should refine the time step proportionally to the mesh width if we are to use an explicit scheme.

For physiologically relevant systems, the "diffusion" constant in (35), a/C_{m}, is approximately equal to 10^5 μm/ms. We may thus infer that a mesh width on the order of 1 μm will necessitate a time step on the order of 10 ns if an explicit scheme is used. On the other hand, D_i is on the order of 1 $\mu\mathrm{m}^2/\mathrm{ms}$. Thus, the time step restriction imposed by ionic diffusion is much less stringent, on the order of submilliseconds. The time step restriction thus arises chiefly from membrane potential diffusion, and a time step on the order of nanoseconds is unacceptable given that biophysical phenomena of interest occur on the millisecond time scale [1; 18; 10]. For example, a single synaptic transmission event in the central nervous system, a process we believe our modeling methodology to be useful for, typically has a duration on the order of 1–10 ms [18; 15].

We note that in [29], the authors introduce a one-dimensional model of cellular electrophysiology incorporating ionic diffusion, where they use time steps as small as 1 ns to simulate their system. This small time step requirement is related to the time step restriction that would apply to an explicit scheme in our case, too, as discussed above. In [18], the author argues that this has been a major impediment in incorporating electrodiffusion of ions in modeling studies of cellular or subcellular electrophysiology.

This difficulty is overcome by treating ϕ and c_i implicitly in the potential/concentration step. The membrane potential ϕ_{m} becomes an unknown to be determined. Note that ϕ_{m} is the jump in ϕ across the cell membrane. We here have an elliptic interface problem in which we must solve for the unknown jumps across interfaces. In the context of time-dependent PDEs, similar problems arise in implicit discretizations of fluid structure interaction problems where one must solve for the unknown jump in the derivative of the velocity field across the immersed elastic interface. (see for example [22; 20]).

We label our time step by n, where n is an integer. We let the time step duration be Δt. Suppose we know values of s_g, c_i, ϕ and λ_i at time $(n-1)\Delta t$. In the gating substep, we advance s_g to find values at time n for every membrane patch.

$$\frac{s_g^{(p,p'),n} - s_g^{(p,p'),n-1}}{\Delta t} = f_g(s^{(p,p'),n}, \phi_{\mathrm{m}}^{(p,p'),n-1}, c^{p,n-1}, c^{p',n-1}). \tag{36}$$

Note that the evolution of the gating variables s_g does not involve any spatial coupling, and thus, can be solved independently for every membrane patch.

In the potential/concentration substep, we advance c_i, ϕ and λ_i. Whether we are considering cylindrical geometry or arbitrary two-dimensional membrane geometry, the semidiscretized evolution equation for c_i is (28). To discretize (28) in time, we use a backward Euler type discretization to march from time $(n-1)\Delta t$ to time $n\Delta t$, where c_i, ϕ and λ_i are treated implicitly, whereas the gating variables s_g are given quantities:

$$\begin{aligned}\frac{c_i^{p,n} - c_i^{p,n-1}}{\Delta t} &= -\frac{1}{V_p}\sum_{p'\neq p}(e^{(p,p')}F_i^{(p,p'),n}(c_i^n, \phi^n) + \gamma^{(p,p')}G_i^{(p,p'),n}),\\ qz_i G_i^{(p,p'),n} &= C_{\mathrm{m}}\Big(\frac{\lambda_i^{(p,p'),n}\phi_{\mathrm{m}}^{(p,p'),n} - \lambda_i^{(p,p'),n-1}\phi_{\mathrm{m}}^{(p,p'),n-1}}{\Delta t}\Big)\\ &\qquad + j_i^{(p,p'),n}(s^{(p,p'),n}, \phi_{\mathrm{m}}^{(p,p'),n}, c^{p,n-1}, c^{p',n-1}).\end{aligned} \tag{37}$$

Note that ϕ is treated implicitly, so that the membrane potential ϕ_{m} is an unknown to be solved for at each time step. Of the arguments of $j_i^{(p,p'),n}$, we evaluate c at time $(n-1)\Delta t$, whereas ϕ_{m} and s are evaluated at $n\Delta t$. The evolution of λ_i is given by

$$\frac{\lambda_i^{(p,p'),n}-\lambda_i^{(p,p'),n-1}}{\Delta t}=\frac{\tilde{\lambda}_i^{(p,p'),n}-\lambda_i^{(p,p'),n}}{r_d^2/D_0}, \tag{38}$$

where $\tilde{\lambda}_i^{(p,p'),n}$ is evaluated using $c_i^{p,n}$. By summing (37) over i and recalling that $\sum_i \tilde{\lambda}_i=1$, we conclude that $\sum_i \lambda_i^{p,n}$ relaxes geometrically to 1 as n increases. In particular, if this sum is equal to 1 initially it remains equal to 1 at every time step. Assuming that this is the case, we may multiply the above by qz_i and sum in i to get an equation in $\phi^{p,n}$.

$$\begin{aligned}-\frac{\rho_0^p+\sum_{i=1}^N qz_ic_i^{p,n-1}}{\Delta t}&=-\frac{1}{V_p}\sum_{p'\neq p}e^{(p,p')}\sum_{i=1}^N qz_iF_i^{(p,p'),n}\\&\quad-\frac{1}{V_p}\sum_{p'\neq p}\gamma^{(p,p')}\left(C_{\mathrm{m}}\left(\frac{\phi_{\mathrm{m}}^{(p,p'),n}-\phi_{\mathrm{m}}^{(p,p'),n-1}}{\Delta t}\right)+\sum_{i=1}^N j_i^{(p,p'),n}\right).\end{aligned} \tag{39}$$

This can be viewed as the full discretization of (33). A subtle point is that we have only made use of the electroneutrality condition $\rho_0^p+\sum_i qz_ic_i^p=0$ at time $n\Delta t$ and not at time $(n-1)\Delta t$; thus we retain the term $-(\rho_0^p+\sum_i qz_ic_i^{p,n-1})/\Delta t$ on the left hand side of (39). If electroneutrality were strictly satisfied at each time step, this term would be equal to 0. Since we cannot solve the above system of equations exactly in a numerical computation, electroneutrality is never strictly satisfied. The term $-(\rho_0^p+\sum_i qz_ic_i^{p,n-1})/\Delta t$ acts to correct deviations from electroneutrality that may have been present at time $(n-1)\Delta t$.

6. Solution of nonlinear equations

We now solve the above discretized nonlinear algebraic equations for c_i, ϕ and λ_i. At each time step, we first solve for ϕ and λ_i fixing c_i, and subsequently solve for c_i fixing ϕ and λ_i. This procedure is iterated to convergence.

We chose to use the above simpler procedure in favor of a Newton iteration for the following reasons. There two major nonlinearities in the equations: the drift term in the drift-diffusion equation and the ion channel current terms in the membrane boundary conditions. The drift term couples the concentration term c_i with the gradient of the electrostatic potential. Suppose there are N_{fv} FVs and $N(=3,4$ in computational runs presented in Section 7 but potentially much larger) ionic species of interest. A Newton iteration will require solution of a nonsymmetric linear system with $N_{\mathrm{fv}}\times(N+1)$ unknowns at each iterative step. In the simpler procedure to be explained below, all linear systems are positive symmetric (semi)definite with

N_{fv} unknowns. The complicated dependence of ion channel current terms on c_i and ϕ add further algebraic complications in generating the Jacobian matrix needed at each iteration. A possible future direction is to use the simpler solution iterative procedure adopted here as a preconditioner in a Jacobian-free Newton–Krylov framework [17].

Let $c_i^{p,n,m}$, $\phi^{p,n,m}$, $\lambda_i^{(p,p'),n,m}$ denote the m-th iterate of the solution procedure, where $m = 0, 1, 2, \ldots$. We set our initial iterate for each variable to be equal to the value of that variable at time step $n-1$:

$$c_i^{p,n,0} = c_i^{p,n-1}, \quad \phi^{p,n,0} = \phi^{p,n-1}, \quad \lambda_i^{(p,p'),n,0} = \lambda_i^{(p,p'),n-1}. \tag{40}$$

We first solve for $\phi^{p,n,m}$. We take (39) and fix the ionic concentrations to their values at the previous iteration step $c_i = c_i^{n,m-1}$ so that the only unknown is $\phi^{n,m}$.

$$\begin{aligned}
-\frac{\rho_0^p + \sum_{i=1}^N qz_i c_i^{p,n-1}}{\Delta t} = \\
&-\frac{1}{V_p}\sum_{p'\neq p} e^{(p,p')}\sum_{i=1}^N qz_i F_i^{(p,p')}(c_i^{n,m-1}, \phi^{n,m}) \\
&-\frac{1}{V_p}\sum_{p'\neq p} \gamma^{(p,p')} C_m\left(\frac{\phi_{\text{m}}^{(p,p'),n,m} - \phi_{\text{m}}^{(p,p'),n-1}}{\Delta t}\right) \\
&-\frac{1}{V_p}\sum_{p'\neq p} \gamma^{(p,p')}\sum_i j_i^{(p,p'),n,m}(s^{n,m}, \phi_{\text{m}}^{(p,p'),n,m}, c^{n-1}).
\end{aligned} \tag{41}$$

By evaluating c_i at $c_i^{n,m-1}$ in the flux term $F_i^{(p,p')}$, we avoid dealing with the nonlinearity that arises from the drift term. The only possibility for a nonlinearity in the above is in the transmembrane current term. In many applications, j_i is assumed linear in ϕ_{m}. If not, we linearize as follows:

We first recall the functional form of transmembrane current terms. The general functional form of ion channel currents is written as

$$j_i = \sum_\alpha j_{i,\alpha}, \quad j_{i,\alpha} = g_{i,\alpha}(\boldsymbol{x}, s, \phi_{\text{m}}, c^{(k)}, c^{(l)})\mathcal{I}_{i,\alpha}(\phi_{\text{m}}, c_i^{(k)}, c_i^{(l)}), \tag{42}$$

where α labels the types of ion channels present, and $j_{i,\alpha}$ the transmembrane current through ion channels of this type. $g_{i,\alpha}$ is the density of the such open ion channels per unit area of membrane, and $\mathcal{I}_{i,\alpha}$ is the instantaneous current voltage relationship of a single open channel. We choose a suitable linearization of the instantaneous current voltage relation with respect to $\phi_{\text{m}}^{(p,p')}$ around $\phi_{\text{m}}^{(p,p'),n,m-1}$:

$$\mathscr{I}^{L}_{i,\alpha}(\phi_{\mathrm{m}}^{(p,p'),n,m}, c_i^{p}, c_i^{p'}) = \mathscr{D}_{\mathscr{I}_{i,\alpha}}(\phi_{\mathrm{m}}^{p,n,m-1}, c_i^{p}, c_i^{p'})(\phi_{\mathrm{m}}^{(p,p'),n,m} - \phi_{\mathrm{m}}^{(p,p'),n,m-1}) + \mathscr{I}_{i,\alpha}(\phi_{\mathrm{m}}^{(p,p'),n,m-1}, c_i^{p'}, c_i^{p'}). \quad (43)$$

The term $\mathscr{D}_{\mathscr{I}_{i,\alpha}}$ will typically be the derivative of $I_{i,\alpha}$ with respect to ϕ^{m}. Instead of j_i itself, we shall therefore use the following linearization in its place in (41).

$$\begin{aligned} j_i^{L,p,n,m} &= \sum_{\alpha} j_{i,\alpha}^{L,p,n,m}, \\ j_{i,\alpha}^{L,n,p,m} &= g_{i,\alpha}(s^{(p,p'),n}, \phi_{\mathrm{m}}^{(p,p'),n-1}, c^{n-1})\mathscr{I}^{L}_{i,\alpha}(\phi_{\mathrm{m}}^{(p,p'),n,m}, c_i^{n-1}). \end{aligned} \quad (44)$$

Note that $s^{(p,p'),n}$ is already a known quantity; we do not have to solve for it.

The result is a linear equation in ϕ, which can now be solved.

Solving for $\lambda_i^{(p,p'),n,m}$ is simple:

$$\frac{\lambda_i^{(p,p'),n,r} - \lambda_i^{(p,p'),n-1}}{\Delta t} = \frac{\tilde{\lambda}_i^{(p,p'),n,m-1} - \lambda_i^{(p,p'),n,m}}{r_d^2/D_0}, \quad (45)$$

where $\tilde{\lambda}_i^{(p,p'),n,m-1}$ is evaluated using $c_i^{p,n,m-1}$.

Given $\phi^{p,n,m}$ and $\lambda_i^{(p,p'),n,m}$, we solve for $c_i^{p,n,m}$ as follows:

$$\frac{c_i^{p,n,m} - c_i^{p,n-1}}{\Delta t} = -\frac{1}{V_p}\sum_{p' \neq p}(e^{(p,p')}F_i^{(p,p'),n,m} + \gamma^{(p,p')}G_i^{(p,p'),n,m}), \quad (46)$$

where the flux density expressions are given by

$$\begin{aligned} F_i^{(p,p'),n,m} &= D_i\Big(\frac{c_i^{p,n,m} - c_i^{p',n,m}}{h}\Big) \\ &\quad + D_i\frac{qz_i}{k_BT}\Big(\frac{c_i^{p,n,m-1} + c_i^{p',n,m-1}}{2}\Big)\Big(\frac{\phi^{p,n,m-1} - \phi^{p',n,m-1}}{h}\Big), \quad (47) \\ qz_iG_i^{(p,p'),n,m} &= C_{\mathrm{m}}\Big(\frac{\lambda_i^{(p,p'),n,m}\phi_{\mathrm{m}}^{(p,p'),n,m} - \lambda_i^{(p,p'),n-1}\phi_{\mathrm{m}}^{(p,p'),n-1}}{\Delta t}\Big) + j_i^{(p,p'),n,m}, \end{aligned}$$

where the flux expression (47) is to be suitably modified when dealing with nonuniform meshes; see (29). In the above (47), the diffusive flux is treated implicitly, whereas drift flux is left explicit. The rationale for this difference in treatment of the two flux terms is that the diffusive flux involves derivatives of c_i but the drift flux does not.

With the above expression for $F_i^{(p,p'),n,m}$, (46) is a linear equation in c_i. In fact, this is just a familiar discretization of the diffusion equation with a source term and flux boundary conditions.

We now iterate this procedure in m a suitable number of times, and set the final iterate to be the values at time n. Note that one iteration is enough to obtain a first order scheme in time. We also point out that the scheme is conservative in exact arithmetic: we have ion conservation regardless of how may iterations we perform.

We iterate so that electroneutrality is better satisfied at time n. Multiplying (46) with qz_i and summing in i does not reproduce (41) because the concentrations in the flux approximation F_i^q are evaluated using different values in the two expressions. Therefore, the solution to (46) only satisfies electroneutrality in the limit $m \to \infty$.

Our termination criterion for the above iteration is to check whether the electroneutrality condition is satisfied to within a certain tolerance after the r-th iteration. We use the following criterion:

$$\frac{\sum_p V_p |\rho_0 + \sum_i qz_i c_i^{p,n,m}|}{\sum_p V_p} < \epsilon_{\text{tol}} q c_0. \tag{48}$$

In all computations, we take $\epsilon_{\text{tol}} = 1 \times 10^{-5}$ and $c_0 = 100\,\text{mmol/l}$, the typical ionic concentration. We set this final iterate to be the value of c_i, ϕ at the next time step, except for the adjustment we discuss below.

When we use no-flux boundary conditions at the outer rim of the computational domain, we perform the following adjustment at the end of each computational step, in order to correct for the nonconservation of ions that is purely the result of round-off error. We fix the concentrations so that the global amount of each ionic species is conserved as strictly as possible by setting

$$c_i^{p,n} = \left(\frac{Q_i^{\text{init}} - \Lambda_i^{n,m}}{Q_i^{n,m}}\right) c_i^{p,n,m}, \tag{49}$$

where

$$Q_i^{n,m} = \sum_p V_p c_i^{p,n,m}, \qquad \Lambda_i^{n,m} = \frac{1}{qz_i} \sum_{(p,p'),p\neq p'} \gamma^{(p,p')} \lambda_i^{(p,p'),n,m} C_{\text{m}} \phi_{\text{m}}^{(p,p'),n,m}.$$

The summation in the definition of $\Lambda_i^{n,m}$ is over all ordered pairs (p, p'). The index m denotes the final iterate, that is, the result before this adjustment is made. The term Q_i^{init} is the total amount of the i-th ion at the initial time:

$$Q_i^{\text{init}} = \sum_p V_p c_i^{p,0} + \frac{1}{qz_i} \sum_{(p,p'),p\neq p'} \gamma^{(p,p')} \lambda_i^{(p,p'),0} C_{\text{m}} \phi_{\text{m}}^{(p,p'),0}. \tag{50}$$

The first sum represents the ions in the bulk solution, whereas the second term is the contribution from the membrane surface charge.

Why do we need to perform this fix when we know that the scheme is in fact conservative? The unfortunate reality, however, is that the scheme is conservative only in exact arithmetic. With floating point arithmetic, errors tend to accumulate and, with time, ion conservation is violated. Computational experiments indicate that this error is negligible as far as the values of c_i are concerned. This has to be corrected nonetheless because this small violation leads to global charge accumulation, which in turn leads to nonconvergence of Krylov iterations for ϕ when no-flux boundary conditions are used (for no-flux boundary conditions, we perform Krylov iterations in the subspace spanned by all grid functions that integrate to 0 over the spatial domain, and global charge accumulation leads to nonexistence of solutions, as can be seen by considering the Fredholm alternative). The scheme presented above has an inherent mechanism to eliminate *local* charge accumulation (39), but cannot eliminate *global* charge accumulation. The above adjustment is on the order of round-off error at each time step.

We note that this fix is only necessary for the no-flux boundary condition. When Dirichlet or mixed boundary conditions are imposed at the outer rim of the computational domain, global accumulation in charge in the computational domain will eventually dissipate through communication with the outer bath.

The solution to the nonlinear algebraic equations requires the solution of a linear system at each iteration. We note that solving for the electrostatic potential as well as the concentrations involve solving a positive definite symmetric system. We thus either use a direct solver(Cholesky decomposition) or the conjugate gradient method [36]. The code for cylindrical geometry has been implemented using Matlab, where we use a direct solver. The code for general two-dimensional geometry has been written in C++, where we use PETSc for the linear algebra routines [2]. PETSc is a package that provides sparse linear solvers and is designed to be suitable for parallel algorithms. Although we do not yet use parallel machines, having coded in PETSc should facilitate this transition in the future.

7. Convergence study: cylindrical geometry

We test the convergence for the cylindrical case for two kinds of situations, the standard Hodgkin–Huxley axon [11; 16; 18], and for a cardiac model of ephaptic coupling.

7.1. ***Hodgkin–Huxley axon.*** The neuronal axon is the standard biological system to which the cable model is applied. We take this system as our first test case.

For ionic channel parameters, we shall use those of the standard Hodgkin–Huxley model. There are several parameters that are required for computation with the electroneutral model but not with the cable model. They are the diffusion coefficients for each ionic species and the initial concentration of the ions. We consider three ionic species Na^+, Cl^-, and K^+. The initial concentrations and the diffusion coefficients we use are listed in Table 1. The membrane charge ratios λ_i

T	Absolute temperature	273.15+37 K
D_{Na^+}	Diffusion coefficient of Na^+	1.33 $\mu\mathrm{m}^2/\mathrm{ms}$ [18]
D_{K^+}	Diffusion coefficient of K^+	1.96 $\mu\mathrm{m}^2/\mathrm{ms}$ [18]
D_{Cl^-}	Diffusion coefficient of Cl^-	2.03 $\mu\mathrm{m}^2/\mathrm{ms}$ [18]
$c^{\mathrm{int}}_{\mathrm{Na}^+}\big\vert_{t=0}$	Initial intracellular concentration of Na^+	10 mmol/l
$c^{\mathrm{ext}}_{\mathrm{Na}^+}\big\vert_{t=0}$	Initial extracellular concentration of Na^+	145 mmol/l
$c^{\mathrm{int}}_{\mathrm{K}^+}\big\vert_{t=0}$	Initial intracellular concentration of K^+	140 mmol/l
$c^{\mathrm{ext}}_{\mathrm{K}^+}\big\vert_{t=0}$	Initial extracellular concentration of K^+	5 mmol/l
$c^{\mathrm{int}}_{\mathrm{Cl}^-}\big\vert_{t=0}$	Initial intracellular concentration of Cl^-	150 mmol/l
$c^{\mathrm{ext}}_{\mathrm{Cl}^-}\big\vert_{t=0}$	Initial extracellular concentration of Cl^-	150 mmol/l
$\phi_{\mathrm{m}}\vert_{t=0}$	Initial transmembrane potential, $\phi^{\mathrm{int}} - \phi^{\mathrm{ext}}$	-70 mV

Table 1. Parameter values used in the Hodgkin–Huxley simulations of the axon.

are initialized so that $\lambda_i\big|_{t=0} = \tilde{\lambda}_i\big|_{t=0}$. The immobile charge density was taken so that electroneutrality is satisfied at each spatial point at $t = 0$. The Hodgkin–Huxley model has one free parameter, the value of the equilibrium potential [16], which we take to be -70 mV. The initial value of the gating variables are set to the equilibrium values at -70 mV.

We take the axon to be a cylinder of radius $l\,\mu$m and axial length $l_A\,\mu$m. Take the z axis along the axis of the cylinder, with the axonal ends at $z = \pm l_A/2$, and the radial axis r from the center of the cylinder. The cylindrical axon is bathed in an extracellular medium located between the cell membrane at $r = l$ and $r = 2l$, where we impose no-flux boundary conditions. We also impose no-flux boundary conditions at $z = \pm l_A/2$. The total simulation time be T_e. We choose the diameter $2l$ and the axial length l_A to be

$$2l = 0.1, 1, 10\,\mu\mathrm{m}, \quad l_A = 4\sqrt{2l} \times 10^3\,\mu\mathrm{m}, \quad T_e = 4\,\mathrm{ms}. \tag{51}$$

We note that the axonal length is much greater than the radial length. This length of the axon l_A was chosen so that we can see a wave of propagating action potential. This choice also roughly corresponds to the action propagation speed seen in unmyelinated neuronal axons on the order of 10 mm/ms at an axonal diameter of 1–10 μm. We use the above dependence of l_A on l since, according to cable theory, the electrotonic length (the typical length scale for the spread of membrane potential) scales with the square root of the axonal diameter.

At time $t = 0$, we initiate an action potential by transiently increasing the Cl^- conductance, for which we specify the following spatial distribution and time dependence:

$$G_{\mathrm{Cl}^-} = \begin{cases} 5\Big(1+\cos\dfrac{12\pi z}{l_A}\Big)\Big(1-\cos\dfrac{2\pi t}{T_s}\Big) & \text{if } -\dfrac{l_A}{12} < z < \dfrac{l_A}{12},\ t < T_s = 1\,\mathrm{ms}, \\ 0 & \text{otherwise.} \end{cases}$$

We thus give a brief change in the membrane chloride conductance at the center of the axon. An action potential is initiated here and spreads towards the two ends of the axon. A snapshot from a sample run where the diameter $2l = 1\ \mu$m is shown in Figures 4 and 5.

In the case of the cylindrical axon, the computational runs exhibit little radial variation in the electrostatic potential, and one may argue against the use of this computationally intensive model in place of much simpler models such as the cable model. In Section 7.2, we shall see a case in which a radial variation in the electrostatic potential is seen. Even in the case of a cylindrical axon, however, we

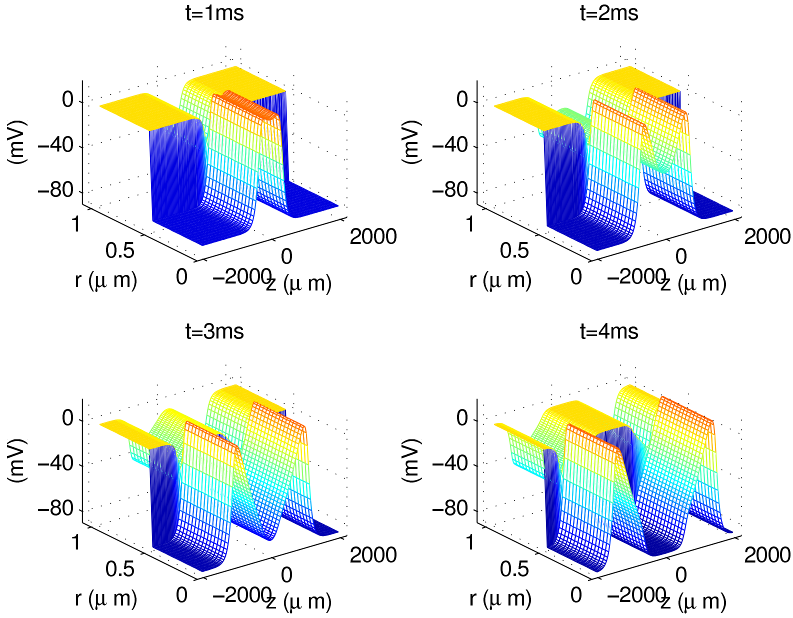


Figure 4. Electrostatic potential ϕ at $t = 1, 2, 3, 4$ ms, $2l = 1\ \mu$m. Since the solutions we seek are radially symmetric, the radial cross-section (r between $0\ \mu$m and $1\ \mu$m) is plotted in the graph. The jump discontinuity at $r = 0.5\ \mu$m signifies the jump in the electrostatic potential. The mesh size is $N_z \times N_r = 128 \times 32$. Potential is measured in mV and length in μm.

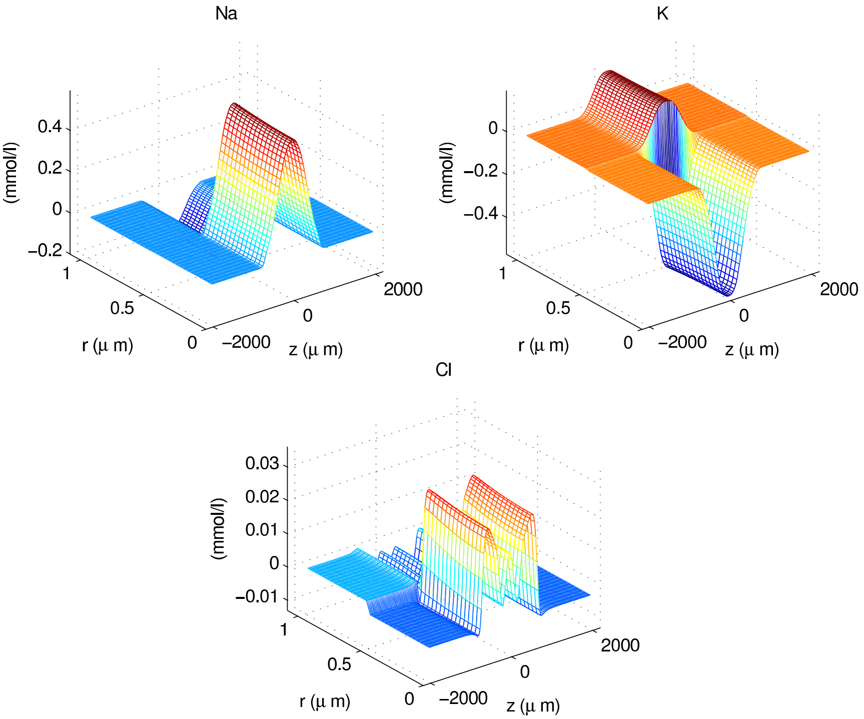


Figure 5. Cumulative *change* in ionic concentrations from $t = 0$ at $t = 2\,\text{ms}$, $2l = 1\,\mu\text{m}$. The cumulative concentration changes in Na^+, K^+, and Cl^- are shown. As with the previous figure, the radial cross section is plotted. The mesh size is $N_z \times N_r = 128 \times 32$. Concentration is measured in mmol/l and length in μm.

believe that this model can be useful in the following respects. First, it allows us to track ionic concentration, whose evolution cannot be determined without solving for the electrostatic potential which ensures that electroneutrality be satisfied pointwise in space. The constraint of electroneutrality may give rise not only to quantitative but also qualitatively different behavior compared to simple diffusion [32]. Second, this model can be used as a validation tool to judge when the cable model is a good approximation [18; 33]. We hope to make a more detailed comparison between the cable model and our model in a future publication.

Convergence in space. We take a uniform grid of $N_z \times N_r$ over the simulation domain. We set

$$N_z = 64 \times 2^{n-1}, \quad N_r = 16 \times 2^{n-1}, \quad \Delta t = 0.02\,\text{ms}, \quad N_T = \frac{T_e}{\Delta t} = 200, \quad (52)$$

where $n = 1, \dots, 4$. Note that here and throughout the paper the time step remains fixed in our spatial convergence studies. The possibility of proceeding in this way without encountering numerical instability is conceptually related to the unconditional stability of our implicit computational scheme. During spatial grid refinement, the bounds on spatial difference operators grow because of the appearance of the mesh width in the denominators of the difference operators. This reflects the unbounded nature of the corresponding differential operators. In an explicit scheme, the growth of the operator norms needs to be compensated by refinement of the time step, but we do not have to do that here.

To measure the convergence rate, we define the discrete p-norm as

$$\|u\|_{L^p} = \Big(\sum_{k=1}^{2N_r} |V_k||u_k|^p\Big)^{1/p}, \quad 1 \le p < \infty, \quad \|u\|_{L^\infty} = \max_k |u_k|. \tag{53}$$

The convergence rate is measured by comparing the interpolation of the numerical solution at a finer level to the numerical solution at a coarser level. Let c_i computed with an $N_r \times N_r$ mesh be written as $c_i^{N_r}$. We define a measure of error $e_p^s[c_i; N_r]$ as follows.

$$e_p^s[c_i; N_r] = \big\|c_i^{N_r} - \mathcal{I}^{2N_r \to N_r} c_i^{2N_r}\big\|_{L^p}. \tag{54}$$

Here, $\mathcal{I}^{2N_r \to N_r}$ is an interpolation operator from the finer to the coarser grid.

For the electrostatic potential ϕ, we need to take into account the arbitrariness of ϕ, up to addition of a constant. Thus, we measure the error in ϕ as

$$e_p^s[\phi; N_r] = \min_{c_\phi \in \mathbb{R}} \big\|\phi^{N_r} - \mathcal{I}^{2N_r \to N_r} \phi^{2N_r} - c_\phi\big\|_{L^p}. \tag{55}$$

As an empirical measure of convergence rate in space, we use

$$r_p^s[\psi; N_r] = \log_2\Big(\frac{e_p^s[\psi; N_r]}{e_p^s[\psi; 2N_r]}\Big), \tag{56}$$

where ψ can be either c_i or ϕ.

Table 2 lists the rate of convergence for both c_i and ϕ at the three diameters with three norms, L^1, L^2 and L^∞, at time $t = 4$ ms. Convergence rates at other time points were similar.

We see second order convergence for most parameter regions considered. The second order convergence observed here is, however, lost in the case of general two-dimensional geometry (Section 8). This favorable property is thus tied to the fact that the membrane geometry conforms to the underlying Cartesian grid. The deterioration in convergence rate when the axonal diameter is equal to 10 μm seems attributable to the fact that the concentration gradients near the membrane are not fully resolved when $N_r = 32$. Since the mesh has been scaled with the axon size, the largest axon also has the coarsest mesh, in absolute terms. This affects the

diameter	norm	$r_p^s[c_1,32]$	$r_p^s[c_2,32]$	$r_p^s[c_3,32]$	$r_p^s[\phi,32]$
0.1	L^1	1.97	1.97	1.93	1.96
	L^2	1.97	1.97	1.94	1.98
	L^∞	1.97	1.97	1.78	1.90
1	L^1	1.97	1.97	1.93	1.96
	L^2	1.97	1.97	1.95	1.97
	L^∞	1.97	1.97	1.88	1.82
10	L^1	1.97	1.97	1.92	1.97
	L^2	1.96	1.97	1.84	1.98
	L^∞	1.44	1.80	1.41	1.82

Table 2. Convergence rate in space (r_p^s) for different axonal diameters. Values computed at $t = 4\,\text{ms}$, and $N_r = 32$.

quality of the computed solution because the radial concentration profiles do *not* scale with the size of the axon.

Convergence in time. Convergence in time is measured similarly to the spatial case. We vary the time step so that

$$\Delta t = 0.04 \times 2^{1-n}, \quad N_T \equiv \frac{T_e}{\Delta t} = 100 \times 2^{n-1}, \tag{57}$$

where $n = 1, \dots, 4$. We take $N_r = 32$ as our spatial grid to assess time convergence.

The convergence rate and error is computed analogously to the spatial case.

$$e_p^t[c_i;\, N_r] = \left\| c_i^{N_T} - \mathcal{I}^{2N_T \to N_T} c_i^{2N_T} \right\|_{L^p}. \tag{58}$$

Here, $\mathcal{I}^{2N_T \to N_T}$ is an interpolation operator from the finer to the coarser time step. For the electrostatic potential ϕ, we let

$$e_p^t[\phi;\, N_T] = \min_{c_\phi \in \mathbb{R}} \left\| \phi^{N_T} - \mathcal{I}^{2N_T \to N_T} \phi^{2N_T} - c_\phi \right\|_{L^p}. \tag{59}$$

As an empirical measure of convergence rate in time, we use

$$r_p^t[\psi;\, N_T] = \log_2 \frac{e_p^t[\psi;\, N_T]}{e_p^t[\psi;\, 2N_T]}. \tag{60}$$

where ψ can be either c_i or ϕ.

Table 3 lists the rate of convergence for both c_i and ϕ at the three diameters with three norms, L^1, L^2 and L^∞ at $t = 4\,\text{ms}$. We see approximate first order convergence in time over all parameter ranges considered, although the convergence rate is slightly sublinear overall. The source of this sublinear convergence rate is unclear.

diameter	norm	$r_p^t[c_1, 200]$	$r_p^t[c_2, 200]$	$r_p^t[c_3, 200]$	$r_p^t[\phi, 200]$
0.1	L^1	0.93	0.93	0.93	0.92
	L^2	0.94	0.94	0.90	0.89
	L^∞	0.96	0.96	0.90	0.83
1	L^1	0.93	0.93	0.93	0.92
	L^2	0.94	0.94	0.89	0.89
	L^∞	0.95	0.96	0.75	0.81
10	L^1	0.93	0.93	0.91	0.92
	L^2	0.93	0.94	0.86	0.89
	L^∞	0.77	0.95	0.75	0.82

Table 3. Convergence rate in time (r_p^t) for different axonal diameters. Values computed at $t = 4$ ms, and $N_T = 200$.

Convergence in space and time. We next refine in both space and time to demonstrate that the approximation approaches the solution to the PDE system. Given that we observe second order convergence in space and first order convergence in time, we should be able to observe second order convergence overall if we make the time step proportional to the square of the mesh width. We let

$$N_z = 4 \times N_r, \quad N_r = 32 \times 2^{n-1}, \quad \Delta t = 0.02 \times 4^{1-n}, \quad N_T \equiv \frac{T_e}{\Delta t} = 200 \times 4^{n-1},$$

for $n = 1, \ldots, 3$. The spatiotemporal convergence rate r_p^{st} is measured similarly to the empirical spatial and temporal rates r_p^s and r_p^t defined in (56) and (60). Table 4 exhibits approximate second order convergence overall.

diameter	norm	$r_p^{st}[c_1, 32]$	$r_p^{st}[c_2, 32]$	$r_p^{st}[c_3, 32]$	$r_p^{st}[\phi, 32]$
0.1	L^1	1.94	1.94	1.93	1.93
	L^2	1.93	1.95	1.89	1.90
	L^∞	1.95	1.95	1.80	1.86
1	L^1	1.94	1.94	1.93	1.93
	L^2	1.94	1.95	1.89	1.90
	L^∞	1.95	1.96	1.82	1.86
10	L^1	1.94	1.94	1.90	1.93
	L^2	1.93	1.94	1.82	1.90
	L^∞	1.57	1.87	1.50	1.86

Table 4. Convergence rate in space and time (r_p^{st}) for different values of axonal diameters. Values computed at $t = 4$ ms, and $N_r = 32$, $N_T = 200$.

7.2. ***Cardiac geometry.*** We next consider a test geometry based on cardiac microscopic anatomy [35; 19; 26]. The motivation for this test case is the following. Cardiac tissue is composed of muscle cells which are linked to one another through gap junctions, pore forming proteins similar to ion channels that straddle two adjacent cell membranes. These low resistance passage ways of electric current have conventionally been regarded as essential for successful cell-to-cell propagation of the cardiac electric signal, which in turn coordinates the synchronization of the heart beat [31]. Recent experimental as well as theoretical studies suggest, however, that gap junctions are not absolutely essential for propagation of the electric signal. Indeed, knock-out mice that do not express the principal gap junction isoforms in cardiac cells do produce a functional heart beat [37; 9]. One hypothesis that attempts to explain this anomalous conduction is the ephaptic hypothesis, in which two adjacent cardiac cells interact with one another through the very narrow cleft (the intercalating disc) between them [19]. The presence of the narrow cleft raises the possibility of steep voltage gradients and large ionic concentration changes, and is thus an ideal system in which our model could make interesting physiological predictions. This program has been partially carried out in [26], to which we refer the reader for further physiological discussion.

As a testbed, we consider 2 cells of equal length separated by a narrow intercellular space of width l_g. In fact, we consider two "half" cells, each of length $l_A/2$ as we shall see shortly. The radius of the cell is l and the whole system is bathed in an extracellular medium contained within a cylinder of radius $2l$ (Figure 6).

Similarly to the axonal case, we take z to be the axial direction and r to be the radial coordinate. We take the origin to be in the middle of the gap. Formally, the intracellular region can be written as

$$\left(-\frac{l_A+l_g}{2} < z < -\frac{l_g}{2} \quad \text{or} \quad \frac{l_g}{2} < z < \frac{l_A+l_g}{2}\right) \quad \text{and} \quad r < l. \tag{61}$$

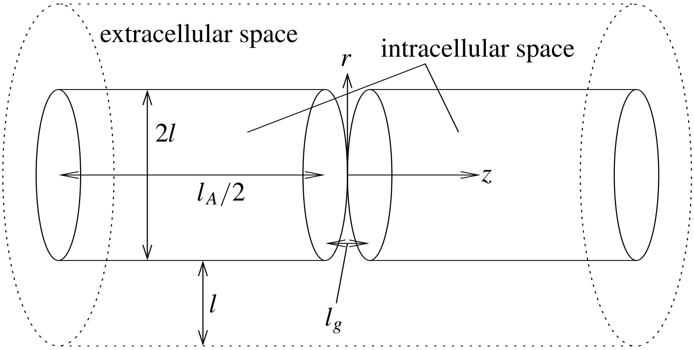


Figure 6. Schematic of the geometry used for the cardiac model.

T	Absolute temperature	$273.15+37$ K
$D^{\mathrm{us}}_{\mathrm{Na}^+}$	Unscaled diffusion coefficient of Na^+	$1.33\ \mu\mathrm{m}^2/\mathrm{ms}$
$D^{\mathrm{us}}_{\mathrm{K}^+}$	Unscaled diffusion coefficient of K^+	$1.96\ \mu\mathrm{m}^2/\mathrm{ms}$
$D^{\mathrm{us}}_{\mathrm{Ca}^{2+}}$	Unscaled diffusion coefficient of Ca^{2+}	$0.3\ \ \mu\mathrm{m}^2/\mathrm{ms}$
$D^{\mathrm{us}}_{\mathrm{Cl}^-}$	Unscaled diffusion coefficient of Cl^-	$2.03\ \mu\mathrm{m}^2/\mathrm{ms}$
$c^{\mathrm{int}}_{\mathrm{Na}^+}\big\|_{t=0}$	Initial intracellular concentration of Na^+	10 mmol/l
$c^{\mathrm{ext}}_{\mathrm{Na}^+}\big\|_{t=0}$	Initial extracellular concentration of Na^+	145 mmol/l
$c^{\mathrm{int}}_{\mathrm{K}^+}\big\|_{t=0}$	Initial intracellular concentration of K^+	140 mmol/l
$c^{\mathrm{ext}}_{\mathrm{K}^+}\big\|_{t=0}$	Initial extracellular concentration of K^+	5 mmol/l
$c^{\mathrm{int}}_{\mathrm{Cl}^-}\big\|_{t=0}$	Initial intracellular concentration of Cl^-	10 mmol/l
$c^{\mathrm{ext}}_{\mathrm{Cl}^-}\big\|_{t=0}$	Initial extracellular concentration of Cl^-	see text
$c^{\mathrm{int}}_{\mathrm{Ca}^{2+}}\big\|_{t=0}$	Initial intracellular concentration of Ca^{2+}	$0.4\ \mu$mol/l
$c^{\mathrm{ext}}_{\mathrm{Ca}^{2+}}\big\|_{t=0}$	Initial extracellular concentration of Ca^{2+}	2 mmol/l
$\phi_{\mathrm{m}}\|_{t=0}$	Initial transmembrane potential, $\phi^{\mathrm{int}}-\phi^{\mathrm{ext}}$	-90 mV

Table 5. Parameter values used in cardiac simulation.

The intracellular region is open-ended at $z=\pm\frac{1}{2}(l_A+l_g)$. This is what we mean by "half cell". We impose no-flux boundary conditions at $z=\pm\frac{1}{2}(l_A+l_g)$ and at $r=2l$.

The values for l_g, l_A and l are

$$l_g=20\,\mathrm{nm},\quad l_A=100\,\mu\mathrm{m},\quad l=11\,\mu\mathrm{m}. \tag{62}$$

We note that l_g is about 4 orders of magnitude smaller than l_A, and thus we use a nonuniform mesh, the details of which we shall describe shortly.

We consider 4 ion types in the calculation, Na^+, K^+, Ca^{2+} and Cl^-. The initial condition for all ionic species except Cl^- in the extracellular space are listed in Table 5. In the intracellular medium, we set the fixed negative charge density ρ_0 so that electroneutrality is satisfied everywhere. In the gap we introduce a nonuniform fixed negative charge density. This represents the charged groups on extracellular macromolecules that may be present within the gap. We initialize the fixed charge density ρ_0 in the extracellular space to be

$$\rho_0=\begin{cases}-\big(54+50\big(1-(r/l)^2\big)\big)\ \mathrm{mmol/l} & \text{if } r<l \text{ and } -l_g/2<z<l_g/2,\\ -54\ \mathrm{mmol/l} & \text{if } r\geq l.\end{cases}$$

Set the extracellular Cl$^-$ concentration so that the electroneutrality condition is satisfied everywhere:

$$c_{\mathrm{Cl}^-}^{\mathrm{ext}}\Big|_{t=0} = \begin{cases} 100 - 50\bigl(1-(r/l)^2\bigr)\,\mathrm{mmol/l} & \text{if } r < l \text{ and } -l_g/2 < z < l_g/2, \\ 100\,\mathrm{mmol/l} & \text{if } r \geq l. \end{cases}$$

The diffusion coefficients are adjusted in the following way. If we ignore ionic diffusion, electric current is solely driven by the gradient of the electrostatic potential. In this case, the ohmic cytoplasmic conductance is given by $a(\boldsymbol{x}, t)$ defined in (5). If one computes the cytoplasmic or extracellular conductance using (5) according to values of D_i in an aqueous solution, the values used in Table 1, we obtain an overestimate which deviates from the experimentally observed value by a factor of 2–5 [19]. We thus scale the diffusion coefficient in aqueous solution by a uniform factor α so that the cytoplasmic or extracellular conductance calculated above is approximately within the experimental range. More concretely, we let

$$g^{\mathrm{observed}} = \alpha \overline{\sum_{i=1}^{N} \frac{(qz_i)^2\, c_i|_{t=0}}{k_B T} D_i^{\mathrm{us}}}, \tag{63}$$

where g^{observed} is the cytoplasmic conductance, which we take to be equal to the extracellular conductance, D_i^{us} is the unscaled diffusion coefficient, and overline denotes averaging over the computational domain. Following [19], we let $1/g^{\mathrm{observed}} = 150\,\Omega\,\mathrm{cm}$.

For the ion channel composition for the membrane, we use the model of Bernus et al. [4], in which the authors model the electrical activity of human ventricular myocytes. The only change we make concerns the localization of the Na$^+$ channels. We concentrate their distribution so that 99% of the total Na$^+$ conductance sits at the membranes facing the gap. Evidence for such localization of Na$^+$ channel expression has been presented in [19]. This may allow an action potential to propagate across the gap without the two intracellular spaces being directly connected by gap junctions forming a cytoplasmic bridge.

All instantaneous current voltage relations for ionic channels in the model of Bernus et al. are linear in the transmembrane voltage. We do not have to linearize the current voltage relationship to obtain a linear system. The ionic pump currents are nonlinear in the transmembrane voltage, but will be treated explicitly. This does not result in numerical instabilities because ionic pump currents are typically small in magnitude.

We simulate this system for time $T_e = 4\,\mathrm{ms}$. We add a transient excitation to the system by way of an increase in Na$^+$ conductance distributed along the lateral cell membrane of the cell on the left according to

$$G^{\mathrm{add}}_{\mathrm{Na}^+} = \begin{cases} \frac{5}{4}\left(1+\cos\frac{\pi(z+L_z)}{l_A/2}\right)\left(1-\cos\frac{2\pi t}{\tau_e}\right) & \text{if } z < -\frac{l_g}{2},\ t < \tau_e, \\ 0 & \text{otherwise,} \end{cases}$$

where $L_z = (l_A + l_g)/2$ and $\tau_e = 1$ ms. Thus, we stimulate the system at one end of the cell located in $z < 0$, and see whether the action potential propagates into the next cell. Snapshots from this simulation are shown in Figures 7 and 8. We note a radial gradient in the electrostatic potential in the thin gap spaces, an effect that cannot be modeled with a simple use of the cable model. Note in these figures that the action potential propagates across a thin gap between two cells even if there are no gap junctions (low resistance connections) that connect the two cells. The feasibility of such *ephaptic* transmission (a term borrowed from neuroscience [12]), in the context of cardiac action potential propagation has been a subject of much debate [19; 35]. We have used this model to explore the biophysics of this mechanism in [26].

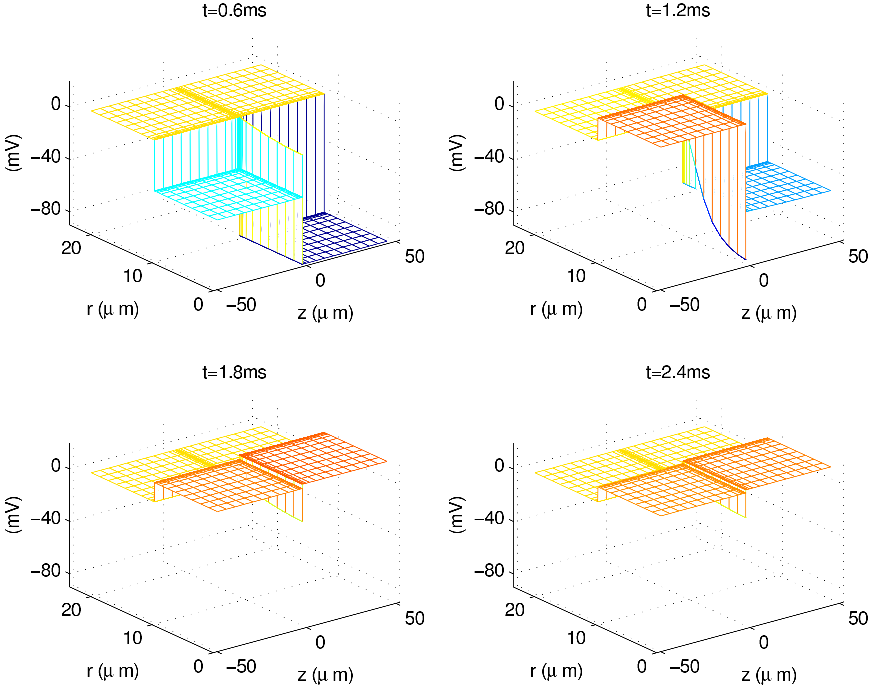


Figure 7. The evolution of the electrostatic potential in the cardiac simulation with variable mesh width. The radial cross section (r from $0\,\mu$m to $20\,\mu$m) is shown. Snapshots shown at $t = 0.6, 1.2, 1.8, 2.4$ ms. The mesh size is $N_z \times N_r = 48 \times 32$ in this computation. Potentials are in mV and lengths in μm.

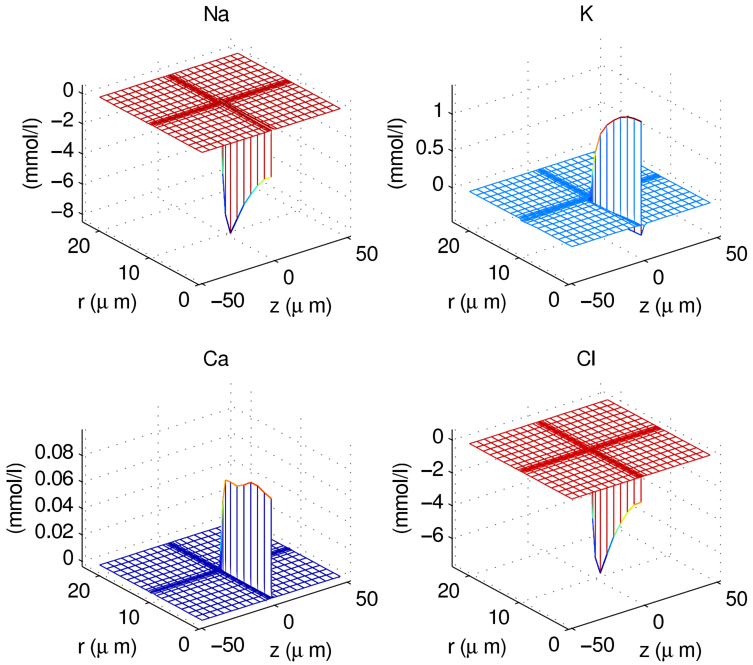


Figure 8. The cumulative *change* in ionic concentrations from the initial value, in the cardiac simulation with variable mesh width. The plot of the cumulative concentration changes of Na^+, K^+ and Cl^- at $t = 2\,\text{ms}$ shown. As with the previous figure, the radial cross section is plotted. The mesh size is $N_z \times N_r = 48 \times 32$ in this computation. Concentrations are in mmol/l and lengths in μm.

Convergence in space. As we remarked above, the gap width l_g is four orders of magnitude smaller than the cell length l_A. We therefore use a nonuniform mesh, both in the axial and radial directions.

In the axial direction, we lay a mesh whose width is of order $l_g/2^n$ when $-l_g/2 < z < l_g/2$ and of order l_A/n away from the gap where $z \sim \pm(l_A/2)$, where n is the number of steps in each direction into which the lengths of our system are divided. For meshes in between, we interpolate the two widths with an approximate geometric sequence. In the radial direction, we lay a mesh of width of order l_g near $r = l$ and of order l where $r = l \pm l$. Again, we interpolate between the extremes with an approximate geometric sequence. We give details of this construction below.

We first define a function f on $0 \le z \le (l_A + l_g)/2$:

$$f(z) = \begin{cases} 2z/l_g & \text{if } 0 \le z \le l_g/2, \\ (2/(l_g b)) \log\bigl(1 + b(z - l_g/2)\bigr) + 1 & \text{if } l_g/2 \le z \le z_\beta, \\ ((n_z - 1)/l_A)(z - l_A/2) + n_z & \text{if } z_\beta \le z \le l_A/2, \end{cases} \tag{64}$$

where n_z is an integer parameter that we specify, and b and z_β are determined so that f is continuously differentiable at $z = z_\beta$. We define the FV boundaries z_k using $f(z)$ as follows:

$$z_k = f^{-1}\Big(\frac{n_z}{N_z/2}k\Big), \quad k = 0, \ldots, N_z/2, \tag{65}$$

where $N_z/2$ is a multiple of n_z. This construction adjusts the FV width depending on whether the location is far away from the intercellular gap. For $z < 0$, we take the FV boundaries to be the reflection of the z_k above with respect to $z = 0$.

In the radial direction, we shall take the following mesh. We first define the following function g analogous to f above. For $r > l$ let

$$g(r) = \begin{cases} (2/(l_g b)) \log(1 + b(r - l)) & \text{if } l \le r \le r_\beta, \\ (n_r/2l)(r - l) + n_r & \text{if } r_\beta \le r \le 2l, \end{cases} \tag{66}$$

where n_r is an integer parameter that we specify, and b and r_β are determined so that g is continuously differentiable at $r = r_\beta$. We define the FV boundaries r_k using $g(r)$ as follows:

$$r_k = g^{-1}\Big(\frac{n_r}{N_r/2}k\Big), \quad k = 0, \ldots, N_r/2, \tag{67}$$

where $N_r/2$ is a multiple of n_r. For $r < l$, we take the points $2l - r_k$ as the FV boundaries. This construction again has the benefit of concentrating the meshes toward the membranes and near the gaps.

The coarsest level starts with 2 meshes $-l_g/2 < z < l_g/2$ and 5 meshes each for $z < -l_g/2$ and $z > l_g/2$, a total of $N_z = 12$ meshes in the axial direction. This corresponds to $n_z = 6$ in (64). In the radial direction, the coarsest level is $N_r = 8$ meshes, which corresponds to $n_r = 4$ in (66). We take

$$N_r = 32 \times 2^{n-1}, \quad N_z = 48 \times 2^{n-1}, \quad \Delta t = 0.02\,\text{ms}, \quad N_T = \frac{T_e}{\Delta t} = 200, \tag{68}$$

where $n = 1, \ldots, 4$.

Spatial convergence is assessed in exactly the same way as in the axonal case. Table 6 lists the rate of convergence for both c_i and ϕ at the three diameters with three norms, L^1, L^2 and L^∞ at $t = 4$ ms. Convergence rates at other time points were similar. We see approximate second order convergence overall, similarly to the neuronal axon calculation of Section 7.1.

Convergence in time. We vary the time step so that

$$\Delta t = 0.02 \times 2^{1-n}, \quad N_T \equiv \frac{T_e}{\Delta t} = 200 \times 2^{n-1}, \tag{69}$$

where $n = 1, \ldots, 4$. As the spatial mesh, we use $N_r = 64$, $N_z = 96$.

norm	$r_p^s[c_1, 64]$	$r_p^s[c_2, 64]$	$r_p^s[c_3, 64]$	$r_p^s[c_4, 64]$	$r_p^s[\phi, 64]$
L^1	1.90	1.91	1.94	1.94	2.00
L^2	1.95	1.87	1.89	1.97	2.00
L^∞	1.88	1.89	2.10	1.83	2.00

Table 6. Convergence rate in space (r_p^s) in the cardiac simulation. Values computed at $t = 4\,\text{ms}$, and $N_r = 64$.

norm	$r_p^t[c_1, 400]$	$r_p^t[c_2, 400]$	$r_p^t[c_3, 400]$	$r_p^s[c_4, 400]$	$r_p^s[\phi, 400]$
L^1	1.01	1.02	1.03	1.02	1.06
L^2	1.02	1.02	1.03	1.02	1.07
L^∞	1.06	1.04	1.08	1.06	1.07

Table 7. Convergence rate in time (r_p^t) in the cardiac simulation. Values computed at $t = 4\,\text{ms}$, and $N_T = 400$.

Table 7 lists the rate of convergence for both c_i and ϕ with three norms, L^1, L^2 and L^∞ at $t = 4\,\text{ms}$. We see first order convergence for all variables, similarly to the corresponding results in Section 7.1.

Convergence in space and time. We vary the time step and spatial mesh so that

$$N_z = 96 \times 2^{n-1}, \quad N_r = 64 \times 2^{n-1}, \quad \Delta t = 0.02 \times 4^{1-n}, \quad N_T \equiv \frac{T_e}{\Delta t} = 200 \times 4^{n-1},$$

where $n = 1, \dots, 3$. The observed convergence rate in Table 8 is the expected order of two overall similarly to corresponding results in Section 7.1.

norm	$r_p^{st}[c_1, 64]$	$r_p^{st}[c_2, 64]$	$r_p^{st}[c_3, 64]$	$r_p^{st}[c_4, 64]$	$r_p^{st}[\phi, 64]$
L^1	2.05	2.07	2.09	2.05	2.16
L^2	2.06	2.08	2.09	2.05	2.20
L^∞	2.16	2.11	2.22	2.16	2.20

Table 8. Convergence rate in space and time (r_p^{st}) in cardiac simulation. Values computed at $t = 4\,\text{ms}$, and $N_z = 64$, $N_T = 200$.

8. General two-dimensional geometry

In this section we consider three examples of general two-dimensional geometry. All three cases involve one or more cells in a two-dimensional square computational domain. Let the computational domain be of size l. Take the origin of the domain to be at the center of the computational domain, and take the x and y axes parallel

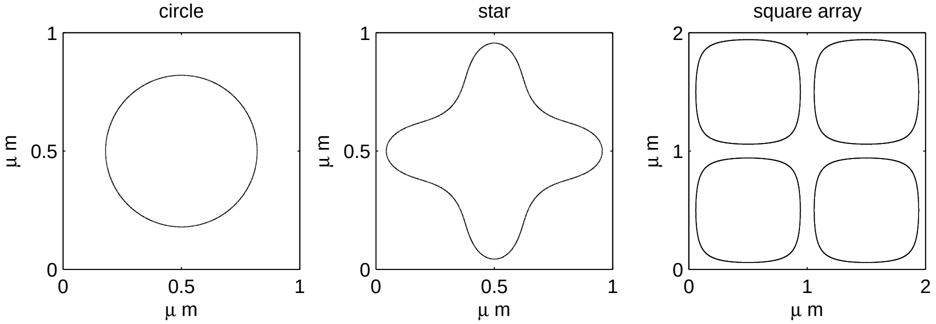


Figure 9. Shapes of cells used in computational experiments.

to the sides of the square computational domain. We consider the following three cases as regions of the intracellular domain.

$$\begin{aligned}\left(\frac{2x}{l}\right)^2+\left(\frac{2y}{l}\right)^2 &< \frac{105}{256}, \quad l=1\,\mu\text{m},\\ \exp\left(-\left(\frac{2x}{l}\right)^2-10y\right)+\exp\left(-\left(\frac{2y}{l}\right)^2-10x\right) &> \frac{1}{2}, \quad l=1\,\mu\text{m},\\ \sin^2\frac{4\pi x}{l}\sin^2\frac{4\pi y}{l} &> \frac{1}{30}, \quad l=2\,\mu\text{m}.\end{aligned}$$

The first represents a circular cell, the second a star-shaped cell and the third represents four intracellular domains in a 2 × 2 square array (Figure 9).

We use parameter values that are as close as possible to physiological parameters in the context of a two-dimensional geometry. The value of l is chosen so that it is a typical scale for microstructures in the central nervous system[18; 15; 30]. For the ionic channel model, we use the Hodgkin–Huxley kinetics. Given the geometries are two-dimensional, we cannot claim that our geometries correspond closely to those of specific physiological systems. However, even our two dimensional studies may be of physiological interest as the cross-sectional profile of systems with large longitudinal extent. For example, adjacent axons that run parallel may influence the electrical activity of one another. Such coupling has been implicated in neuropathic pain [12] and has been suggested to play a role in the corpus callosum and optic and auditory nerves [33]. The square array example above may be seen as a cross section of four axons running parallel.

At the outer boundary of the computational domain, we impose either no-flux or Dirichlet boundary conditions. In the case of Dirichlet boundary conditions, we set the c_i to be equal to their initial values, and we set ϕ equal to 0. If we can demonstrate that the scheme performs well under no-flux and Dirichlet boundary conditions, it would then seem likely that the scheme will perform well for mixed boundary conditions.

T	Absolute temperature	$273.15+37$ K
D_{Na^+}	Diffusion coefficient of Na^+	$0.266\,\mu\mathrm{m}^2/\mathrm{ms}$
D_{K^+}	Diffusion coefficient of K^+	$0.392\,\mu\mathrm{m}^2/\mathrm{ms}$
D_{Cl^-}	Diffusion coefficient of Cl^-	$0.406\,\mu\mathrm{m}^2/\mathrm{ms}$
$c^{\mathrm{int}}_{\mathrm{Na}^+}\big\|_{t=0}$	Initial intracellular concentration of Na^+	10 mmol/l
$c^{\mathrm{ext}}_{\mathrm{Na}^+}\big\|_{t=0}$	Initial extracellular concentration of Na^+	145 mmol/l
$c^{\mathrm{int}}_{\mathrm{K}^+}\big\|_{t=0}$	Initial intracellular concentration of K^+	140 mmol/l
$c^{\mathrm{ext}}_{\mathrm{K}^+}\big\|_{t=0}$	Initial extracellular concentration of K^+	5 mmol/l
$c^{\mathrm{int}}_{\mathrm{Cl}^-}\big\|_{t=0}$	Initial intracellular concentration of Cl^-	20 mmol/l
$c^{\mathrm{ext}}_{\mathrm{Cl}^-}\big\|_{t=0}$	Initial extracellular concentration of Cl^-	150 mmol/l
$\phi_{\mathrm{m}}\|_{t=0}$	Initial transmembrane potential, $\phi^{\mathrm{int}}-\phi^{\mathrm{ext}}$	-70 mV
G_{Na}	Maximal Na^+ channel conductance	600 mS/cm^2
G_{K}	Maximal K^+ channel conductance	180 mS/cm^2
G_{L}	Leak conductance (carried by K^+ ions)	1.5 mS/cm^2

Table 9. Parameter values used in the simulation for general two-dimensional geometries.

In order to observe appreciable changes in ionic concentrations over the time range of the computational study, we scaled the maximal conductances by a factor of 5 and decreased the diffusion coefficient by a factor of 5 with respect to the values of Table 1. The ionic makeup of the simulation is therefore Na^+, K^+ and Cl^-, where the concentrations and the diffusion coefficients used are summarized in Table 9. As before, the immobile charge density is initialized so that electroneutrality is strictly satisfied at $t=0$. We initialize λ_i with $\tilde{\lambda}_i$ evaluated using the initial concentrations and membrane potential.

We add the following the membrane conductances for $0\le t\le \tau_e$ to initiate an action potential for each of the three geometries.

$$G^{\mathrm{add}}_{\mathrm{Cl}^-}=G^{\mathrm{add}}_{\mathrm{Na}^+}=G^{\mathrm{add}}_{\mathrm{K}^+}=\begin{cases}200\left(\dfrac{2y}{l}\right)^2\left(1-\cos\dfrac{2\pi t}{\tau_e}\right) & \text{if } y<0,\ t<\tau_e=1\,\mathrm{ms},\\ 0 & \text{otherwise.}\end{cases}$$

We run the simulation for a total of $T_e=2\,\mathrm{ms}$. Snapshots from the simulation are given in Figures 10–12, where a no-flux boundary condition is used at the boundary of the computational domain.

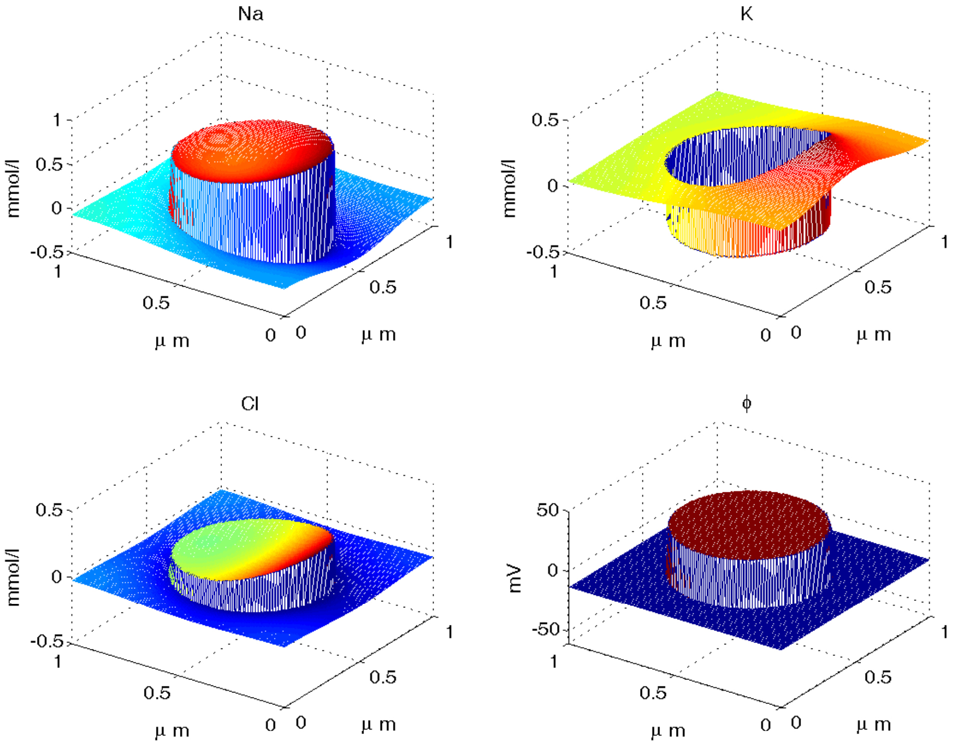


Figure 10. Circular geometry: cumulative change in ionic concentrations and electrostatic potential ϕ computed under no-flux boundary conditions at the outer boundary of the computational domain. Snapshot at $t = 0.6$ ms. Mesh size: 128×128.

8.1. ***Convergence in space.*** We lay a uniform mesh of $N_x \times N_x$ over the computational domain, where the membrane cuts through the uniform mesh as described in the above. We vary N_x in multiples of 2. We take

$$N_x = 32 \times 2^{n-1}, \qquad \Delta t = 0.02\,\text{ms}, \qquad N_T = T_e/\Delta t = 100, \tag{70}$$

where $n = 1, \dots, 5$. Convergence is measured similarly to the cylindrical cases discussed above.

We tabulate the convergence rates in Table 10. We see that it is linear to supralinear. For ϕ, the rate of convergence is not smooth as the mesh is refined (see Figure 13 on page 125). Given that we observe smooth second order convergence in the case of cylindrical geometry, we infer that this behavior arises because the Cartesian cells are randomly cut by the membrane as the mesh is refined. A direction for future study would be to improve the discretization of the equations near the membrane or adopt a better discretization of the geometry to improve the convergence profile.

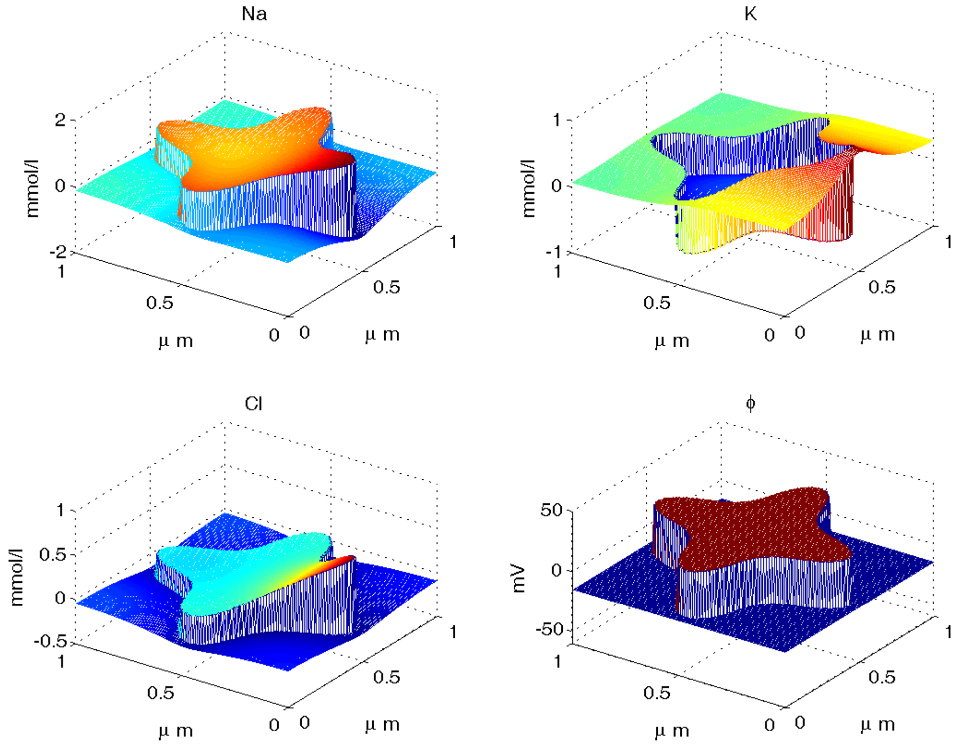


Figure 11. Star geometry: cumulative change in ionic concentrations and electrostatic potential ϕ computed under no-flux boundary conditions at the outer boundary of the computational domain. Snapshot at $t = 0.6$ ms. Mesh size: 128 × 128.

Geometry and norm		No Flux				Dirichlet			
		c_1	c_2	c_3	ϕ	c_1	c_2	c_3	ϕ
Circle	L^1	1.52	1.54	1.51	1.50	1.07	1.04	1.15	2.24
	L^2	1.49	1.52	1.51	1.50	1.02	1.01	1.14	2.25
	L^∞	0.92	1.02	1.48	1.50	0.94	1.02	1.23	2.21
Star	L^1	0.83	1.23	1.73	2.76	1.24	1.18	1.14	1.87
	L^2	1.27	1.38	1.79	2.77	1.05	1.04	1.05	1.87
	L^∞	1.16	1.12	1.51	2.47	1.14	1.12	0.97	1.86
Square Array	L^1	2.01	2.02	1.54	1.79	1.13	1.11	1.22	1.11
	L^2	1.71	1.84	1.58	1.82	1.05	1.05	1.07	1.10
	L^∞	0.80	0.78	0.76	1.61	0.96	1.01	1.01	1.10

Table 10. Convergence rate in space (r_p^s). Computed at $t = 2$ ms, and $N_x = 128$.

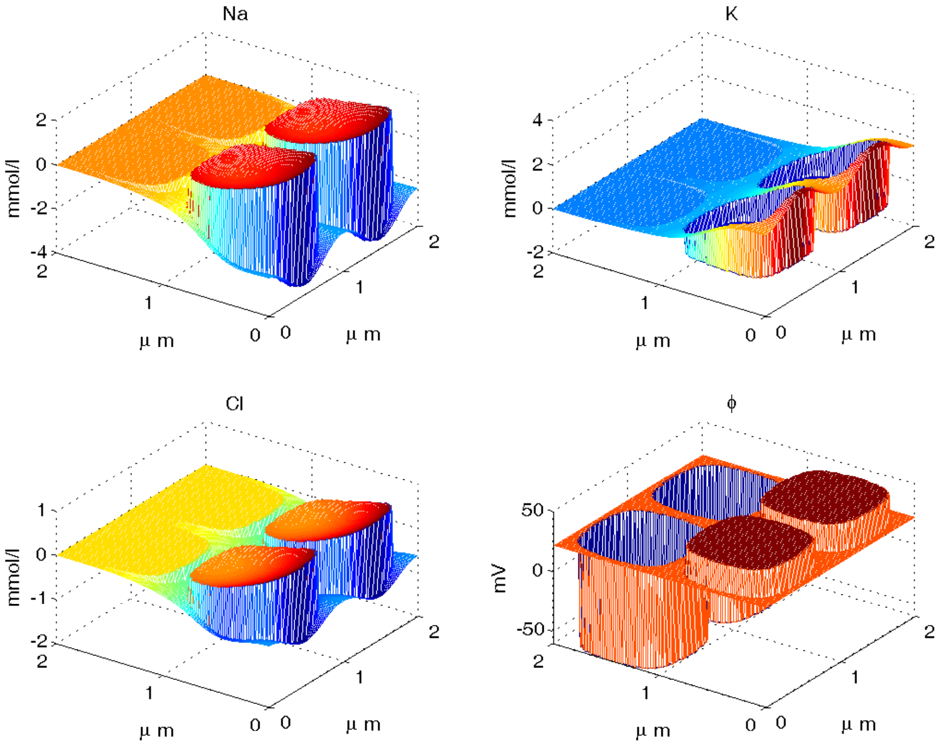


Figure 12. Square array geometry: cumulative change in ionic concentrations and electrostatic potential ϕ computed under no-flux boundary conditions at the outer boundary of the computational domain. Snapshot at $t = 0.6$ ms. Mesh size: 128 × 128.

Geometry and norm		No Flux c_1	No Flux c_2	No Flux c_3	No Flux ϕ	Dirichlet c_1	Dirichlet c_2	Dirichlet c_3	Dirichlet ϕ
Circle	L^1	1.00	1.00	0.99	1.00	1.00	1.00	1.00	1.00
	L^2	1.00	1.00	0.99	1.00	1.00	1.00	0.99	1.00
	L^∞	1.00	1.01	0.69	1.00	1.00	1.01	1.00	1.00
Star	L^1	1.00	1.00	0.99	1.00	1.00	1.00	0.99	1.00
	L^2	1.00	1.00	0.99	1.00	1.00	1.00	0.99	1.00
	L^∞	1.00	0.91	0.86	1.00	1.00	0.91	0.86	0.99
Square Array	L^1	0.99	0.99	0.98	1.00	0.99	0.99	0.98	0.99
	L^2	0.99	0.99	0.98	1.00	0.99	0.99	0.97	0.99
	L^∞	0.99	0.99	1.08	1.00	0.99	0.82	0.95	0.99

Table 11. Convergence rates in time (r_p^t). Computed at $t = 2$ ms, and $N_T = 200$.

8.2. ***Convergence in time.*** For convergence in time, we let

$$\Delta t = 0.04 \times 2^{1-n}\text{ms}, \qquad N_T = \frac{T_e}{\Delta t} = 50 \times 2^{n-1}, \tag{71}$$

where $n = 1, \ldots, 5$. For the spatial grid, we take $N_x = 64$. We give a table of the convergence rates in Table 11. We see first order convergence in all cases.

8.3. ***Convergence in space and time.*** We test convergence in space and time. As was demonstrated above, convergence in space is linear to supralinear. We thus refine the time step proportionally to the spatial step in order to study spatiotemporal convergence. We expect to see first order convergence in this case. We let

$$\Delta t = 0.04 \times 2^{1-n}\text{ms}, \qquad N_T = \frac{T_e}{\Delta t} = 50 \times 2^{n-1}, \qquad N_x = 64 \times 2^{n-1}, \tag{72}$$

where $n = 1, \ldots, 4$. We give a table of the convergence rates in Table 12. We observe approximate first order convergence in all cases as expected.

Geometry and norm		No Flux				Dirichlet			
		c_1	c_2	c_3	ϕ	c_1	c_2	c_3	ϕ
Circle	L^1	1.01	1.01	1.01	1.01	1.00	1.00	1.02	1.00
	L^2	1.01	1.01	1.01	1.01	1.01	1.01	1.01	1.00
	L^∞	1.00	1.01	1.01	1.01	1.00	1.01	1.01	1.00
Star	L^1	0.99	1.00	0.92	1.00	0.99	0.99	0.93	0.99
	L^2	0.99	0.99	0.92	1.00	0.99	0.99	0.92	0.99
	L^∞	0.98	0.99	0.93	1.00	0.98	0.99	0.93	0.99
Square Array	L^1	0.98	0.99	0.94	0.99	0.97	0.98	0.94	0.99
	L^2	0.98	0.99	0.94	1.00	0.97	0.98	0.94	0.99
	L^∞	0.99	1.01	0.94	0.99	0.96	0.97	0.96	0.99

Table 12. Convergence rates in space and time r_p^{st}. Computed at $t = 2\,\text{ms}$, and $Nx = 128$, $N_T = 100$.

9. Conclusion

We have presented a numerical method for an electrodiffusion model of cellular electrical activity, which we call the electroneutral model. The ionic concentrations c_i obey the drift diffusion equations and the electrostatic potential ϕ evolves so as to ensure electroneutrality. The boundary conditions at the membrane are expressed in terms of the capacitive current term $C_{\text{m}}((\partial\phi_{\text{m}})/(\partial t))$ as well as the ionic channel current term j_i. We have a system of partial differential equations satisfied in both

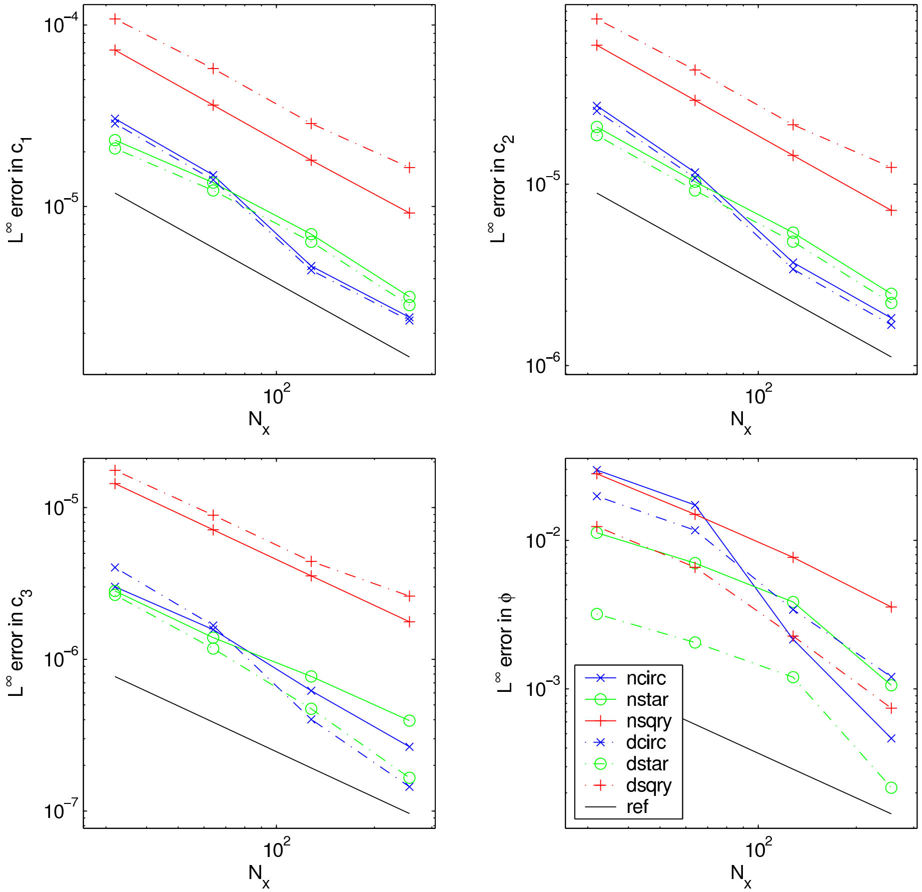


Figure 13. L^∞ error in space for c_i(mmol/l) and ϕ(mV). Error measured at 2 ms. ncirc, dcirc: circular geometry with no-flux/Dirichlet boundary conditions; nstar; dstar: star geometry; nsqry, dsqry: square array geometry; ref: reference line indicating first order convergence.

the intracellular and extracellular regions supplemented with nonlinear evolutionary interface conditions at the membrane.

We use a finite volume method in space, a natural discretization since all equations can be written in conservation form. We develop code for both cylindrical and general two dimensional membrane geometries. In the latter case, we use an embedded boundary method, in which the membrane cuts through a regular Cartesian mesh.

The model possesses two diffusive time scales, one that originates from the "diffusion" of the membrane potential and the other from the physical diffusion of ions. The membrane potential "diffusion" is fast compared to the time scale of biophysical phenomena of interest. We thus develop an implicit scheme to overcome this severe time step restriction that an explicit scheme would face as a result of this disparity of time scales. This means in particular that we must solve an elliptic interface problem where the jump in ϕ is not known a priori. The resulting nonlinear algebraic equations in c_i and ϕ are solved using an iterative scheme. We fix c_i and solve for ϕ, and fix ϕ to solve for c_i. This reduces each linear algebra task to the solution of a symmetric positive definite system. We use either a direct solver or a conjugate gradient iteration to solve these linear systems.

We examined the convergence properties of our scheme in both the cylindrical case and also in the case of the scheme for general two-dimensional geometry. In the cylindrical case, we applied the method to the Hodgkin–Huxley axon and to a model of cardiac action potential propagation. We observe close to second order accuracy in space and first order accuracy in time. For general two dimensional geometries, we test convergence with three geometries in which realistic biophysical parameters are used. We see first order accuracy in time. In space, the convergence rate is linear to supralinear, although in some cases, the convergence profile seems to be somewhat erratic. Improving both the order and the profile of spatial convergence is a direction for future research. We would also like to improve the accuracy of our time stepping scheme. We have employed an operator-splitting framework in which the gating variables defined on the membrane and the electrostatic potential/ionic concentrations defined in the bulk are marched alternately, each of which are discretized using a backward Euler type scheme. Merely replacing the backward Euler scheme with a second order L-stable method will not yield a second order scheme, since the splitting errors incurred will still be first order in time. One future direction would be to adapt splitting methods developed, for example, in [6] to develop higher order time marching schemes.

We anticipate many applications for the numerical scheme introduced in this paper. These include any situation in electrophysiology in which detailed membrane geometry and/or local changes in ionic concentrations are important. One such application was already used as a test problem in this paper. It concerns the transmission of the cardiac action potential across the narrow gap that separates the ends of adjacent myocytes. This gap is normally spanned by specialized channels known as gap junctions [1], but we study here the transmission that can occur even in the absence of these direct connections between neighboring cells [35]. A detailed study of this issue using the present model can be found in [26].

Potential applications in neuroscience include specialized synapses where geometrical relationships, localized extracellular currents, and ionic concentration changes in restricted spaces are thought to play a role [18; 34]. An example of this would be the ribbon synapse of the retina, in which horizontal cells mediate the interaction between photoreceptors and bipolar cells in ways that are only partly understood [14]. Yet another potential arena of application concerns intracellular electrophysiology, that is, the role of electrodiffusion of ions in the function of such intracellular organelles as the sarcoplasmic reticulum or the mitochondrion.

Most of the applications discussed above will probably require for their full realization a three-dimensional generalization of the code for general two-dimensional geometry that we have developed for the purpose of testing the basic methodology in the present paper. Local mesh refinement will most likely be needed to accommodate the different spatial scales that will interact in any particular application. The principles on which our two-dimensional method are based extend readily to the three-dimensional case even with local mesh refinement. There are, however, significant implementation difficulties that must be overcome, most notably in the representation of the complicated three-dimensional membrane geometry and its interaction with a locally refined mesh. Parallel implementation and efficient solvers will also be needed in the three-dimensional case. This substantial research effort will be rewarded by the ability to make detailed simulations of electrically active cells at a level that takes into account their intricate and beautiful microscopic anatomy.

Appendix

A.1. ***Derivation of*** $\tilde{\boldsymbol{\lambda}}_i^{(k)}$**.** We give a derivation of the expression for $\tilde{\lambda}_i^{(k)}$ in (8). What is presented here is an adaptation of a calculation contained in [27]. For a derivation using matched asymptotics we refer to [25] and [24].

We take a closer look at what is happening in the space charge layer, the thin layers of electric charge accumulation that form on both sides of the membrane. We now derive the ionic composition of the space charge layer when it is in equilibrium with the bulk solution in the immediate vicinity of the space charge layer.

Our starting point is the following Poisson–Nernst–Planck system satisfied in both the intracellular and extracellular regions.

$$\frac{\partial c_i}{\partial t} = -\nabla \cdot \boldsymbol{f}_i, \tag{73}$$

$$\boldsymbol{f}_i = -D_i\left(\nabla c_i + \frac{q z_i c_i}{k_B T}\nabla\phi\right), \tag{74}$$

$$-\epsilon\,\Delta\phi = \rho_0 + \sum_{i=1}^{N} q z_i c_i. \tag{75}$$

All quantities except for the dielectric constant ϵ have been introduced in (1)–(3). Instead of the electroneutrality condition (3), we have the Poisson equation (75). Taking c_0 and L_0 to be the representative ionic concentration and spatial scales respectively, the Poisson equation can be nondimensionalized as

$$-\Big(\frac{r_d}{L_0}\Big)^2\tilde{\Delta}\tilde{\phi} = \tilde{\rho_0} + \sum_{i=1}^{N} z_i\tilde{c}_i, \tag{76}$$

where $\tilde{\cdot}$ denote the respective nondimensionalized quantities and operators. In (76) r_d is the Debye length given by

$$r_d = \sqrt{\frac{\epsilon k_B T}{q^2 c_0}}. \tag{77}$$

Given that $r_d \approx 1$ nm and L_0 is on the order or μm to cm in biophysical systems, $(r_d/L_0)^2$ is a very small quantity. We may thus safely disregard the left hand side of (76) provided we are sufficiently far away from the membrane. This amounts to taking the right hand side of (75) to be equal to 0. This leads to the electroneutrality condition (3). However close to the membrane we have a boundary layer of thickness $\mathbb{O}(r_d)$, within which the ionic concentrations deviate from electroneutrality. In this space charge layer, we must deal with the Poisson–Nernst–Planck system (73), (74) and (75).

We make some assumptions in our analysis of the space charge layer. We suppose that the quantities within the space charge layer experience fast spatial variation in the direction normal to the membrane but slow spatial variation in the direction parallel to the membrane. Under this "boundary layer" assumption, all quantities may be treated as functions only of the distance from the membrane. This also implies that the ionic fluxes must be equal to 0 to leading order within the space charge layer. It is possible to formalize this argument within the traditional framework of matched asymptotics as presented in [25] and [24]. We also make the assumption that the deviation of the electrostatic potential and hence the ionic concentration from its bulk values is small. This assumption is justified because the membrane capacitance is "small" in biophysical systems. For a further elaboration of this point we refer the reader to [27].

Let x denote the distance coordinate normal to the membrane surface. Then, according to the assumptions just stated, (73)–(75) become

$$0 = -D_i\Big(\frac{\partial c_i}{\partial x} + \frac{q z_i c_i}{k_B T}\frac{\partial \phi}{\partial x}\Big), \tag{78}$$

$$-\frac{\partial^2\phi}{\partial x^2} = \frac{1}{\epsilon}\Big(\rho_0 + \sum_{i=1}^{N} q z_i c_i\Big), \tag{79}$$

which hold on $0 < x < \infty$ with $c_i(\infty)$ and $\phi(\infty)$ given. Here, $x = 0$ is the intracellular or extracellular face of the membrane, and $x = \infty$ corresponds to the bulk solution where c_i and ϕ values in the space charge layer are to be matched with the bulk values. For now we shall assume that $c_i(\infty)$ and $\phi(\infty)$ are constant in time. Assuming that the background fixed charge density ρ_0 varies on the scale of the cellular size L_0, its variation within the thickness of the space charge layer is negligible, of order $\mathcal{O}(r_d/L_0)$. Thus, we will treat ρ_0 as being constant within the space charge layer. It is important to note that these values at $x = \infty$ satisfy electroneutrality, that is,

$$\rho_0 + \sum_{i=1}^{N} qz_i c_i(\infty) = 0. \tag{80}$$

Equation (78) can be integrated easily to obtain,

$$c_i(x) = c_i(\infty) \exp\Big(-\frac{qz_i}{k_B T}(\phi(x) - \phi(\infty))\Big). \tag{81}$$

We substitute this into (79) to find,

$$-\frac{\partial^2 \phi}{\partial x^2} = \frac{1}{\epsilon}\Bigg(\rho_0 + \sum_{i=1}^{N} qz_i c_i(\infty) \exp\Big(-\frac{qz_i}{k_B T}(\phi(x) - \phi(\infty))\Big)\Bigg). \tag{82}$$

We now assume, as we stated earlier, that the deviation of the electrostatic potential within the space charge layer from the bulk value is small.

$$\left|\frac{qz_i}{k_B T}(\phi(x) - \phi(\infty))\right| \ll 1. \tag{83}$$

Then, taking into account the electroneutrality condition at $x = \infty$ (80), we obtain

$$c_i(x) = c_i(\infty)\Big(1 - \frac{qz_i}{k_B T}(\phi(x) - \phi(\infty))\Big) \tag{84}$$

and

$$\frac{\partial^2}{\partial x^2}(\phi(x) - \phi(\infty)) = \gamma^2(\phi(x) - \phi(\infty)),$$

where

$$\gamma^2 = \sum_{i=1}^{N} \frac{(qz_i)^2 c_i(\infty)}{\epsilon k_B T}. \tag{85}$$

Letting γ be the positive square root of γ^2 ($1/\gamma$ is nothing other than the Debye length), we find the unique bounded solution

$$\phi(x) - \phi(\infty) = (\phi(0) - \phi(\infty)) \exp(-\gamma x), \tag{86}$$

and hence according to (84),

$$c_i(x) - c_i(\infty) = -c_i(\infty)\frac{qz_i}{k_B T}(\phi(0) - \phi(\infty)) \exp(-\gamma x). \tag{87}$$

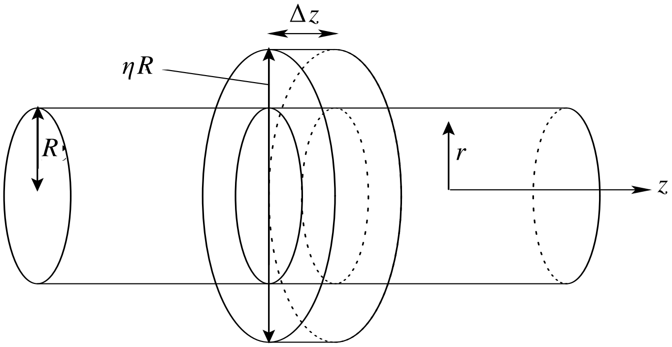


Figure 14. Derivation of the cable model. A portion of a cylindrical cell is shown. As explained in the main text, the divergence theorem is applied to the intracellular and extracellular slabs of thickness Δz, shown above.

Using this equation, we may compute σ_i as

$$\sigma_i = \int_0^\infty q z_i (c_i(x) - c_i(\infty)) dx = -c_i(\infty) \frac{(q z_i)^2}{k_B T \gamma} (\phi(0) - \phi(\infty)). \tag{88}$$

Using the above and noting that $\sum_{i=1}^N \sigma_i = \sigma$, we immediately obtain

$$\sigma_i = \frac{z_i^2 c_i(\infty)}{\sum_{i'=1}^N z_{i'}^2 c_{i'}(\infty)} \sigma. \tag{89}$$

The coefficient in front of σ corresponds to the expression for $\tilde{\lambda}_i$ in (8).

If the bulk concentrations are not changing in time, the fractional contribution $\sigma_i/\sigma = \lambda_i$ of (8) will be equal to $\tilde{\lambda}_i$. If the bulk concentration changes slowly with time, we expect the ionic concentration profile within the space charge layer to closely follow the corresponding equilibrium profile calculated above, on the diffusive time scale within the space charge layer. We thus let λ_i relax to $\tilde{\lambda}_i$ with the time constant $\tau = r_d^2/D_0$, where D_0 is the representative magnitude of ionic diffusion coefficient.

A.2. ***Derivation of the cable model.*** We give a short derivation of the cable model from (15) and (16). Our derivation here can be formalized using thin domain asymptotics. See [25] and [24] for details.

Suppose the cell is cylindrical in shape (Figure 14). Let z be the axial coordinate and r be the radial coordinate. The membrane is located at $r = R$. The intracellular space corresponds to $r < R$ and the extracellular space to $R < r < \eta R$ where $\eta > 1$ is some constant. Now, consider the cross sectional slab between $z = z_0$ and $z = z_0 + \Delta z$. Let us compute the integral of $\Delta \phi$ over the region $z_0 < z < z_0 + \Delta z$,

$r < R$. We get

$$\int_{z_0<z<z_0+\Delta z, r<R} \Delta\phi dV$$
$$= \int_{z=z_0+\Delta z, r<R} \frac{\partial\phi}{\partial z} dA - \int_{z=z_0, r<R} \frac{\partial\phi}{\partial z} dA + \int_{z_0<z<z_0+\Delta z, r=R} \frac{\partial\phi}{\partial r} dA, \quad (90)$$

where we used the divergence theorem. The symbols dV and dA denote volume and surface integration respectively. Now, note by (17) that $\Delta\phi = 0$. Therefore, the left hand side of (90) is 0, and we have

$$\int_{z=z_0+\Delta z, r<R} \frac{\partial\phi}{\partial z} dA - \int_{z=z_0, r<R} \frac{\partial\phi}{\partial z} dA = -\int_{z_0<z<z_0+\Delta z, r=R} \frac{\partial\phi}{\partial r} dA. \quad (91)$$

Dividing by Δz and taking the limit as Δz goes to 0, we obtain the following relationship.

$$\frac{\partial}{\partial z} \int_{z=z_0} \frac{\partial\phi}{\partial z} dA = -\int_{z=z_0, r=R} \frac{\partial\phi}{\partial r} d\ell, \quad (92)$$

where $d\ell$ denotes a line integral. Now, we make the assumption that the cylindrical diameter is small so that the electrostatic potential ϕ varies very little over the diameter of the cylinder. We thus take the approximation that $\phi = \phi^{\text{int}}(z)$ does not depend on the radial direction in $r < R$. Under this approximation, the above becomes

$$a^{\text{int}} \pi R^2 \frac{\partial^2 \phi^{\text{int}}}{\partial z^2} = \int_{r=R} \left(C_{\text{m}} \frac{\partial \phi_m}{\partial t} + I \right) d\ell, \quad (93)$$

where we used (18). Note that the above is valid for any value of z_0, and thus, we have omitted reference to z_0. ϕ_m is the membrane potential, the difference in ϕ across the membrane. a^{int} is the value of a (which appears in (16) as $a^{(k)}$ and $a^{(l)}$) in the intracellular space.

A similar calculation, applied to the extracellular region $z_0 < z < z_0 + \Delta z$, $R < r < R^{\text{ext}}$, yields

$$a^{\text{ext}} \pi ((\eta R)^2 - R^2) \frac{\partial^2 \phi^{\text{ext}}}{\partial z^2} = -\int_{r=R} \left(C_{\text{m}} \frac{\partial \phi_m}{\partial t} + I \right) d\ell, \quad (94)$$

where we have assumed that ϕ in the extracellular region is again, a function only of z and does not depend on the radial direction r. Note that the membrane potential ϕ_m can now be expressed as $\phi_m = \phi^{\text{int}} - \phi^{\text{ext}}$, and is thus a function only of z, and does not depend on the angular coordinate. Rearranging (93) and (94), we may write an equation solely in terms of ϕ_m:

$$C_{\text{m}} \frac{\partial \phi_m}{\partial t} + \bar{I} = \frac{a^{\text{eff}} R}{2} \frac{\partial^2 \phi_m}{\partial z^2}, \quad a^{\text{eff}} = ((a^{\text{int}})^{-1} + (a^{\text{ext}}(\eta^2 - 1))^{-1})^{-1}. \quad (95)$$

This is nothing other than (20). If we let $a = a^{\text{int}} = a^{\text{ext}}$ and consider the case in which η is very large, we may replace a^{eff} by a, as used in (19).

Acknowledgments

The authors thank Joseph W. Jerome for helpful discussion. Mori was supported by the Henry McCracken and Dissertation Fellowships from New York University, where most of this work was done. Mori also acknowledges support from a MITACS (Mathematics of Information Technology and Complex Systems) team grant (Leah Keshet, PI) and from an NSERC (Natural Sciences and Engineering Research Council) discovery grant (also with Leah Keshet).

References

[1] D. Aidley, *The physiology of excitable cells*, 4th ed., Cambridge Univ. Press, NY, 1998.

[2] S. Balay, K. Buschelman, W. Gropp, D. Kaushik, M. Knepley, L. McInnes, B. Smith, and H. Zhang, *PETSc Web page*, 2001.

[3] V. Barcilon, J. Cole, and R. Eisenberg, *A singular perturbation analysis of induced electric fields in nerve cells*, SIAM Journal on Applied Mathematics (1971), 339–354. Zbl 0231.92016

[4] O. Bernus, R. Wilders, C. Zemlin, H. Verscelde, and A. Panfilov, *A computationally efficient electrophysiological model of human ventricular cells*, Am. J. Physiol. Heart Circ. Physiol. **282** (2002), 2296–2308.

[5] J. Bockris and A. Reddy, *Modern electrochemistry. Vol. 1, Ionics*, Plenum, 1998.

[6] A. Bourlioux, A. T. Layton, and M. L. Minion, *High-order multi-implicit spectral deferred correction methods for problems of reactive flow*, J. Comput. Phys. **189** (2003), no. 2, 651–675. MR 2004f:76084 Zbl 1061.76053

[7] D. Calhoun and R. J. LeVeque, *A Cartesian grid finite-volume method for the advection-diffusion equation in irregular geometries*, J. Comput. Phys. **157** (2000), no. 1, 143–180. MR 2000k:65166 Zbl 0952.65075

[8] R. Eisenberg and E. Johnson, *Three-dimensional electrical field problems in physiology*, Prog. Biophys. Mol. Biol **20** (1970), no. 1, 1–65.

[9] D. Gutstein, G. Morley, H. Tamaddon, D. Vaidya, M. Schneider, J. Chen, K. Chien, H. Stuhlmann, and G. Fishman, *Conduction slowing and sudden arrhythmic death in mice with cardiac-restricted inactivation of connexin 43*, Circulation Research **88** (2001), no. 8, 333–339.

[10] B. Hille, *Ion channels of excitable membranes*, 3rd ed., Sinauer Associates, 2001.

[11] A. Hodgkin and A. Huxley, *A quantitative description of the membrane current and its application to conduction and excitation in nerve*, Journal of Physiology **117** (1952), 500–544.

[12] J. Jeffreys, *Nonsynaptic modulation of neuronal activity in the brain: electric currents and extracellular ions*, Physiological Reviews **75** (1995), 689–723.

[13] H. Johansen and P. Colella, *A Cartesian grid embedded boundary method for Poisson's equation on irregular domains*, J. Comput. Phys. **147** (1998), no. 1, 60–85. MR 99m:65231 Zbl 0923.65079

[14] M. Kamermans, I. Fahrenfort, K. Schultz, U. Janssen-Bienhold, T. Sjoerdsma, and R. Weiler, *Hemichannel-mediated inhibition in the outer retina*, Science **292** (2001), no. 5519, 1178–1180.

[15] E. Kandel, J. Schwartz, and T. Jessel, *Principles of neural science*, 4th ed., McGraw-Hill/Appleton & Lange, NY, 2000.

[16] J. Keener and J. Sneyd, *Mathematical physiology*, Interdisciplinary Applied Mathematics, no. 8, Springer, New York, 1998. MR 2000c:92010 Zbl 0913.92009

[17] D. A. Knoll and D. E. Keyes, *Jacobian-free Newton–Krylov methods: a survey of approaches and applications*, J. Comput. Phys. **193** (2004), no. 2, 357–397. MR 2004j:65066 Zbl 1036.65045

[18] C. Koch, *Biophysics of computation*, Oxford Univ. Press, New York, 1999.

[19] J. Kucera, S. Rohr, and Y. Rudy, *Localization of sodium channels in intercalated disks modulates cardiac conduction*, Circulation Research **91** (2002), no. 12, 1176–82.

[20] L. Lee and R. J. Leveque, *An immersed interface method for incompressible Navier–Stokes equations*, SIAM J. Sci. Comput. **25** (2003), no. 3, 832–856. MR 2005a:65086

[21] R. J. LeVeque, *Finite volume methods for hyperbolic problems*, Cambridge Texts in App. Math., Cambridge University Press, Cambridge, 2002. MR 2003h:65001 Zbl 1010.65040

[22] A. A. Mayo and C. S. Peskin, *An implicit numerical method for fluid dynamics problems with immersed elastic boundaries*, Fluid dynamics in biology, Contemp. Math., no. 141, Amer. Math. Soc., Providence, RI, 1993, pp. 261–277. MR 93k:76132 Zbl 0787.76055

[23] P. McCorquodale, P. Colella, and H. Johansen, *A Cartesian grid embedded boundary method for the heat equation on irregular domains*, J. Comput. Phys. **173** (2001), no. 2, 620–635. MR 2002h:80009 Zbl 0991.65099

[24] Y. Mori, *From three-dimensional electrophysiology to the cable model: an asymptotic study*, 2009, preprint. arXiv 0901.3914

[25] Y. Mori, *A three-dimensional model of cellular electrical activity*, Ph.D. thesis, New York Univ., 2006. Zbl 1129.92038

[26] Y. Mori, G. I. Fishman, and C. S. Peskin, *Ephaptic conduction in a cardiac strand model with 3d electrodiffusion*, Proceedings of the National Academy of Sciences (2008), to appear.

[27] Y. Mori, J. W. Jerome, and C. S. Peskin, *A three-dimensional model of cellular electrical activity*, Bull. Inst. Math. Acad. Sin. (N.S.) **2** (2007), no. 2, 367–390. MR 2008f:92022 Zbl 1129.92038

[28] J. Neu and W. Krassowska, *Homogenization of syncytial tissues*, Critical reviews in biomedical engineering **21** (1993), no. 2, 137–199.

[29] N. Qian and T. Sejnowski, *An electro-diffusion model for computing membrane potentials and ionic concentrations in branching dendrites, spines and axons*, Biol. Cybern. **62** (1989), 1–15.

[30] ———, *When is an inhibitory synapse effective?*, Proc. Natl. Acad. Sci. USA **87** (1990), 8145–8149.

[31] S. Rohr, *Role of gap junctions in the spread of the cardiac action potential*, Cardiovascular Research **62** (2004), 309–322.

[32] I. Rubinstein, *Electro-diffusion of ions*, SIAM Studies in Applied Mathematics, no. 11, Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 1990. MR 91m:78019

[33] A. Scott, *Neuroscience: A mathematical primer*, Springer, New York, 2002. MR 2003i:92001 Zbl 1018.92003

[34] G. M. Shepherd (ed.), *Synaptic organization of the brain*, Oxford University Press, 1997.

[35] N. Sperelakis, *An electric field mechanism for transmission of excitation between myocardial cells*, Circulation Research **91** (2002), 985–987.

[36] H. A. van der Vorst, *Iterative Krylov methods for large linear systems*, Cambridge Monographs on Applied and Computational Mathematics, no. 13, Cambridge University Press, Cambridge, 2003. MR 2005k:65075 Zbl 1023.65027

[37] ______, *Iterative Krylov methods for large linear systems*, Cambridge Monographs on Applied and Computational Mathematics, no. 13, Cambridge University Press, Cambridge, 2003. MR 2005k:65075 Zbl 1023.65027

Received June 20, 2007. Revised June 22, 2009.

YOICHIRO MORI: ymori@umn.edu
School of Mathematics, University of Minnesota, 206 Church St. SE, Minneapolis, MN 55455-0487, United States
http://www.math.umn.edu/~ymori

CHARLES S. PESKIN: peskin@cims.nyu.edu
Courant Institute of Mathematical Sciences, New York University, 251 Mercer St.,
New York, NY 10012-1110, United States

COMM. APP. MATH. AND COMP. SCI.
Vol. 4, No. 1, 2009

A HIGHER-ORDER GODUNOV METHOD FOR RADIATION HYDRODYNAMICS: RADIATION SUBSYSTEM

MICHAEL DAVID SEKORA AND JAMES M. STONE

A higher-order Godunov method for the radiation subsystem of radiation hydrodynamics is presented. A key ingredient of the method is the direct coupling of stiff source term effects to the hyperbolic structure of the system of conservation laws; it is composed of a predictor step that is based on Duhamel's principle and a corrector step that is based on Picard iteration. The method is second-order accurate in both time and space, unsplit, asymptotically preserving, and uniformly well behaved from the photon free streaming (hyperbolic) limit through the weak equilibrium diffusion (parabolic) limit and to the strong equilibrium diffusion (hyperbolic) limit. Numerical tests demonstrate second-order convergence across various parameter regimes.

1. Introduction

Radiation hydrodynamics is a fluid description of matter (plasma) that absorbs and emits electromagnetic radiation and in so doing modifies dynamical behavior. The coupling between matter and radiation is significant in many phenomena related to astrophysics and plasma physics, where radiation comprises a major fraction of the internal energy and momentum and provides the dominant transport mechanism. Radiation hydrodynamics governs the physics of radiation-driven outflows, supernovae, accretion disks, and inertial confinement fusion [Castor 2004; Mihalas and Mihalas 1984]. Such physics is described mathematically by a nonlinear system of conservation laws that is obtained by taking moments of the Boltzmann and photon transport equations. A key difficulty is choosing the frame of reference in which to take the moments of the photon transport equation. In the comoving and mixed frame approaches, one captures the matter/radiation coupling by adding relativistic source terms correct to $\mathbb{O}(u/c)$ to the right side of the conservation laws, where u is the material flow speed and c is the speed of light. These source terms are stiff because of the variation in time/length scales associated with such problems [Mihalas and Klein 1982]. This stiffness causes

MSC2000: 35B40, 35L65, 35M10, 76M12.

Keywords: Godunov methods, radiation hydrodynamics, asymptotic preserving methods, hyperbolic conservation laws, stiff source terms, stiff relaxation.

numerical difficulties and makes conventional methods such as operator splitting and method of lines break down [LeVeque 1992; 2002].

Previous research in numerically solving radiation hydrodynamical problems was carried out by Caster [1972], Pomraning [1973], Mihalas and Klein [1982], and Mihalas and Mihalas [1984]. There are a variety of algorithms for radiation hydrodynamics. One of the simplest approaches was developed by Stone et al. [1992] and implemented in the ZEUS code, which was based on operator splitting and Crank–Nicholson finite differencing. Since then, higher-order Godunov methods have emerged as a valuable technique for solving hyperbolic conservation laws (for example, hydrodynamics), particularly when shock capturing and adaptive mesh refinement is important [Stone et al. 2008]. However, developing upwind differencing methods for radiation hydrodynamics is a difficult mathematical and computational task. In many cases, Godunov methods for radiation hydrodynamics either:

(i) neglect the heterogeneity of weak/strong coupling and solve the system of equations in an extreme limit [Dai and Woodward 1998; 2000];

(ii) are based on a manufactured limit and solve a new system of equations that attempts to model the full system [Jin and Levermore 1996; Buet and Despres 2006]; or

(iii) use a variation on flux limited diffusion [Levermore and Pomraning 1981; Gonzalez et al. 2007].

All of these approaches fail to treat the full generality of the problem. For example, Balsara [1999] proposed a Riemann solver for the full system of equations. However, as pointed out by Lowrie and Morel [2001], Balsara's method failed to maintain coupling between radiation and matter. Moreover, Lowrie and Morel were critical of the likelihood of developing a Godunov method for full radiation hydrodynamics.

In radiation hydrodynamics, there are three important dynamical scales and each scale is associated with either the material flow (speed of sound), radiation flow (speed of light), or source terms. When the matter-radiation coupling is strong, the source terms define the fastest scale. However, when the matter-radiation coupling is weak, the source terms define the slowest scale. Given such variation, one aims for a scheme that treats the stiff source terms implicitly. Following [Miniati and Colella 2007], this paper presents a method that is a higher-order modified Godunov scheme that directly couples stiff source term effects to the hyperbolic structure of the system of conservation laws; it is composed of a predictor step that is based on Duhamel's principle and a corrector step that is based on Picard iteration. The method is explicit on the fastest hyperbolic scale (radiation flow) but is unsplit and fully couples matter and radiation with no approximation made to the full system of equations for radiation hydrodynamics.

A challenge for the modified Godunov method is its use of explicit time differencing when there is a large range in the time scales associated with the problem, $c/a_\infty \gg 1$, where a_∞ is the reference material sound speed. One could have built a fully implicit method that advanced time according to the material flow scale, but a fully implicit approach was not pursued because such methods often have difficulties associated with conditioning, are expensive because of matrix manipulation and inversion, and are usually built into central difference schemes rather than higher-order Godunov methods. An explicit method may even out perform an implicit method if one considers applications that have flows where $c/a_\infty \lesssim 10$. A modified Godunov method that is explicit on the fastest hyperbolic scale (radiation flow) as well as a hybrid method that incorporates a backward Euler upwinding scheme for the radiation components and the modified Godunov scheme for the material components are under construction for full radiation hydrodynamics. A goal of future research is to directly compare these two methods in various limits for different values of c/a_∞.

2. Radiation hydrodynamics

The full system of equations for radiation hydrodynamics in the Eulerian frame that is correct to $\mathbb{O}(1/\mathbb{C})$ is

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\boldsymbol{m}) = 0, \tag{1}$$

$$\frac{\partial \boldsymbol{m}}{\partial t} + \nabla \cdot \Big(\frac{\boldsymbol{m} \otimes \boldsymbol{m}}{\rho}\Big) + \nabla p = -\mathbb{P}\Big[-\sigma_t \Big(\boldsymbol{F}_r - \frac{\boldsymbol{u} E_r + \boldsymbol{u} \cdot \mathsf{P}_r}{\mathbb{C}}\Big) + \sigma_a \frac{\boldsymbol{u}}{\mathbb{C}}(T^4 - E_r)\Big], \tag{2}$$

$$\frac{\partial E}{\partial t} + \nabla \cdot \Big((E+p)\frac{\boldsymbol{m}}{\rho}\Big) = -\mathbb{P}\mathbb{C}\Big[\sigma_a (T^4 - E_r) + (\sigma_a - \sigma_s)\frac{\boldsymbol{u}}{\mathbb{C}} \cdot \Big(\boldsymbol{F}_r - \frac{\boldsymbol{u} E_r + \boldsymbol{u} \cdot \mathsf{P}_r}{\mathbb{C}}\Big)\Big], \tag{3}$$

$$\frac{\partial E_r}{\partial t} + \mathbb{C}\nabla \cdot \boldsymbol{F}_r = \mathbb{C}\Big[\sigma_a (T^4 - E_r) + (\sigma_a - \sigma_s)\frac{\boldsymbol{u}}{\mathbb{C}} \cdot \Big(\boldsymbol{F}_r - \frac{\boldsymbol{u} E_r + \boldsymbol{u} \cdot \mathsf{P}_r}{\mathbb{C}}\Big)\Big], \tag{4}$$

$$\frac{\partial \boldsymbol{F}_r}{\partial t} + \mathbb{C}\nabla \cdot \mathsf{P}_r = \mathbb{C}\Big[-\sigma_t \Big(\boldsymbol{F}_r - \frac{\boldsymbol{u} E_r + \boldsymbol{u} \cdot \mathsf{P}_r}{\mathbb{C}}\Big) + \sigma_a \frac{\boldsymbol{u}}{\mathbb{C}}(T^4 - E_r)\Big], \tag{5}$$

$$\mathsf{P}_r = \mathsf{f} E_r \text{ (closure relation).} \tag{6}$$

For the material quantities, ρ is density, $\boldsymbol{m}$ is momentum, p is pressure, E is total energy density, and T is temperature. For the radiative quantities, E_r is energy density, $\boldsymbol{F}_r$ is flux, P_r is pressure, and f is the variable tensor Eddington factor.

In the source terms, σ_a is the absorption cross section, σ_s is the scattering cross section, and $\sigma_t = \sigma_a + \sigma_s$ is the total cross section.

Following [Lowrie et al. 1999; Lowrie and Morel 2001], the system of equations above has been nondimensionalized with respect to the material flow scale so that one can compare hydrodynamical and radiative effects as well as identify terms that are $\mathbb{O}(u/c)$. This scaling gives two important parameters:

$$\mathbb{C} = c/a_\infty, \quad \mathbb{P} = \frac{a_r T_\infty^4}{\rho_\infty a_\infty^2}.$$

$\mathbb{C}$ measures relativistic effects, while $\mathbb{P}$ measures how radiation affects material dynamics and is proportional to the equilibrium radiation pressure over material pressure. $a_r = (8\pi^5 k^4)/(15c^3h^3)$ is a radiation constant, T_∞ is the reference material temperature, and ρ_∞ is the reference material density.

For this system of equations, one has assumed that scattering is isotropic and coherent in the comoving frame, emission is defined by local thermodynamic equilibrium (LTE), and that spectral averages for the cross-sections can be employed (gray approximation). The coupling source terms are given by the modified Mihalas–Klein description [Lowrie et al. 1999; Lowrie and Morel 2001], which is more general and more accurate than the original Mihalas–Klein [1982] source terms because it maintains an important $\mathbb{O}(1/\mathbb{C}^2)$ term that ensures the correct equilibrium state and relaxation rate to equilibrium.

Before investigating full radiation hydrodynamics, it is useful to examine the radiation subsystem, which is a simpler system that minimizes complexity while maintaining the rich hyperbolic-parabolic behavior associated with the stiff source term conservation laws. This simpler system allows one to develop a reliable and robust numerical method. Consider Equations (4) and (5) for radiation hydrodynamics in one spatial dimension not affected by transverse flow. If one only considers radiative effects and holds the material flow stationary such that $u \to 0$, then the conservative variables, fluxes, and source terms for the radiation subsystem are given by

$$\frac{\partial E_r}{\partial t} + \mathbb{C}\frac{\partial F_r}{\partial x} = \mathbb{C}\sigma_a(T^4 - E_r), \qquad \frac{\partial F_r}{\partial t} + \mathbb{C}f\frac{\partial E_r}{\partial x} = -\mathbb{C}\sigma_t F_r. \tag{7}$$

Motivated by the asymptotic analysis of Lowrie et al. [1999] for full radiation hydrodynamics, one investigates the limiting behavior for this simpler system of equations. For nonrelativistic flows $1/\mathbb{C} = \mathbb{O}(\epsilon)$, where $\epsilon \ll 1$. Assume that there is a moderate amount of radiation in the flow such that $\mathbb{P} = \mathbb{O}(1)$. Furthermore, assume that scattering effects are small such that $\sigma_s/\sigma_t = \mathbb{O}(\epsilon)$. Lastly, assume that the optical depth can be represented as $\mathscr{L} = \ell_{\text{mat}}/\lambda_t = \ell_{\text{mat}}\,\sigma_t$, where λ_t is the total mean free path of the photos and $\ell_{\mathrm{mat}} = \mathbb{O}(1)$ is the material flow length scale [Lowrie et al. 1999].

***Free streaming limit:* $\boldsymbol{\sigma_a, \sigma_t \sim \mathbb{O}(\epsilon)}$.** In this regime, the right side of (7) is negligible, so that the system is strictly hyperbolic; $f \to 1$ and the Jacobian of the quasilinear conservation law has eigenvalues $\pm\mathbb{C}$:

$$\frac{\partial E_r}{\partial t} + \mathbb{C}\frac{\partial F_r}{\partial x} = 0, \qquad \frac{\partial F_r}{\partial t} + \mathbb{C}\frac{\partial E_r}{\partial x} = 0, \tag{8}$$

***Weak equilibrium diffusion limit:* $\boldsymbol{\sigma_a, \sigma_t \sim \mathbb{O}(1)}$.** One obtains this limit by plugging in $\sigma_a, \sigma_t \sim \mathbb{O}(1)$, matching terms of like order, and combining the resulting equations. From the definition of the equilibrium state, $E_r = T^4$ and $F_r = -(1/\sigma_t)\partial P_r/\partial x$. Therefore, the system is parabolic and resembles a diffusion equation, where $f \to 1/3$:

$$\frac{\partial E_r}{\partial t} = \frac{\mathbb{C}}{3\sigma_t}\frac{\partial^2 E_r}{\partial x^2}, \qquad F_r = -\frac{1}{3\sigma_t}\frac{\partial E_r}{\partial x}. \tag{9}$$

***Strong equilibrium diffusion limit:* $\boldsymbol{\sigma_a, \sigma_t \sim \mathbb{O}(1/\epsilon)}$.** One obtains this limit by plugging in $\sigma_a, \sigma_t \sim \mathbb{O}(1/\epsilon)$ and following the steps outlined for the weak equilibrium diffusion limit. One can consider the system to be hyperbolic, where $f \to 1/3$ and the Jacobian of the quasilinear conservation law has eigenvalues $\pm\epsilon$:

$$\frac{\partial E_r}{\partial t} = 0, \qquad F_r = 0. \tag{10}$$

Lowrie et al. [1999] investigated an additional limit for full radiation hydrodynamics, the isothermal regime. This limit has some dynamical properties in common with the weak equilibrium diffusion limit, but its defining characteristic is that the material temperature $T(x,t)$ is constant. When considering the radiation subsystem, there is little difference between the weak equilibrium diffusion and isothermal limits because the material quantities, including the material temperature T, do not evolve. T enters the radiation subsystem as a parameter rather than a dynamical quantity.

3. Higher-order Godunov method

In one spatial dimension, systems of conservation laws with source terms have the form

$$\frac{\partial U}{\partial t} + \frac{\partial F(U)}{\partial x} = S(U), \tag{11}$$

where $U : \mathbb{R} \times \mathbb{R} \to \mathbb{R}^n$ is an n-dimensional vector of conserved quantities. For the radiation subsystem,

$$U = \begin{pmatrix} E_r \\ F_r \end{pmatrix}, \quad F(U) = \begin{pmatrix} \mathbb{C}F_r \\ \mathbb{C}fE_r \end{pmatrix}, \quad S(U) = \begin{pmatrix} \mathbb{C}S_E \\ \mathbb{C}S_F \end{pmatrix} = \begin{pmatrix} \mathbb{C}\sigma_a(T^4 - E_r) \\ -\mathbb{C}\sigma_t F_r \end{pmatrix}.$$

The quasilinear form of this system of conservation laws is

$$\frac{\partial U}{\partial t} + A\frac{\partial U}{\partial x} = S(U), \quad A = \frac{\partial F}{\partial U} = \begin{pmatrix} 0 & \mathbb{C} \\ \mathbb{C}f & 0 \end{pmatrix}. \tag{12}$$

A has eigenvalues $\lambda = \pm f^{1/2}\mathbb{C}$ and it also has right eigenvectors R (stored as columns) and left eigenvectors L (stored as rows):

$$R = \begin{pmatrix} 1 & 1 \\ -f^{1/2} & f^{1/2} \end{pmatrix}, \quad L = \begin{pmatrix} \frac{1}{2} & -\frac{1}{2}f^{-1/2} \\ \frac{1}{2} & \frac{1}{2}f^{-1/2} \end{pmatrix}. \tag{13}$$

Godunov's method obtains solutions to systems of conservation laws by using characteristic information within the framework of a conservative method:

$$U_i^{n+1} = U_i^n - \frac{\Delta t}{\Delta x}(F_{i+1/2} - F_{i-1/2}) + \Delta t S(U_i^n). \tag{14}$$

Numerical fluxes $F_{i\pm 1/2}$ are obtained by solving the Riemann problem at the cell interfaces with left/right states to get $U_{i-1/2}^{n\pm 1/2}$ and computing

$$F_{i\pm 1/2} = F\big(U_{i\pm 1/2}^{n+1/2}\big),$$

where i represents the location of a cell center, $i \pm 1/2$ represents the location cell faces to the right and left of i, and superscripts represent the time discretization. An HLLE (Harten–Lax–van Leer–Einfeldt) solver, used in this work, or any other approximate Riemann solver may be employed because the Jacobian $\partial F/\partial U$ for the radiation subsystem is a constant valued matrix and by definition a Roe matrix [LeVeque 1992; 2002; Roe 1981]. This property also implies that one does not need to transform the system into primitive variables ($\nabla_U W$). The power of the method presented in this paper is that the spatial reconstruction, eigenanalysis, and cell-centered updating directly plug into conventional Godunov machinery.

3.1. ***Predictor step.*** One computes the flux divergence $(\nabla \cdot F)^{n+1/2}$ by using the quasilinear form of the system of conservation laws and the evolution along Lagrangian trajectories:

$$\frac{DU}{Dt} + A^L\frac{\partial U}{\partial x} = S(U), \quad A^L = A - uI, \quad \frac{DU}{Dt} = \frac{\partial U}{\partial t} + \Big(u\frac{\partial}{\partial x}\Big)U. \tag{15}$$

From the quasilinear form, one derives a system that includes (at least locally in time and state space) the effects of the stiff source terms on the hyperbolic structure.

Following [Miniati and Colella 2007; Trebotich et al. 2005], one applies Duhamel's principle to the system of conservation laws, thus giving

$$\frac{DU^{\text{eff}}}{Dt} = \mathcal{I}_{\dot{S}_n}(\eta)\Big(-A^L\frac{\partial U}{\partial x} + S_n\Big), \tag{16}$$

where $\mathcal{I}_{\dot{S}_n}$ is a propagation operator that projects the dynamics of the stiff source terms onto the hyperbolic structure and $\dot{S}_n = \nabla_U S|_{U_n}$. The subscript n designates time $t = t_n$. Since one is considering a first-order accurate predictor step in a second-order accurate predictor-corrector method, one chooses $\eta = \Delta t/2$ and the effective conservation law is

$$\frac{DU}{Dt} + \mathcal{I}_{\dot{S}_n}(\Delta t/2)A^L\frac{\partial U}{\partial x} = \mathcal{I}_{\dot{S}_n}(\Delta t/2)S_n$$

which implies

$$\frac{\partial U}{\partial t} + A_{\text{eff}}\frac{\partial U}{\partial x} = \mathcal{I}_{\dot{S}_n}(\Delta t/2)S_n, \tag{17}$$

where $A_{\text{eff}} = \mathcal{I}_{\dot{S}_n}(\Delta t/2)A^L + uI$. In order to compute $\mathcal{I}_{\dot{S}_n}$, one first computes $\dot{S}_n$. Since $\mathbb{C}$, σ_a, and σ_t are constant and one assumes that $\partial T/\partial E_r, \partial T/\partial F_r = 0$:

$$\dot{S}_n = \begin{pmatrix} -\mathbb{C}\sigma_a & 0 \\ 0 & -\mathbb{C}\sigma_t \end{pmatrix}. \tag{18}$$

$\mathcal{I}_{\dot{S}_n}$ is derived from Duhamel's principle and is given by

$$\mathcal{I}_{\dot{S}_n}(\Delta t/2) = \frac{1}{\Delta t/2}\int_0^{\Delta t/2} e^{\tau\dot{S}_n}d\tau = \begin{pmatrix} \alpha & 0 \\ 0 & \beta \end{pmatrix}, \tag{19}$$

with

$$\alpha = \frac{1-e^{-\mathbb{C}\sigma_a\Delta t/2}}{\mathbb{C}\sigma_a\Delta t/2}, \quad \beta = \frac{1-e^{-\mathbb{C}\sigma_t\Delta t/2}}{\mathbb{C}\sigma_t\Delta t/2}. \tag{20}$$

Before applying $\mathcal{I}_{\dot{S}_n}$ to A_L, it is important to understand that moving-mesh methods can be accommodated in nonrelativistic descriptions of radiation hydrodynamics whenever an Eulerian frame treatment is employed. These methods do not require transformation to the comoving frame [Lowrie and Morel 2001]. Since the nondimensionalization is associated with the hydrodynamic scale, one can use $u_{\text{mesh}} = u$ from Lagrangean hydrodynamic methods.

The effects of the stiff source terms on the hyperbolic structure are accounted for by transforming to a moving-mesh (Lagrangean) frame $A_L = A - uI$, applying the propagation operator $\mathcal{I}_{\dot{S}_n}$ to A_L, and transforming back to an Eulerian frame $A_{\text{eff}} = \mathcal{I}_{\dot{S}_n}A_L + uI$ [Miniati and Colella 2007]. However, because only the radiation subsystem of radiation hydrodynamics is considered $u_{\text{mesh}} = u \to 0$. Therefore, the effective Jacobian is given by

$$A_{\text{eff}} = \begin{pmatrix} 0 & \alpha\mathbb{C} \\ \beta f\mathbb{C} & 0 \end{pmatrix}, \tag{21}$$

which has eigenvalues $\lambda_{\text{eff}} = \pm(\alpha\beta)^{1/2} f^{1/2}\mathbb{C}$ with the limits

$$\begin{aligned} &\sigma_a, \sigma_t \to 0 \Rightarrow \alpha, \quad \beta \to 1 \Rightarrow \lambda_{\text{eff}} \to \pm f^{1/2}\mathbb{C} \quad \text{(free streaming)}, \\ &\sigma_a, \sigma_t \to \infty \Rightarrow \alpha, \quad \beta \to 0 \Rightarrow \lambda_{\text{eff}} \to \pm\epsilon \quad \text{(strong equilibrium diffusion)}. \end{aligned} \tag{22}$$

A_{eff} has right eigenvectors R_{eff} (stored as columns) and left eigenvectors L_{eff} (stored as rows):

$$R_{\text{eff}} = \begin{pmatrix} 1 & 1 \\ -(\beta f/\alpha)^{1/2} & (\beta f/\alpha)^{1/2} \end{pmatrix}, \quad L_{\text{eff}} = \begin{pmatrix} \frac{1}{2} & -\frac{1}{2}(\alpha/\beta f)^{1/2} \\ \frac{1}{2} & \frac{1}{2}(\alpha/\beta f)^{1/2} \end{pmatrix}. \tag{23}$$

3.2. ***Corrector step.*** The time discretization for the source term is a single-step, second-order accurate scheme based on the ideas from [Dutt et al. 2000; Minion 2003; Miniati and Colella 2007]. Given the system of conservation laws, one aims for a scheme that has an explicit approach for the conservative flux divergence term $\nabla \cdot F$ and an implicit approach for the stiff source term $S(U)$. Therefore, one solves a following collection of ordinary differential equations at each grid point:

$$\frac{dU}{dt} = S(U) - (\nabla \cdot F)^{n+1/2}, \tag{24}$$

where the time-centered flux divergence term is taken to be a constant source which is obtained from the predictor step. Assuming time $t = t_n$, the initial guess for the solution at the next time step is

$$\hat{U} = U^n + \Delta t (I - \Delta t \nabla_U S(U)|_{U^n})^{-1} (S(U^n) - (\nabla \cdot F)^{n+1/2}), \tag{25}$$

where

$$(I - \Delta t \nabla_U S(U)) = \begin{pmatrix} 1 + \Delta t\mathbb{C}\sigma_a & 0 \\ 0 & 1 + \Delta t\mathbb{C}\sigma_t \end{pmatrix}, \tag{26}$$

$$(I - \Delta t \nabla_U S(U))^{-1} = \begin{pmatrix} (1 + \Delta t\mathbb{C}\sigma_a)^{-1} & 0 \\ 0 & (1 + \Delta t\mathbb{C}\sigma_t)^{-1} \end{pmatrix}. \tag{27}$$

The error ϵ is defined as the difference between the initial guess and the solution obtained from the Picard iteration equation, where the initial guess was used as a starting value:

$$\epsilon(\Delta t) = U^n + \frac{\Delta t}{2}\big(S(\hat{U}) + S(U^n)\big) - \Delta t (\nabla \cdot F)^{n+1/2} - \hat{U}. \tag{28}$$

Following [Miniati and Colella 2007], the correction to the initial guess is given by

$$\delta(\Delta t) = (I - \Delta t \nabla_U S(U)|_{\hat{U}})^{-1} \epsilon(\Delta t). \tag{29}$$

Therefore, the solution at time $t = t_n + \Delta t$ is

$$U^{n+1} = \hat{U} + \delta(\Delta t). \tag{30}$$

3.3. ***Stability and algorithmic issues.*** The higher-order Godunov method satisfies important conditions that are required for numerical stability [Miniati and Colella 2007]. First, $\lambda_{\text{eff}} = \pm(\alpha\beta)^{1/2} f^{1/2}\mathbb{C}$ indicates that the subcharacteristic condition for the characteristic speeds at equilibrium is always satisfied, such that: $\lambda^- < \lambda^-_{\text{eff}} < \lambda^0 < \lambda^+_{\text{eff}} < \lambda^+$. This condition is necessary for the stability of the system and guarantees that the numerical solution tends to the solution of the equilibrium equation as the relaxation time tends to zero. Second, since the structure of the equations remains consistent with respect to classic Godunov methods, one expects the CFL (Courant–Friedrichs–Lewy) condition to apply: $\max(|\lambda^*|)(\Delta\text{t}/\Delta\text{x}) \leq 1$, for $* = -, 0, +$.

Depending upon how one carries out the spatial reconstruction to solve the Riemann problem in Godunov's method, the solution is either first-order accurate in space (piecewise constant reconstruction) or second-order accurate in space (piecewise linear reconstruction). Piecewise linear reconstruction was employed in this paper, where left/right states (with respect to the cell center) are modified to account for the stiff source term effects [Miniati and Colella 2007; Colella 1990]:

$$\begin{aligned} U^n_{i,\pm} &= U^n_i + \frac{\Delta t}{2}\mathcal{I}_{\dot{S}_n}\left(\frac{\Delta t}{2}\right)S(U^n_i) + \frac{1}{2}\left(\pm I - \frac{\Delta t}{\Delta x}A^n_{\text{eff}}\right)P_\pm(\Delta U_i), \\ P_\pm(\Delta U_i) &= \sum_{\pm\lambda_k > 0} (L^k_{\text{eff}} \cdot \Delta U_i) \cdot R^k_{\text{eff}}. \end{aligned} \tag{31}$$

Left/right one-sided slopes as well as cell center slopes are defined for each cell centered quantity U_i. A van Leer limiter is applied to these slopes to ensure monotonicity, thus giving the local slope ΔU_i.

4. Numerical tests

Four numerical tests spanning a range of mathematical and physical behavior were carried out to gauge the temporal and spatial accuracy of the higher-order Godunov method. The numerical solution is compared with the analytic solution where possible. Otherwise, a self-similar comparison is made. Using piecewise constant reconstruction for the left/right states, one can show that the Godunov method reduces to a consistent discretization in each of the limiting cases.

The optical depth τ is a useful quantity for classifying the limiting behavior of a system that is driven by radiation hydrodynamics:

$$\tau = \int_{x_{\min}}^{x_{\max}} \sigma_t dx = \sigma_t (x_{\max} - x_{\min}), \tag{32}$$

Optically thin/thick regimes are characterized by

$$\begin{aligned} \tau &< O(1) \quad \text{(optically thin)}, \\ \tau &> O(1) \quad \text{(optically thick)}. \end{aligned}$$

In optically thin regimes (free streaming limit), radiation and hydrodynamics decouple such that the resulting dynamics resembles an advection process. In optically thick regimes (weak/strong equilibrium diffusion limit), radiation and hydrodynamics are strongly coupled and the resulting dynamics resembles a diffusion process.

We use the following definitions for the norms and convergence rates throughout this paper. Given the numerical solution q^r at resolution r and the analytic solution u, the error at a given point i is: $\epsilon_i^r = q_i^r - u$. Likewise, given the numerical solution q^r at resolution r and the numerical solution q^{r+1} at the next finer resolution $r+1$ (properly spatially averaged onto the coarser grid), the error resulting from this self-similar comparison at a given point i is: $\epsilon_i^r = q_i^r - q_i^{r+1}$. The 1-norm and max-norm of the error are

$$L_1 = \sum_i |\epsilon_i^r| \Delta x^r, \qquad L_{\max} = \max_i |\epsilon_i^r|. \tag{33}$$

The convergence rate is measured using Richardson extrapolation:

$$R_n = \frac{\ln(L_n(\epsilon^r)/L_n(\epsilon^{r+1}))}{\ln(\Delta x^r/\Delta x^{r+1})}. \tag{34}$$

4.1. ***Exponential growth/decay to thermal equilibrium.*** The first numerical test examines the temporal accuracy of how variables are updated in the corrector step. Given the radiation subsystem and the initial conditions

$$E_r^0 = \text{constant across space}, \qquad F_r^0 = 0, \qquad T = \text{constant across space},$$

We have $F_r \to 0$ for all time. Therefore, the radiation subsystem reduces to the ordinary differential equation

$$\frac{dE_r}{dt} = \mathbb{C}\sigma_a (T^4 - E_r), \tag{35}$$

which has the analytic solution

$$E_r = T^4 + (E_r^0 - T^4)\exp(-\mathbb{C}\sigma_\mathrm{a}\mathrm{t}). \tag{36}$$

For $E_r^0 < T^4$ and $F_r^0 = 0$, one expects exponential growth in E_r until thermal equilibrium ($E_R = T^4$) is reached. For $E_r^0 > T^4$ and $F_r^0 = 0$, one expects exponential decay in E_r until thermal equilibrium is reached. This numerical test allows one to examine the order of accuracy of the stiff ODE integrator.

Parameters:

$$\mathbb{C} = 10^5, \quad \sigma_a = 1, \quad \sigma_t = 2, \quad f = 1,$$

$$N_{\text{cell}} = [32, 64, 128, 256],$$

$$x_{\min} = 0, \quad x_{\max} = 1, \quad \Delta x = \frac{x_{\min} - x_{\max}}{N_{\text{cell}}}, \quad \text{CFL} = 0.5, \quad \Delta t = \frac{\text{CFL}\,\Delta x}{f^{1/2}\mathbb{C}},$$

$$\text{IC for growth:} \quad E_r^0 = 1, \qquad F_r^0 = 0, \quad T = 10,$$

$$\text{IC for decay:} \quad E_r^0 = 10^4, \quad F_r^0 = 0, \quad T = 1.$$

From Figure 1, one sees that the numerical solution corresponds with the analytic solution. In Table 1 on the next page, the errors and convergence rates are see to be identical for growth and decay. This symmetry illustrates the robustness of the Godunov method. Furthermore, one finds that the method is well behaved and obtains the correct solution with second-order accuracy for stiff values of the e folding time ($\Delta t\, \sigma_a \mathbb{C} \geq 1$), although with a significantly larger amplitude in the norm of the error. This result credits the flexibility of the temporal integrator in the corrector step.

In a similar test, the initial conditions for the radiation energy and flux are zero and the temperature is defined by some spatially varying profile (a Gaussian pulse). As time increases, the radiation energy grows into $T(x)^4$. Unless the opacity is sufficiently high, the radiation energy approaches but does not equal $T(x)^4$. This

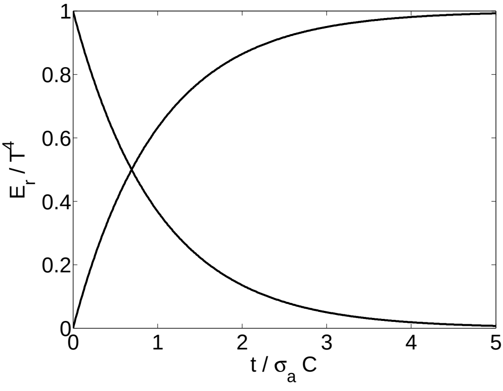


Figure 1. Exponential growth/decay to thermal equilibrium; $N_{\text{cell}} = 256$.

N_{cell}	$L_1(E_r^g)$	Rate	$L_\infty(E_r^g)$	Rate	$L_1(E_r^d)$	Rate	$L_\infty(E_r^d)$	Rate
32	1.4E-1	–	1.4E-1	–	1.4E-1	–	1.4E-1	–
64	3.7E-2	2.0	3.7E-2	2.0	3.7E-2	2.0	3.7E-2	2.0
128	9.3E-3	2.0	9.3E-3	2.0	9.3E-3	2.0	9.3E-3	2.0
256	2.3E-3	2.0	2.3E-3	2.0	2.3E-3	2.0	2.3E-3	2.0

Table 1. Errors and convergence rates for exponential growth and decay in E_r to thermal equilibrium. Errors were obtained through analytic comparison. $t = 10^{-5} = 1/\sigma_a \mathbb{C}$.

result shows that the solution has reached thermal equilibrium and any spatially varying temperature will diffuse.

4.2. ***Free streaming limit.*** In the free streaming limit, $\tau \ll O(1)$ and the radiation subsystem reduces to (8). If one takes an additional temporal and spatial partial derivative of the radiation subsystem in the free streaming limit and subtracts the resulting equations, then one finds two decoupled wave equations that have the analytic solutions

$$E_r(x,t) = E_0(x - f^{1/2}\mathbb{C}t), \quad F_r(x,t) = F_0(x - f^{1/2}\mathbb{C}t). \tag{37}$$

Parameters:

$$\mathbb{C} = 10^5, \quad \sigma_a = 10^{-6}, \quad \sigma_t = 10^{-6}, \quad f = 1, \quad T = 1,$$

$$N_{\text{cell}} = [32, 64, 128, 256],$$

$$x_{\min} = 0, \quad x_{\max} = 1, \quad \Delta x = \frac{x_{\min} - x_{\max}}{N_{\text{cell}}}, \quad \text{CFL} = 0.5, \quad \Delta t = \frac{\text{CFL}\,\Delta x}{f^{1/2}\mathbb{C}},$$

$$\text{IC for Gaussian pulse:} \quad E_r^0, F_r^0 = \exp(-(\nu(x-\mu))^2), \quad \nu = 20, \quad \mu = 0.3,$$

$$\text{IC for square pulse:} \quad E_r^0, F_r^0 = \begin{cases} 1 & \text{if } 0.2 < x < 0.4, \\ 0 & \text{otherwise.} \end{cases}$$

Since the Gaussian pulse results from smooth initial data, one expects $R_1 = 2.0$. However, the square wave results from discontinuous initial data and one expects $R_1 \simeq 0.67$. This is true for all second-order spatially accurate numerical methods when applied to an advection-type problem ($u_t + au_x = 0$) [LeVeque 1992]. See Figure 2 for the shape of the pulses in the free streaming limit, and Table 2 for the corresponding errors and convergence rates.

4.3. ***Weak equilibrium diffusion limit.*** In the weak equilibrium diffusion limit, $\tau > O(1)$ and the radiation subsystem reduces to (9). The optical depth suggests the range of total opacities for which diffusion is observed: if $\tau = \sigma_t\ \ell_{\text{diff}} > 1$,

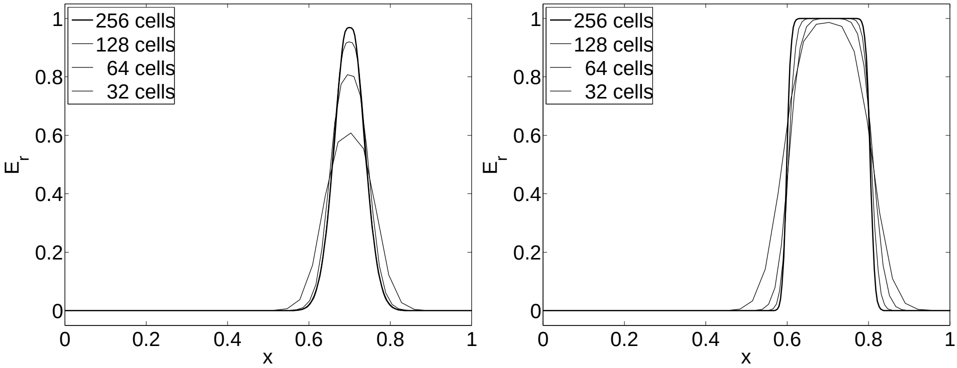


Figure 2. Gaussian pulse (left) and square pulse (right) in free streaming limit; $t = 4 \times 10^{-6} = 0.4\,(x_{\max} - x_{\min})/\mathbb{C}$.

Gaussian pulse

N_{cell}	$L_1(E_r)$	Rate	$L_\infty(E_r)$	Rate	$L_1(F_r)$	Rate	$L_\infty(F_r)$	Rate
32	3.8E-2	–	3.9E-1	–	3.8E-2	–	3.9E-1	–
64	1.3E-2	1.5	1.8E-1	1.1	1.3E-2	1.5	1.8E-1	1.1
128	3.6E-3	1.9	8.0E-2	1.2	3.6E-3	1.9	8.0E-2	1.2
256	8.6E-4	2.1	3.1E-2	1.4	8.6E-4	2.1	3.1E-2	1.4

square pulse

N_{cell}	$L_1(E_r)$	Rate	$L_1(F_r)$	Rate
32	6.0E-2	–	6.0E-2	–
64	4.2E-2	0.5	4.2E-2	0.5
128	2.6E-2	0.7	2.6E-2	0.7
256	1.5E-2	0.8	1.5E-2	0.8

Table 2. Errors (obtained through analytic comparison) and convergence rates for Gaussian and square pulses in free streaming limit; $t = 4 \times 10^{-6} = 0.4(x_{\max} - x_{\min})/\mathbb{C}$.

then one expects diffusive behavior for $\sigma_t > 1/\ell_{\text{diff}}$. Additionally, Equation (9) sets the time scale t_{diff} and length scale ℓ_{diff} for diffusion, where $t_{\text{diff}} \sim \ell_{\text{diff}}^2/D$ and $D = f\mathbb{C}/\sigma_t$ for the radiation subsystem. Given a diffusion problem for a Gaussian pulse defined over the entire real line ($u_t - Du_{xx} = 0$), the analytic solution is given by the method of Green's functions:

$$u(x,t) = \int_{-\infty}^{\infty} f(\bar{x}) G(x,t;\bar{x},0)\,d\bar{x} = \frac{1}{(4Dt\nu^2+1)^{1/2}} \exp\Bigl(\frac{-(\nu(x-\mu))^2}{4Dt\nu^2+1}\Bigr). \quad (38)$$

Parameters:

$$\mathbb{C} = 10^5, \quad \sigma_a = 40, \quad \sigma_t = 40, \quad f = 1/3, \quad T^4 = E_r,$$

$$N_{\text{cell}} = [320, 640, 1280, 2560],$$

$$x_{\min} = -5, \quad x_{\max} = 5, \quad \Delta x = \frac{x_{\min} - x_{\max}}{N_{\text{cell}}}, \quad \text{CFL } = 0.5, \quad \Delta t = \frac{\text{CFL}\,\Delta x}{f^{1/2}\mathbb{C}},$$

$$\text{IC for Gaussian pulse: } \begin{cases} E_r^0 = \exp\left(-(\nu(x-\mu))^2\right), \ \nu = 20, \ \mu = 0.3, \\ F_r^0 = -\dfrac{f}{\sigma_t}\dfrac{\partial E_r^0}{\partial x} = \dfrac{2f\nu^2(x-\mu)}{\sigma_t}E_r^0 \end{cases}$$

One's intuition about diffusive processes is based on an infinite domain. So to minimize boundary effects in the numerical calculation, the computational domain and number of grid cells were expanded by a factor of 10. In Figure 3, one observes the diffusive behavior expected for this parameter regime. Additionally, the numerical solution compares well with the analytic solution for a diffusion process defined over the entire real line (38). However, diffusive behavior is only a first-order approximation to more complicated hyperbolic-parabolic dynamics taking place in radiation hydrodynamics as well as the radiation subsystem. Therefore, one needs to compare the numerical solution self-similarly. In Table 3, one sees convergence results for two different time steps: a hyperbolic time step $\Delta t_h = \text{CFL } \Delta x/(f^{1/2}\mathbb{C})$, and parabolic one, $\Delta t_p = \text{CFL } (\Delta x)^2/(2D)$. This difference in the convergence rate results from the temporal accuracy in the numerical solution. In the weak equilibrium diffusion limit, the Godunov method reduces to a forward-time/centered-space discretization of the diffusion equation. Such a discretization requires a parabolic time step $\Delta t \sim (\Delta x)^2$ in order to see second-order convergence because the truncation error of the forward-time/centered-space discretization of the diffusion equation is $\mathbb{O}(\Delta t, (\Delta x)^2)$.

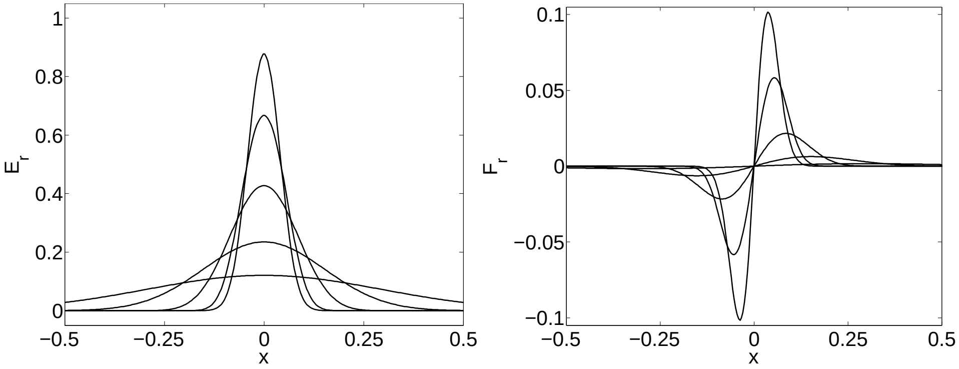


Figure 3. E_r (left) and F_r (right) in weak equilibrium diffusion limit; $t = [0.25, 1, 4, 16, 64] \times 10^{-6}$.

Hyperbolic time step: $\Delta t_h = \text{CFL } \Delta x/(f^{1/2}\mathbb{C})$

N_{cell}	$L_1(E_r)$	Rate	$L_\infty(E_r)$	Rate	$L_1(F_r)$	Rate	$L_\infty(F_r)$	Rate
320	8.9E-3	–	4.5E-2	–	1.1E-3	–	3.7E-3	–
640	6.6E-3	0.4	3.4E-2	0.4	8.3E-4	0.4	3.1E-3	0.2
1280	3.4E-3	1.0	1.6E-2	1.1	4.1E-4	1.0	1.4E-3	1.2
2560	1.6E-3	1.1	7.1E-3	1.1	1.9E-4	1.1	6.0E-4	1.2

Parabolic time step: $\Delta t_p = \text{ CFL } (\Delta x)^2/(2D)$

N_{cell}	$L_1(E_r)$	Rate	$L_\infty(E_r)$	Rate	$L_1(F_r)$	Rate	$L_\infty(F_r)$	Rate
320	1.7E-2	–	8.3E-2	–	2.0E-3	–	7.9E-3	–
640	5.0E-3	1.7	2.5E-2	1.7	6.0E-4	1.7	2.0E-3	2.0
1280	1.1E-3	2.2	5.1E-3	2.3	1.3E-4	2.3	3.6E-4	2.4
2560	2.5E-4	2.1	1.2E-3	2.1	2.8E-5	2.2	7.4E-5	2.3

Table 3. Errors (obtained through analytic comparison) and convergence rates for E_r and F_r in the weak equilibrium diffusion limit, when time is advanced according to each indicated scheme; $t = 4 \times 10^{-6}$.

4.4. ***Strong equilibrium diffusion limit.*** In the strong equilibrium diffusion limit, $\tau \gg O(1)$. From (10), we have $F_r \to 0$ for all time and space while $E_r = E_r^0$.

Parameters:

$$\mathbb{C} = 10^5, \quad \sigma_a = 10^6, \quad \sigma_t = 10^6, \quad f = 1/3, \quad T^4 = E_r,$$
$$N_{\text{cell}} = [320, 640, 1280, 2560],$$
$$x_{\min} = -5, \quad x_{\max} = 5, \quad \Delta x = \frac{x_{\min} - x_{\max}}{N_{\text{cell}}}, \quad \text{CFL} = 0.5, \quad \Delta t = \frac{\text{CFL}\Delta x}{f^{1/2}\mathbb{C}},$$

$$\text{IC for Gaussian Pulse:} \quad \begin{cases} E_r^0 = \exp\left(-(\nu(x-\mu))^2\right), \ \nu = 20, \ \mu = 0.3, \\ F_r^0 = -\dfrac{f}{\sigma_t}\dfrac{\partial E_r^0}{\partial x} = \dfrac{2f\nu^2(x-\mu)}{\sigma_t} E_r^0 \end{cases}$$

In this test, the numerical solution is held fixed at the initial distribution because σ_a, σ_t are so large. However, if one fixed ℓ_{diff} and scaled time according to

$$t_{\text{diff}} \approx \ell_{\text{diff}}^2/D = \ell_{\text{diff}}^2 \sigma_t/f\mathbb{C},$$

then one would observe behavior similar to Figure 3. This test illustrates the robustness of the Godunov method to handle very stiff source terms. (See Table 4 for errors and convergence rates.)

N_{cell}	$L_1(E_r)$	Rate	$L_\infty(E_r)$	Rate
320	2.2E-3	–	1.8E-2	–
640	5.3E-4	2.1	5.6E-3	1.6
1280	1.3E-4	2.0	1.5E-3	1.9
2560	3.3E-5	2.0	3.8E-4	2.0

Table 4. Errors and convergence rates for E_r in the strong equilibrium diffusion limit. Errors were obtained through self-similar comparison. $t = 4 \times 10^{-6}$.

5. Conclusions and future work

This paper presents a Godunov method for the radiation subsystem of radiation hydrodynamics that is second-order accurate in both time and space, unsplit, asymptotically preserving, and uniformly well behaved. Moreover, the method employs familiar algorithmic machinery without a significant increase in computational cost. This work is the starting point for developing a Godunov method for full radiation hydrodynamics. The ideas in this paper should easily extend to the full system in one and multiple dimensions using the MUSCL (monotone upstream-centered schemes for conservation laws) or CTU (corner transport upwind) approaches of [Colella 1990]. A modified Godunov method that is explicit on the fastest hyperbolic scale (radiation flow) as well as a hybrid method that incorporates a backward Euler upwinding scheme for the radiation components and the modified Godunov scheme for the material components are under construction for full radiation hydrodynamics. A goal of future research is to directly compare these two methods in various limits for different values of c/a_∞. Nevertheless, one expects the modified Godunov method that is explicit on the fastest hyperbolic scale to exhibit second-order accuracy for all conservative variables and the hybrid method to exhibit first-order accuracy in the radiation variables and second-order accuracy in the material variables. Work is also being conducted on applying short characteristic and Monte Carlo methods to solve the photon transport equation and obtain the variable tensor Eddington factors. In the present work, these factors were taken to be constant in their respective limits.

Acknowledgment

The authors thank Dr. Phillip Colella for many helpful discussions. MS acknowledges support from the DOE CSGF Program which is provided under grant DE-FG02-97ER25308. JS acknowledges support from grant DE-FG52-06NA26217.

References

[Balsara 1999] D. S. Balsara, "Linearized formulation of the Riemann problem for radiation hydrodynamics", *Journal of Quant. Spect. and Radiative Transfer* **61** (1999), 629–635.

[Buet and Despres 2006] C. Buet and B. Despres, "Asymptotic preserving and positive schemes for radiation hydrodynamics", *J. Comput. Phys.* **215**:2 (2006), 717–740. MR 2007j:85005 Zbl 1090.76046

[Castor 1972] J. I. Castor, "Radiative transfer in spherically symmetric flows", *Astrophys. Journal* **178** (1972), 779–792.

[Castor 2004] J. I. Castor, *Radiation Hydrodynamics*, Cambridge University Press, Cambridge, 2004.

[Colella 1990] P. Colella, "Multidimensional upwind methods for hyperbolic conservation laws", *J. Comput. Phys.* **87**:1 (1990), 171–200. MR 91c:76087 Zbl 0694.65041

[Dai and Woodward 1998] W. Dai and P. R. Woodward, "Numerical simulations for radiation hydrodynamics, I: Diffusion limit", *J. Comput. Phys.* **142**:1 (1998), 182–207. MR 99a:76094 Zbl 0933.76057

[Dai and Woodward 2000] W. W. Dai and P. R. Woodward, "Numerical simulations for radiation hydrodynamics. II. Transport limit", *J. Comput. Phys.* **157**:1 (2000), 199–233. MR 2000j:76110 Zbl 0941.76060

[Dutt et al. 2000] A. Dutt, L. Greengard, and V. Rokhlin, "Spectral deferred correction methods for ordinary differential equations", *BIT* **40**:2 (2000), 241–266. MR 2001e:65104 Zbl 0959.65084

[Gonzalez et al. 2007] M. Gonzalez, E. Audit, and P. Huynh, "HERACLES: a three-dimensional radiation hydrodynamics code", *Astron. and Astrophys.* **464** (2007), 429–435.

[Jin and Levermore 1996] S. Jin and C. D. Levermore, "Numerical schemes for hyperbolic conservation laws with stiff relaxation terms", *J. Comput. Phys.* **126**:2 (1996), 449–467. MR 97g:65173 Zbl 0860.65089

[LeVeque 1992] R. J. LeVeque, *Numerical methods for conservation laws*, 2nd ed., Lectures in Mathematics ETH Zürich, Birkhäuser Verlag, Basel, 1992. MR 92m:65106 Zbl 0723.65067

[LeVeque 2002] R. J. LeVeque, *Finite volume methods for hyperbolic problems*, Cambridge Texts in Applied Mathematics, Cambridge University Press, Cambridge, 2002. MR 2003h:65001 Zbl 1010.65040

[Levermore and Pomraning 1981] C. D. Levermore and G. C. Pomraning, "A flux-limited diffusion theory", *Astrophys. Journal* **248** (1981), 321–334.

[Lowrie and Morel 2001] R. B. Lowrie and J. E. Morel, "Issues with high-resolution Godunov methods for radiation hydrodynamics", *Journal of Quant. Spect. and Radiative Transfer* **69** (2001), 475–489.

[Lowrie et al. 1999] R. B. Lowrie, J. E. Morel, and J. A. Hittinger, "The coupling of radiation and hydrodynamics", *Astrophys. Journal* **521** (1999), 432–450.

[Mihalas and Klein 1982] D. Mihalas and R. Klein, "Solution of the time-dependent inertial-frame equation of radiative transfer in moving media to O(v/c)", *J Comp Phys* **46** (1982), 97–137.

[Mihalas and Mihalas 1984] D. Mihalas and B. W. Mihalas, *Foundations of radiation hydrodynamics*, Oxford University Press, New York, 1984. MR 86h:85004 Zbl 0651.76005

[Miniati and Colella 2007] F. Miniati and P. Colella, "A modified higher order Godunov's scheme for stiff source conservative hydrodynamics", *J. Comput. Phys.* **224**:2 (2007), 519–538. MR 2008c:76068 Zbl 1117.76039

[Minion 2003] M. L. Minion, "Semi-implicit spectral deferred correction methods for ordinary differential equations", *Commun. Math. Sci.* **1**:3 (2003), 471–500. MR 2005f:65085 Zbl 1088.65556

[Pomraning 1973] G. C. Pomraning, *The Equations of Radiation Hydrodynamics*, Pergamon Press, Oxford, 1973.

[Roe 1981] P. L. Roe, "Approximate Riemann solvers, parameter vectors, and difference schemes", *J. Comput. Phys.* **43**:2 (1981), 357–372. MR 82k:65055 Zbl 0474.65066

[Stone et al. 1992] J. M. Stone, D. Mihalas, and M. L. Norman, "ZEUS-2D: a radiation magnetohydrodynamics code for astrophysical flows in two space dimensions, III. The radiation hydrodynamic algorithms and tests", *Astrophys. Journal Supp.* **80** (1992), 819–845.

[Stone et al. 2008] J. M. Stone, T. A. Gardiner, P. Teuben, J. F. Hawley, and J. B. Simon, "Athena: a new code for astrophysical MHD", *Astrophys. Journal Supp. Ser.* **178** (2008), 137–177.

[Trebotich et al. 2005] D. Trebotich, P. Colella, and G. H. Miller, "A stable and convergent scheme for viscoelastic flow in contraction channels", *J. Comput. Phys.* **205**:1 (2005), 315–342. MR 2132311 Zbl 1087.76005

Received February 18, 2008. Revised November 28, 2008.

MICHAEL DAVID SEKORA: sekora@math.princeton.edu
Program in Applied and Computational Mathematics, Princeton University, Princeton, NJ 08540, United States

JAMES M. STONE: jstone@astro.princeton.edu
Department of Astrophysical Sciences, Princeton University, Princeton, NJ 08540, United States

COMM. APP. MATH. AND COMP. SCI.
Vol. 4, No. 1, 2009

SHEAR FLOW LAMINARIZATION AND ACCELERATION BY SUSPENDED HEAVY PARTICLES: A MATHEMATICAL MODEL AND GEOPHYSICAL APPLICATIONS

GRIGORY ISAAKOVICH BARENBLATT

A modified model of turbulent shear flow of a suspension of small heavy particles in a fluid is presented. The modification is based on the assumption that in the flow there are two sorts of particles. For the particles of the first sort the velocity of free fall a_1 is larger than the characteristic velocity fluctuation, for the particles of the second sort the velocity of free fall a_2 is less than the characteristic velocity of fluctuation.

Introduction

The energy of turbulent vortices (energy of turbulence) in a horizontal or slightly inclined shear flow is reduced by suspended heavy particles, and this reduction leads to flow acceleration. The basic model of this seemingly paradoxical phenomenon was suggested by A.N. Kolmogorov (see [13]), and developed quantitatively by the present author ([1; 2], see also the book of Monin and Yaglom [15], pp. 412–416). Later this model, properly modified, was applied to several natural flow phenomena, in particular to dust storms, both terrestrial and Martian [10], and lower quasi-homogeneous layers of the ocean [8; 9]. It is important to mention that in the basic model and its applications it was always assumed that the particles are identical.

Meanwhile, Sir James Lighthill (see his published paper [14]) proposed the "sandwich model" of tropical hurricanes. A detailed analysis of the observations (especially of the expedition on the Russian vessel "Priliv") led Lighthill to the fundamental assumption that a specific feature of hurricanes is the availability of an intermediate layer between the sea and air; Lighthill called it "ocean spray". In this layer, air is filled by suspended water droplets, formed during the process of the breaking of surface water waves. Lighthill specially emphasized "the need to fill the gaps in knowledge about ocean spray at extreme wind speeds".

MSC2000: 76F10.

Keywords: turbulence, turbulent shear flows, laminarization of turbulent flows, dust storms, tropical hurricanes, firestorms.

Following the direct suggestion of M.J. Lighthill, the original basic model was applied by A.J. Chorin, V.M. Prostokishin and the present author [7] to the flow in ocean spray. The main result obtained in this paper is that indeed the droplets accelerate the wind, and, if they are large, "ocean spray" plays the role of a *lubrication layer* for the wind: that is the reason for the wind acceleration. However, the calculated increase of wind velocity happened to be less than was expected.

In the present paper a modified model is proposed for ocean spray. The key point of the modification is that it is assumed that in ocean spray there are droplets of different sizes: small and large ones. The most important result, obtained using this assumption, is that ocean spray is acting not only as a lubrication layer for the wind, but also as a source of smaller droplets which are suspended by the wind and which suppress the turbulence in the core of the air flow. Suppression of turbulence by small droplets in the core of the wind is, according to the modified model, the basic cause of extreme wind speeds.

The same modified model can be suggested for dust storms and for large fires, in particular, forest and grass fires. In particular it allows us to understand the nature of the firestorms observed in great fires (e.g., Chicago 1871, Dresden, 1945, and Hiroshima, 1945). These topics are considered in the present paper.

1. Kolmogorov's example

A.N. Kolmogorov, whose ideas shaped the modern theory of turbulence, posed at the beginning of his course on turbulence at Moscow State University in 1954 the following question: what would the velocity be at the surface of the river Volga (in Russia, its parameters are close to those of the Mississippi River), if, by a miracle, the river, having preserved its geometry, became laminar. It was clear for the listeners that the velocity at the surface will increase, but to what extent?

Naturally, Kolmogorov modelled the river by a weakly inclined (the slope i is small, $i \ll 1$), spatially homogeneous open channel (Figure 1a). In this simple case of a laminar flow in a channel the basic Navier-Stokes equations are reduced to a single equation, and the easily obtainable solution to this equation has the form

$$u = \frac{\rho g i H^2}{2\eta}\Big(\frac{2z}{H} - \frac{z^2}{H^2}\Big). \tag{1-1}$$

Here u is the velocity component along the bottom, η the dynamic viscosity of water, ρ its density, z is the coordinate perpendicular to the bottom, and reckoned from it, and g is the acceleration of gravity, so the the velocity u_{surf} at the surface

$z = H$ is equal to

$$u_{\mathrm{surf}} = \frac{\rho g i H^2}{2\eta}. \tag{1-2}$$

Now, substitute into (1.2) realistic values of the parameters: $\eta/\rho = 10^{-2}\,cm^2/s$, $H = 20\,m = 2 \cdot 10^3\,cm$, $i = 10^{-4}$, $g = 10^3\,cm/s^2$. We obtain a value $u_{\mathrm{surf}} = 2 \cdot 10^7 \mathrm{cm}/s = 200\mathrm{km}/s \cong 400,000$ miles per hour! The reason for this obviously absurd result is that the flow in the river is not laminar, it is "stuffed" with vortices (Figure 1b). These vortices make the flow field random; they transfer the momentum across the flow immensely faster than the thermal oscillations of the water molecules in the laminar flow (which is the mechanism of the molecular fluid viscosity). This basic idea was introduced by the French applied mathematician J. Boussinesq, who in fact was the first to study turbulent flows mathematically. Boussinesq introduced the basic concept of the "eddy viscosity" (viscosité tourbillonaire) η_{turb}: the effective viscosity of the turbulent flow, created by vortices which remains one of the basic concepts in turbulence studies. We emphasize that contrary to the molecular viscosity η, the eddy viscosity η_{turb} is no longer a *fluid property*, it is a *flow property*, different at different places. In the present case the value of η_{turb} needed in (1.2) to obtain a realistic value of the velocity at the river surface is $\sim 200,000\eta$!

However, the Kolmogorov example is especially significant, even fundamental due to the following reason. It demonstrates clearly the huge reserves of energy available in natural fluid flows.

These reserves can be revealed if somehow even a partial flow laminarization is achieved. And this happens in reality: such partial laminarization is achieved in dust storms (the laminarizing factor is the suspended dust particles), tropical hurricanes (the laminarizing factor is the water droplets formed on the oceanic surface when

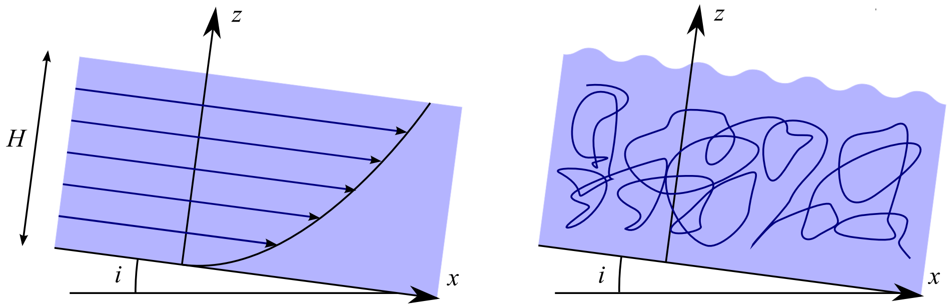


Figure 1. Kolmogorov's example (a) Laminar flow in a channel, (b) Turbulent flow in a river

water waves are breaking), firestorms (the laminarizing factor is unburnt debris and soot particles), and other natural flows.

2. Turbulent shear flows. The Kolmogorov-Prandtl model

Turbulent flows are random, and turbulence studies operate with the averages of flow field properties. In theoretical studies the "ensemble", or "probability" averages are used (the averages over the whole ensemble of possible turbulent flow realizations under given external conditions (e.g., pressure drop at the ends of a pipe).

Shear flow is a steady flow, homogeneous in the direction of average velocity. All properties of the shear flow field vary only in the direction z perpendicular to the direction of the mean velocity.

Studies of turbulent shear flows are of special importance for theoretical and experimental investigations. They allow us, due to substantial simplifications, to advance deeper without accepting doubtful assumptions. Indeed, in general turbulent flows are non-local, both in time and space: The properties of a flow field at a certain point and at a certain moment depend on the flow properties in a certain neighborhood around the point, and at a certain time interval. This is not the case for shear flows: The average flow field at a point can be assumed to be a local property, depending only upon the flow characteristics at this point. Also, an important advantage of shear flows from a practical viewpoint is that ensemble (probability) averages can be replaced (the "ergodicity" property) by averaging over time intervals (due to steadiness) or longitudinal space intervals (due to the spacial homogeneity along the flow direction).

By averaging the Navier-Stokes equations and integrating we obtain only one equation for shear flows due to steadiness and the homogeneity of the mean flow:

$$\frac{d}{dz}(-\rho\overline{u'w'}) = 0, \qquad -\rho\overline{u'w'} = \mathrm{Const} = \tau. \tag{2-1}$$

Here the velocity components u, v, w correspond to the axis x, y, z; bars denote the mean values and primes denotethe fluctuations. In Equation (2.1) the contribution of the molecular viscosity was neglected in comparison with the contribution of the eddy viscosity: we consider the turbulence as a "developed" one. The term neglected is important in the close vicinity of the wall which we exclude from consideration. The term $-\rho\overline{u'w'}$ represents the turbulent flux of momentum, the "Reynolds stress".

Of fundamental importance for future consideration is the equation of turbulent energy balance. For shear flow the equation obtained in this way assumes the form

$$(-\rho\overline{u'w'})\frac{du}{dz} - \rho\epsilon = 0. \tag{2-2}$$

The first term is the rate of inflow of turbulent energy per unit volume from the energy of mean motion, and

$$\epsilon = \frac{\nu}{2}\overline{(\partial_\alpha u'_\beta + \partial_\beta u'_\alpha)(\partial_\alpha u'_\beta + \partial_\beta u'_\alpha)} \tag{2-3}$$

(summation by Greek indexes from 1 to 3 is assumed) is the rate of turbulent energy dissipation into heat per unit mass.

In Equation (2.2) the term neglected is responsible for the contribution of turbulent diffusion of turbulent energy. This assumption is plausible in the main core of shear flow, but not close to the boundaries (see e.g. Monin and Yaglom, 1971).

In the Kolmogorov[12]-Prandtl[16] semi-empirical theory for shear flow, the coefficient of turbulent momentum exchange, $k = (-\rho\overline{u'w'})/\rho(du/dz)$, is the kinematic eddy viscosity. This introduction for shear flow is not a new hypothesis. Equations (2.1) and (2.2) take the form

$$k\frac{du}{dz} = {u_*}^2, \qquad k\left(\frac{du}{dz}\right)^2 - \epsilon = 0. \tag{2-4}$$

Here the quantity $u_* = (\tau/\rho)^{1/2}$ is an important governing parameter of shear flow: “dynamic” or “friction” velocity.

The basic hypothesis underlying the Kolmogorov-Prandtl theory can be presented in the following way: at large Reynolds numbers the local structure of the set of vortices around any point is statistically identical for all shear flows at a given Reynolds number; only the time and space scales are different. Therefore, leaving aside the Reynolds number dependence, all dimensionless flow properties should be identical. This means that all kinematic flow properties at a certain point including the momentum exchange coefficient k and the dissipation rate ϵ could be determined by the local values of two kinematic properties having different dimensions. Properties such as the turbulent energy of the unit mass

$$b = \frac{\overline{u'^2 + v'^2 + w'^2}}{2} \tag{2-5}$$

and the external length scale (mean length scale of vortices) ℓ, can be selected (b, ℓ version). Also in wide use is the (b, ϵ) version, where as basic quantities b and ϵ—the dissipation rate—are selected. We will use the (b, ℓ) version, in fact both versions are logically equivalent.

Dimensional analysis leads to the following relations:

$$k = \ell\sqrt{b}, \qquad \epsilon = \gamma^4 b^{3/2}/\ell. \tag{2-6}$$

The coefficient in the first Equation (2.6) can be selected equal to one because the length scale is determined with accuracy up to a constant factor. The constant γ

is a Reynolds number-dependent quantity; at large Reynolds numbers this quantity is close to 0.5 (see the book of Monin and Yaglom [15]).

Thus, Equations (2.4) assume the form

$$\ell\sqrt{b}\frac{du}{dz} = {u_*}^2, \qquad \ell\sqrt{b}\left(\frac{du}{dz}\right)^2 - \gamma^4\frac{b^{3/2}}{\ell} = 0. \tag{2-7}$$

It is important that from Equations (2.7) without any assumptions concerning the length scale ℓ, the relation for turbulent energy can be obtained:

$$b = \frac{u_*^2}{\gamma^2}. \tag{2-8}$$

Relation (2.8) shows that the dynamic, or friction velocity u_*, determines the order of magnitude of the velocity fluctuations.

Thus, if the length scale is known, the mean velocity u can be easily obtained from the first equation of the system. The situation of determining the length scale is, however, non-trivial. Using dimensional analysis a relation is obtained

$$\ell = z\Phi\left(\mathrm{Re}, \frac{u_*z}{\nu}\right). \tag{2-9}$$

We remind the reader that shear flow at large Reynolds numbers is considered, and also that the value u_*z/ν is large outside the close vicinity of the boundary $z = 0$, which, as was mentioned before, is excluded from consideration. Therefore, traditionally "complete" similarity (see, e.g., [4]) in both parameters Re and u_*z/ν is assumed. This means that function Φ can be replaced by its limit $\Phi(\infty, \infty) = \kappa\gamma$, which is assumed to be finite. The new constant κ is known as the von Kármán constant. The relation $\ell = \kappa\gamma z$ and relation (2.8) are substituted into the first Equation of (2.7), and the resulting relation is integrated, so the equation traditionally obtained is

$$\frac{u}{u_*} = \frac{1}{\kappa}\ln\left(\frac{u_*z}{\nu}\right) + B, \tag{2-10}$$

known as the universal (Reynolds number-independent) von Kármán-Prandtl logarithmic law. It is also tacitly assumed that the constant B is finite and Reynolds number-independent. The values $\kappa = 0.4$, $B = 5.1$ are usually accepted, although large deviations from these values have been reported in processing the experimental data.

However, as it was shown in a cycle of works by A.J. Chorin, V.M. Prostokishin and the present author (see [5; 6] and monograph [4] as well as the references presented there) this is not the case. There is "incomplete similarity" (see e.g. [4]) in parameter u_*z/ν and no similarity in parameter Re. In fact, at large Reynolds numbers and large u_*z/ν the mean velocity is represented by a family of Reynolds number-dependent power laws:

$$\frac{u}{u_*} = \left(\frac{1}{\sqrt{3}}\ln\mathrm{Re} + \frac{5}{2}\right)\left(\frac{u_* z}{\nu}\right)^{3/2\ln\mathrm{Re}}. \tag{2-11}$$

Furthermore, in the basic working interval of $u_* z/\nu$, the family of velocity distribution curves (2.11) can be represented in the form of a *Reynolds number-dependent* logarithmic law:

$$\frac{u}{u_*} = \frac{1}{\kappa(\mathrm{Re})}\ln\left(\frac{u_* z}{\nu}\right) + B(\mathrm{Re}), \tag{2-12}$$

where

$$\kappa(\mathrm{Re}) = \frac{e^{-3/2}}{\sqrt{3}/2 + 15/(4\ln\mathrm{Re})}, \qquad B(\mathrm{Re}) = -\frac{e^{3/2}\ln\mathrm{Re}}{2\sqrt{3}} - \frac{5}{4}e^{3/2}. \tag{2-13}$$

We mention several important properties of (2.12), (2.13). Firstly, at Re → ∞ the quantity $\kappa(\mathrm{Re})$ tends to a limit $\kappa_\infty = 2\sqrt{3}e^{3/2} \simeq 0.2776$. However, this tendency to the limit is very slow, so approximating the limiting value of κ_∞ with accuracy, for example, 10% κ requires huge values of Re, $\mathrm{Re} \sim 10^{20}$. For realistic lower values of Re, $\kappa(\mathrm{Re})$ are significantly less than κ_∞, so the slope of the straight line u/u_* vs $\ln(u_* z/\nu)$ is steeper than the slope of the straight line representing the usually accepted universal logarithmic law. *At the same time the additive constant* $B(\mathrm{Re})$ *at* $\mathrm{Re} \to \infty$ *tends not to a finite limit but to minus infinity.* All that means is that at large but realistic Re the universal (Reynolds number-independent) law for velocity distribution is not valid, although the velocity distributions in the $\ln(u_* z/\nu)$, u/u_* plane are represented by a family of Reynolds number-dependent straight lines (logarithmic laws) in the significant interval of the values of $u_* z/\nu$. These properties of velocity distributions obtained an instructive confirmation in the experiments by Zagarola [18], performed in pipe flows. Summing up we obtain an expression for the length scale ℓ, using formulae (2.7), (2.8) and (2.12):

$$\ell = \kappa(\mathrm{Re})\gamma z. \tag{2-14}$$

From (2.14) and the first Equation of (2.7) it follows

$$\frac{du}{dz} = \frac{u_*}{\kappa(\mathrm{Re})z}. \tag{2-15}$$

By integration we return to the relation (2.12), the constant of integration cannot be assumed to be a universal one.

3. Shear flow laminarization by suspended heavy particles. Mono-disperse particles size distribution

Consider a horizontal or slightly inclined shear flow in a gravity field loaded by small suspended heavy particles. We assume that both volume and mass concentrations of particles are small. Nevertheless as we will see, the dynamic effect of particles can be large: dust storms, firestorms, and tropical hurricanes give instructive examples. The reason for such large influence of heavy particles is that due to large gravitational force the vortices in turbulent flow have to spend a substantial part of their energy on suspending the particles, and this energy is not returned to the flow when the particles fall down but is dissipated into heat via viscosity. Namely, that is the main cause of a substantial laminarization and acceleration of natural flows. An instructive example: The Martian atmosphere is very subtle; the thickness of the sand layer in absence of a wind is a certain fraction of a millimeter only, but this tiny amount of sand was enough in the year 1972 to create a dust storm that quickly destroyed American and Soviet landing vehicles.

The suspended particles are assumed to be smaller than the internal turbulence length scale (the Kolmogorov scale) below which turbulent vortices begin to be affected by viscosity. Therefore the time of viscous relaxation of the velocity of particles can be considered as negligibly small. This means that it can be assumed that the horizontal components of the *instantaneous* velocity of particles coincide with those of fluid whereas the vertical component of the instantaneous velocity of particles is equal to that of fluid minus a constant quantity: the velocity of the free fall of a particle in an infinite fluid a (the concentration of particles, we remind you, is assumed to be small).

The density of the fluid-particles mixture is equal to $\rho_f(1-s)+\rho_p s=\rho_f(1+\sigma s)$, $\sigma=(\rho_p-\rho_f)/\rho_f$, where ρ_p is the density of particles, ρ_f-the density of fluid, and s is the volume concentration of the particles. In agreement with natural observation it can be assumed also that $\sigma s \ll 1$, so the density of the mixture can be taken equal to the density of the fluid everywhere that the difference of fluid density and density of mixture is not multiplied by the gravity acceleration. Therefore the transverse component of the momentum balance equation

$$-\overline{u'w'}={u_*}^2 \tag{3-1}$$

can be taken identical to the corresponding equation for pure fluid.

The balance of particles leads to a simple equation: the turbulent flux of particles is equal to the amount of falling particles per unit time and unit area:

$$-\overline{s'w'}-as=0 \tag{3-2}$$

(we denote by s the average concentration of particles and by s' its fluctuation).

The difference of the energy balance equation for pure fluid and fluid-particles mixture is the key point. Indeed, the inflow rate of turbulent energy from the mean flow is balanced for the mixture not only by the rate of viscous dissipation into heat, but, in addition by the work of suspension of particles by turbulent vortices which, we repeat, is not returned to the mean flow when the particles fall down. This work (per unit time, unit area and unit mass) is equal to the mean turbulent flux of particles $\overline{s'w'}$ times extra-weight (weight minus to Archimedean force $(\rho_p - \rho_f)g$ per unit volume of particles), divided by the fluid density ρ_f. We obtain for this specific work the expression $\sigma g\overline{s'w'}$, so that the equation of balance of turbulent energy for the fluid-particles mixture takes the form:

$$\overline{u'w'}\frac{du}{dz} + \epsilon + \sigma g\overline{s'w'} = 0. \tag{3-3}$$

We emphasize that the last term of (3.3) is the only term where the concentration enters, and it is significant because it contains a large factor — gravity acceleration g.[1] Equation (3.3) can be rewritten in the form, emphasizing its difference from the corresponding equation for pure fluid (2.2):

$$\overline{u'w'}(1 - \mathrm{Ko})\frac{du}{dz} + \epsilon = 0, \tag{3-4}$$

where the dimensionless parameter

$$\mathrm{Ko} = -(\sigma g\overline{s'w'})/(\overline{u'w'}(du/dz)), \tag{3-5}$$

named *the Kolmogorov parameter (number)* represents the relative part of the turbulent energy influx from the mean flow, spent for the work of suspension of particles by turbulent vortices.

Our further consideration follows the lines of the Kolmogorov-Prandtl analysis of turbulent shear flow.

By analogy with the coefficient of the turbulent momentum exchange k we introduce the coefficient of the particle exchange k_s:

$$k_s = -\overline{s'w'}/(ds/dz). \tag{3-6}$$

As is the case of eddy viscosity k the introduction of k_s, for shear flow, is not a new hypothesis. Assuming, following the Kolmogorov-Prandtl shear flow model, the similarity hypothesis we obtain:

$$k_s = \alpha_s \ell\sqrt{b}. \tag{3-7}$$

[1] As far as is known to the author, the expression for the work of suspension of particles was first obtained by M.A. Velikanov[18]. However, Velikanov deliberately included this work in the equation of the energy balance of the mean flow, not the turbulent energy balance, which cannot be considered as quite correct.

The quantity α_s, which can be called the turbulent Prandtl number for the fluid-particles mixture is a Reynolds number-dependent quantity. Generalizing the considerations of the length scale in the previous section we assume

$$\ell = \kappa(\mathrm{Re})\gamma z \Phi_\ell(\mathrm{Re}, \mathrm{Ko}), \tag{3-8}$$

where the function $\Phi_\ell(\mathrm{Re}, \mathrm{Ko})$ is equal to one for $\mathrm{Ko} = 0$ (pure fluid) and is less than one for $\mathrm{Ko} > 0$,

Using similarity relations (2.6): $k = \ell\sqrt{b}$, $k_s = \alpha_s \ell\sqrt{b}$, $\epsilon = \gamma^4 b^{3/2}/\ell$, we come to a closed system of equations of our model

$$\begin{aligned} &\ell\sqrt{b}\,\frac{du}{dz} = {u_*}^2, \\ &\alpha_s \ell\sqrt{b}\,\frac{ds}{dz} + as = 0, \\ &b^2 = \frac{{u_*}^4}{\gamma^4}(1 - \mathrm{Ko}), \\ &\ell = \kappa(\mathrm{Re})\gamma z \Phi_\ell(\mathrm{Re}, \mathrm{Ko}). \end{aligned} \tag{3-9}$$

The Kolmogorov number can be presented in the following form:

$$\begin{aligned} \mathrm{Ko} &= -\frac{\sigma g \overline{s'w'}}{\overline{u'w'}(du/dz)} = -\frac{\sigma g \alpha_s (ds/dz)}{(du/dz)^2} = -\frac{\sigma g a s}{{u_*}^2 (du/dz)} \\ &= \frac{\sigma g a s \cdot \ell\sqrt{b}}{{u_*}^4} = \frac{\sigma g a^2 s^2}{{u_*}^4 \alpha_s (ds/dz)} = \frac{\omega^2}{dR/dZ}, \end{aligned} \tag{3-10}$$

where

$$R = \frac{1}{s}, \qquad Z = \frac{\alpha_s \sigma g \kappa^2}{{u_*}^2} z, \qquad \omega = \frac{a}{\kappa \alpha_s u_*}. \tag{3-11}$$

The system (3.9) can be reduced to a single equation of first order

$$\frac{1}{\omega^2}\frac{dR}{dZ}\Phi_\ell\Big(\frac{\omega^2}{(dR/dZ)}\Big)\Big(1 - \frac{\omega^2}{(dR/dZ)}\Big)^{1/4} = \frac{R}{\omega Z}. \tag{3-12}$$

We recognize that both κ and Φ_ℓ depend on Reynolds number Re, however we omit this argument in the following formulae.

We emphasize that parameter $\omega = a/\kappa\alpha_s u_*$ plays a basic role in our model. Its physical meaning is transparent: with accuracy up to a constant of the order one it is the ratio of the particle free fall velocity to the characteristic value of the velocity fluctuation.

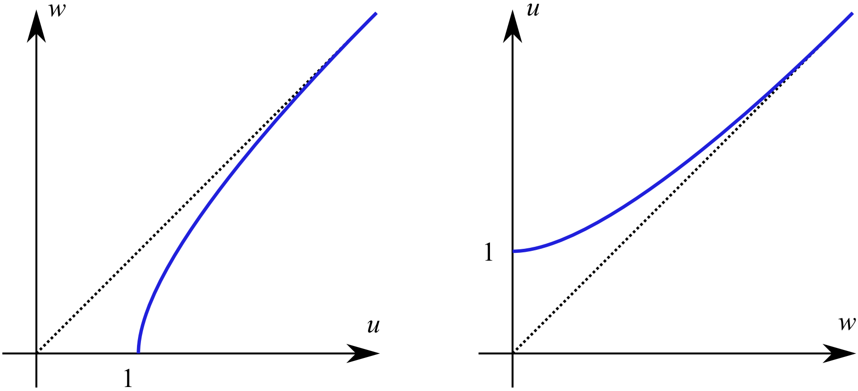


Figure 2. Functions $u(w)$ and $w(u)$ (see the text).

We introduce the function $w(u)$

$$w = u\Phi_\ell\Big(\frac{1}{u}\Big)\Big(1 - \frac{1}{u}\Big)^{1/4}, \tag{3-13}$$

and the function $u(w)$, the inverse to it. They are presented in Figure 2. Both functions at $w, u \to \infty$ have an asymptote $u = w$, represented by the bisectrix of the first quadratures in the u, w plane.

Equation (3.12) can be rewritten in the form

$$\frac{dR}{dZ} = \omega^2 u\Big(\frac{R}{\omega Z}\Big). \tag{3-14}$$

Equation (3.14) is a homogeneous one, and it can be integrated by quadratures. Introducing a new variable $P = R/\omega Z$, so that $dR/dZ = \omega P + \omega Z(dP/dZ)$, and we obtain

$$Z\frac{dP}{dZ} = \omega u(P) - P, \tag{3-15}$$

or, after integration,

$$\ln Z + \text{Const} = \ln\frac{z}{z_0} = \int\limits_{P_0}^{P} \frac{dP}{\omega u(P) - P}. \tag{3-16}$$

Here z_0, P_0 are constants.

The structure of the integral curves of Equation (3.15) in the $P \ln Z$ plane is substantially different in $\omega < 1$ and $\omega > 1$. In the case $\omega < 1$ there exits one and only one root $P = P_*$ of equation $\omega u(P) - P = 0$; this is clear from elementary geometric considerations. There are two classes of integral curves separated by the straight line $P = P_*$. All integral curves approach the asymptotics $P = P_*$ at $\ln(z/z_0) \to \infty$. Returning to the plane s, z, we obtain a picture of integral curves, represented in Figure 3. At $z \to \infty$ all distributions of the concentration of particles tend asymptotically to the curve $s = 1/P_*\omega Z$. The curves of class 1, lying under the separatrix $s = 1/P_*\omega Z$, approach the bottom $z = 0$ asymptotically. The curves of class 2 go to $s = \infty$ at a certain finite value of z. The integral curves of each class can be obtained one from another by shifting along the $\ln Z$ axis. Therefore, the initial height $z = z_0$, where the concentration $s = s_0$ can be prescribed, can be crossed by the integral curves of both classes.

It follows from the previous investigation that at large z the distributions of the concentration of particles, if ω is less than one independently of the boundary condition at a certain level $z = z_0$, are described by the curve $s = 1/P_*\omega Z$. Physically this means that if the velocity fluctuations are sufficiently large, and exceed the free fall velocity a, turbulent flow "takes" as much of the particles as it can, i.e., as much as is allowed by the prescribed shear stress $\tau = \rho {u_*}^2$. Therefore this asymptotic regime is called "the regime of limiting saturation". The regime of limiting saturation corresponds to a constant value of the Kolmogorov number $\mathrm{Ko} = \mathrm{Ko}^*$, which can be obtained from the following equation:

$$\Phi_\ell(\mathrm{Ko}^*)(1 - \mathrm{Ko}^*)^{1/4} = \omega. \tag{3-17}$$

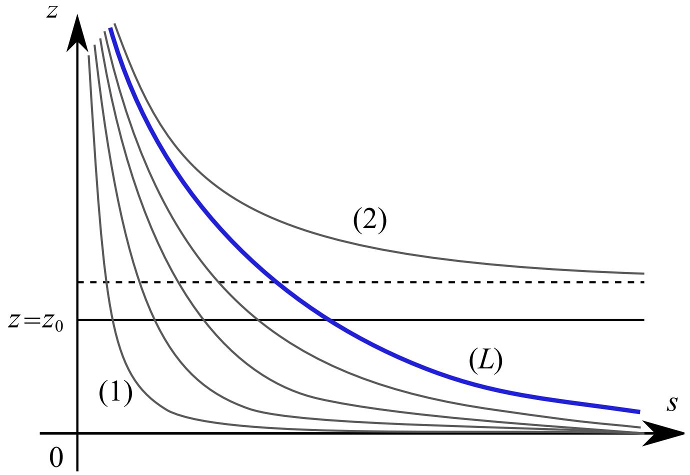


Figure 3. The field of concentration distributions for the case $\omega < 1$. The regime of limiting saturation (Curve L) for which $s \sim \mathrm{Const}/z$ attracts all curves, corresponding to the regimes with various boundary conditions at $z = z_0$. It should be emphasized that these curves have physical meaning only at $s \ll 1$.

Furthermore, using Equation (3.17) we obtain from system (3.9)

$$\frac{du}{dz} = \frac{u_*}{\kappa(\mathrm{Re})\omega z}. \tag{3-18}$$

This means that the velocity gradient at the core of the flow, where the regime of limiting saturation is achieved is $(1/\omega)$ times larger than the velocity gradient in pure fluid flow, given by Equation (2.16). The case when ω is much less than one (very small particles) is of special interest. In this case the Kolmogorov number (in the regime of limiting saturation) is close to one, so nearly the whole inflow of turbulent energy from the mean motion is spent for the suspension of particles. Turbulent energy is strongly reduced. The distribution of concentration in this case takes the form

$$s = \frac{1}{\omega^2 Z} = \frac{\alpha_s {u_*}^4}{a^2 \sigma g z}. \tag{3-19}$$

For the case $\omega > 1$, when the velocity fluctuations are smaller than free fall velocity of particles the situation is different. The denominator of the integrand in (3.16) is positive everywhere. The concentration distributions go to infinity at a certain z, like the curves of the second class in the case $\omega < 1$. Clearly, when s is no longer sufficiently small, these curves make no physical sense. It is important that there is a strong difference between the cases $\omega > 1$ and $\omega < 1$ in the behavior of integral curves, i.e., concentration distributions at large z. It is easy to show that as $z \to \infty$ the distributions behave as $s \sim \mathrm{Const}/z^{\omega}$, and the Kolmogorov number goes to zero as $\mathrm{Const}/z^{\omega-1}$. This means that at large heights the velocity gradient du/dz becomes undisturbed by particles, and is given by relation (2.15). The work of suspension of particles is negligible, and there is no flow acceleration in the core of the flow. The particles create a “lubrication layer”, so that the velocities increase at any height but only due to “lubrication”.

4. Geophysical applications. The modified model

An instructive application of the theoretical construction presented in the previous section is the mathematical modelling of the tropical hurricane. The basis for further consideration will be the “sandwich” model of the hurricane proposed by Sir James Lighthill (see his posthumously published paper [14]). According to this model between the air and the sea there exists an intermediate layer (see Figure 4). Lighthill called it “ocean spray”—which consists of suspended water droplets formed in the process of the breaking of water waves and air. Lighthill proposed to consider ocean spray as a “third fluid”, and strongly emphasized “the need to fill the gap in knowledge about ocean spray at extreme wind speeds”. Lighthill himself concentrated on the thermodynamic side of the modelling. In paper [7] a model, complementary to the Lighthill one was proposed. We emphasize that the possibility of constructing such a model was anticipated by Sir James Lighthill, who discussed it with the authors. However, further analysis showed that a substantially modified model was needed, and it is presented below.

We concentrate here, as in paper [7] mentioned above, on a single effect: flow acceleration in ocean spray by water droplets. We leave aside the effect of the Coriolis force, as well as the cooling effect due to evaporation of droplets and other thermal effects. These effects can be incorporated into the model as was done previously when modelling other atmospheric and oceanic phenomena, see for example [8; 9].

The essence of the modification of the model is as follows. In paper [7] it was naturally assumed as the first step that all the water droplets in ocean spray are identical, and the construction described in the previous section was applied. It was assumed that the water droplets are large, so that the basic parameter ω is larger than one.

The effect of flow acceleration was obtained, but it was less than expected, in spite of the large values of the parameter ω and large concentrations that were assumed in the numerical calculations.

We will demonstrate below that taking into account the availability in ocean spray both of large and small droplets changes the situation. It is difficult to take into consideration the whole spectrum of droplet sizes, in particular, because it is unknown, and it is changing due to various factors, basically unknown. However, it happens to be enough to assume that in ocean spray there are droplets of two sizes, corresponding to the values of parameter $\omega = \omega_1 > 1$ and $\omega = \omega_2 < 1$. Under such a simplified assumption much larger wind accelerations are obtained. The general

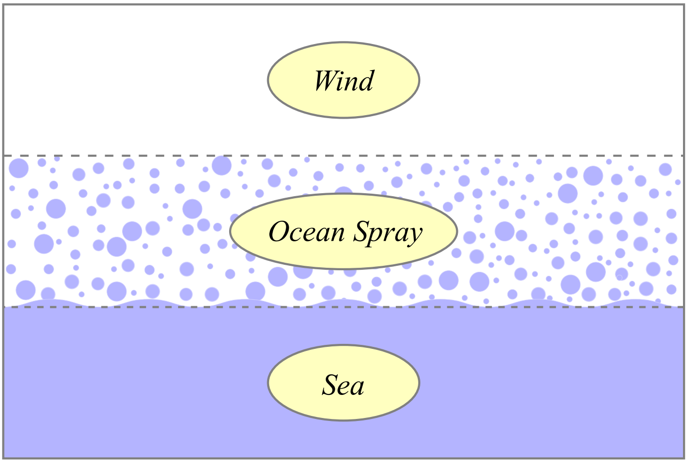


Figure 4. The Lighthill "sandwich model" of a tropical hurricane

consideration of a more realistic case of the continuous spectrum of particle sizes will also be presented below.

As considered previously, we assume that ocean spray occupies the region $z \geq z_0$, where z_0 is the thickness of the layer where the droplets are produced, and the vertical coordinate z is reckoned from the average sea surface. As before we neglect the coalescence of the droplets and the variation of their size due to evaporation. Thus, we assume that two sorts of particles are available in the flow in ocean spray; due to smallness of the concentrations s_1 and s_2 of both kinds of droplets the interference of droplets can be neglected.

The basic system of equations of the modified model is taken in the form suggested by previous analysis, presented in Section 3:

$$\ell\sqrt{b}\,\frac{du}{dz} = {u_*}^2, \tag{4-1}$$

- the momentum balance equation

$$\alpha_s \ell\sqrt{b}\frac{ds_1}{dz} + a_1 s_1 = 0, \tag{4-2}$$

$$\alpha_s \ell\sqrt{b}\frac{ds_2}{dz} + a_2 s_2 = 0, \tag{4-3}$$

- the equations of conservation of both sorts of droplets,

$$b^2 = \frac{u_*^4}{\gamma^4}(1 - \mathrm{Ko}); \tag{4-4}$$

- the equation of turbulent energy balance.

Here the Kolmogorov number $\mathrm{Ko} = \mathrm{Ko}_1 + \mathrm{Ko}_2$, Ko_1 is the Kolmogorov number, corresponding to larger droplets:

$$\mathrm{Ko}_1 = -\frac{\alpha_s \sigma g (ds_1/dz)}{(du/dz)^2}, \tag{4-5}$$

whereas Ko_2 is the Kolmogorov number, corresponding to smaller droplets:

$$\mathrm{Ko}_2 = -\frac{\alpha_s \sigma g (ds_2/dz)}{(du/dz)^2}. \tag{4-6}$$

Thus, a separate balance of droplets of both sizes and their contributions to the work of suspension are considered.

For the length scale the following relation is proposed

$$\ell = \kappa\gamma z \Phi_\ell(\mathrm{Ko}), \tag{4-7}$$

naturally generalizing relation (3.8) for the monodisperse mixture; the Reynolds number dependence of the parameters α_s, κ, γ and the function $\Phi_\ell(\mathrm{Ko})$ are also assumed.

Although system (4.1)–(4.7) seems to be more complicated than the system for the monodisperse mixture, it can also be reduced to a Cauchy problem for an ordinary differential equation of first order due to existence of a first integral. This reduction allows us to perform an asymptotic analysis.

Indeed, we obtain from Equations (4.2), (4.3)

$$\begin{aligned} \frac{ds_1}{ds_2} &= \frac{\omega_1}{\omega_2}\frac{s_1}{s_2}, \\ \omega_1 &= \frac{a_1}{\kappa\alpha_s u_*}, \\ \omega_2 &= \frac{a_2}{\kappa\alpha_s u_*}, \end{aligned} \tag{4-8}$$

and, by integration

$$\frac{s_2}{s_{20}} = \left(\frac{s_1}{s_{10}}\right)^{\omega_2/\omega_1}, \tag{4-9}$$

where s_{10} and s_{20} are the concentrations of both kinds of droplets at $z = z_0$. Also, we obtain, similarly to what was done previously,

$$\begin{aligned} \mathrm{Ko}_1 &= \frac{\sigma g a_1 s_1}{{u_*}^2(du/dz)}, \\ \mathrm{Ko}_2 &= \frac{\sigma g a_2 s_2}{{u_*}^2(du/dz)}. \end{aligned} \tag{4-10}$$

As previously, it is convenient to pass to dimensionless variables

$$U = \frac{\kappa u}{u_*}, \quad Z = \frac{\alpha_s \kappa^2 \sigma g}{{u_*}^2} z, \quad S_1 = \frac{s_1}{s_0}, \quad S_2 = \frac{s_2}{s_0}. \tag{4-11}$$

We assumed here for simplicity $s_{10} = s_{20} = s_0$, and we reduced the system to the form

$$\begin{aligned} \Phi_\ell(\mathrm{Ko})(1-\mathrm{Ko})^{1/4} Z\frac{dU}{dZ} &= 1 \\ \Phi_\ell(\mathrm{Ko})(1-\mathrm{Ko})^{1/4} Z\frac{dS_1}{dZ} + \omega_1 S_1 &= 0 \\ \Phi_\ell(\mathrm{Ko})(1-\mathrm{Ko})^{1/4} Z\frac{dS_2}{dZ} + \omega_2 S_2 &= 0. \end{aligned} \tag{4-12}$$

The boundary conditions we take are of the form

$$S_1 = S_2 = 1, \qquad U = 0 \; at \; Z = Z_0 = \frac{\alpha_s \kappa^2 \sigma g}{u_*^2} z_0. \tag{4-13}$$

Let's estimate the orders of magnitude of all quantities that enter the problem: $z_0 = 10^2 - 10^3$ cm– is the range of the amplitudes of the waves; $\alpha_s\kappa^2$ is of the order of one, $\sigma g \sim 10^6\,\text{cm}/s^2$, u_* is of the order of $10^2\,\text{cm}/s$, therefore Z_0 is in the range $10^4 - 10^6$. Furthermore, s_0 should be in the range of $10^{-6} - 10^{-4}$, so the values of the parameter $A = s_0 Z_0$ can be assumed to be in the range $10^{-1} - 10$.

Introducing the variable $\zeta = \ln(Z/Z_0) = \ln(z/z_0)$ we come to the ultimate system of equations and initial conditions

$$(1-\text{Ko})^{1/4}\Phi_\ell(\text{Ko})\frac{dU}{d\zeta} = 1 \tag{4-14}$$

$$(1-\text{Ko})^{1/4}\Phi_\ell(\text{Ko})\frac{dS_1}{d\zeta} + \omega_1 S_1 = 0 \tag{4-15}$$

$$(1-\text{Ko})^{1/4}\Phi_\ell(\text{Ko})\frac{dS_2}{d\zeta} + \omega_2 S_2 = 0 \tag{4-16}$$

$$\text{Ko} = \frac{Ae^\zeta(\omega_1 S_1 + \omega_2 S_2)}{dU/d\zeta} \tag{4-17}$$

with the boundary conditions $S_1 = S_2 = 1$, $U = 0$ at $\zeta = 0$. The first integral (4.9) takes the form

$$S_2 = S_1^{\omega_2/\omega_1}. \tag{4-18}$$

From system (4.13), (4.14), (4.16), (4.17) a relation for Ko can be obtained:

$$\text{Ko} = Ae^\zeta {\omega_1}^2(1+\theta R_1^{1-\theta})/(dR_1/d\zeta), \tag{4-19}$$

where

$$\begin{aligned} R_1 &= 1/S_1, \\ \theta &= \omega_2/\omega_1. \end{aligned} \tag{4-20}$$

After division by S_1^2 Equation (4.14) can be reduced to the form:

$$(1-\text{Ko})^{1/4}\Phi_\ell(\text{Ko})\frac{dR_1}{d\zeta} - \omega_1 R_1 = 0. \tag{4-21}$$

Finally, dividing by $Ae^\zeta {\omega_1}^2(1+\theta R_1^{1-\theta})$ we obtain

$$\begin{aligned} &\frac{dR_1}{d\zeta}\frac{1}{A\omega_1^2 e^\zeta(1+\theta R_1^{1-\theta})}\left(1-\frac{Ae^\zeta\omega_1^2(1+\theta R_1^{1-\theta})}{dR_1/d\zeta}\right)^{1/4}\Phi_\ell\left(\frac{Ae^\zeta\omega_1^2(1+\theta R_1^{1-\theta})}{dR_1/d\zeta}\right) \\ &\qquad = \frac{R_1}{Ae^\zeta\omega_1(1+\theta R_1^{1-\theta})}. \end{aligned} \tag{4-22}$$

Using the function $u(w)$ introduced by the relation (3.13), we present Equation (4.21) in the form:

$$\frac{dR_1}{d\zeta} = A\omega_1^2 e^{\zeta}(1+\theta R_1^{1-\theta})u\left(\frac{R_1}{Ae^{\zeta}\omega_1(1+\theta R_1^{1-\theta})}\right). \qquad (4\text{-}23)$$

This is an ordinary differential equation of first order, which is to be solved under the initial condition

$$R_1 = 1 \; at \; \zeta = 0. \qquad (4\text{-}24)$$

Under the assumption that $\theta = \omega_2/\omega_1$ is small, the solution to Equation (4.22) can be investigated asymptotically. Indeed, θR_1 is much less than one, i.e., $s_1 \gg \theta s_0$ in a certain interval $0 \le \zeta \le \zeta_*$. In this interval the term $\theta R_1^{1-\theta}$ in (4.22) can be neglected in comparison with 1, and Equation (4.22) takes the form

$$\frac{dR_1}{d\zeta} = A\omega_1^2 e^{\zeta} u\left(\frac{R_1}{Ae^{\zeta}\omega_1}\right). \qquad (4\text{-}25)$$

This equation coincides with Equation (3.12) for $\omega = \omega_1$ (monodisperse flow of large particles). Furthermore, the function $u(w)$ is larger than one, i.e., R_1 is growing faster than $A\omega_1^2 e^{\zeta}$ at all ζ. Therefore there exist a number ζ_{**} where θR_1 becomes much larger than one, and Equation (4.22) takes the form

$$\frac{dR_1}{d\zeta} = A\omega_1^2 e^{\zeta}\theta R_1^{1-\theta} u\left(\frac{R_1^{\theta}}{Ae^{\zeta}\omega_1\theta}\right), \qquad (4\text{-}26)$$

which can be transformed easily using the integral (4.17) to the form

$$\frac{dR_2}{d\zeta} = A\omega_2^2 e^{\zeta} u\left(\frac{R_2}{Ae^{\zeta}\omega_2}\right), \qquad (4\text{-}27)$$

i.e., to Equation (3.12) for $\omega = \omega_2$ (monodisperse flow of small particles).

In the interval $0 \le \zeta < \zeta_*$, according to the investigation of the monodisperse flows, the Kolmogorov number is decreasing; in the interval $\zeta_{**} < \zeta$ it is increasing, reaching the value Ko* satisfying the equation

$$\Phi_{\ell}(\text{Ko}^*)(1-\text{Ko}^*)^{1/4} = \omega_2. \qquad (4\text{-}28)$$

Somewhere in between ζ_* and ζ_{**} a minimum of the Kolmogorov number is reached. Therefore the flow is separated in two regions: the lower region, where the Kolmogorov number is decreasing and reaching a minimum, and the upper region where the Kolmogorov number is growing from the minimum to the final value Ko*. It is natural to consider the lower region as a “lubrication layer” and the upper region as a “suspension layer”. The graphs presented in Figures 5,6, constructed by Dr. C.H. Rycroft on the basis of numerical computations, illustrate a typical structure of the flow in ocean spray if the availability of droplets of two

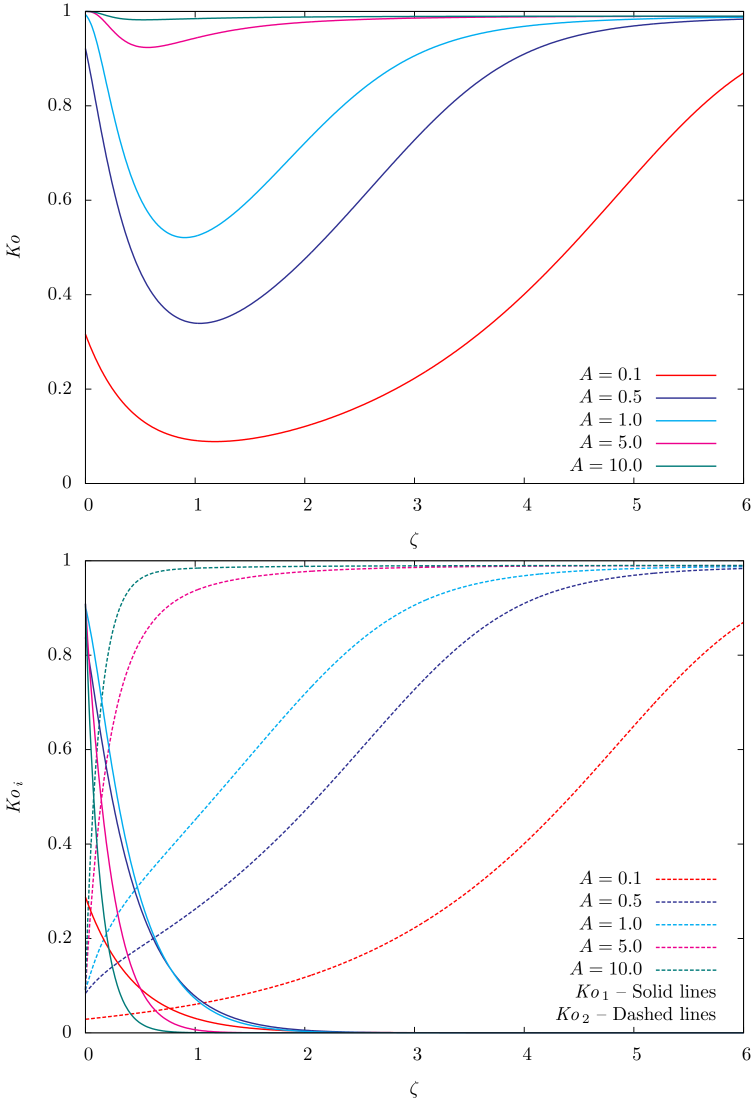


Figure 5. The distribution of the total Kolmogorov number Ko and the Kolmogorov number corresponding to large and small particles Ko_1, Ko_2 for various values of parameter $A = Z_0 s_0$

sizes, large ones ($\omega_1 > 1$) and small ones ($\omega_2 < 1$), is taken into account. In the numerical computations, function $\Phi_\ell(\mathrm{Ko})$ was taken equal to one, and the values of ω of order one were taken in both cases: $\omega_1 = \sqrt{10}$, $\omega_2 = 1/\sqrt{10}$. However, the ratio $\theta = \omega_2/\omega_1 = 1/10$ is a small parameter, allowing an asymptotic analysis. Computations support the results of the asymptotic analysis.

Figure 6 is especially instructive: it demonstrates the strong increase of wind speed in ocean spray in comparison with pure air flow ($S_1 = S_2 = 0$) and also with the flow of fluid-particle mixtures where only large particles are available, $S_2 = 0$.

The analysis presented above can be extended to the case of a continuous spectrum of particle sizes: $\Omega_1 \geq \omega \geq \Omega_2$, where $\Omega_1 > 1$, $\Omega_2 < 1$. Equations (4.13), (4.14) remain valid if ω_1 is a certain reference parameter of value $1 < \omega_1 < \Omega_1$), whereas Equation (4.15) is replaced by the equation of conservation of particles for arbitrary ω in the interval $\Omega_1 \geq \omega \geq \Omega_2$

$$(1-\mathrm{Ko})^{1/4}\Phi_\ell(\mathrm{Ko})\frac{dS}{d\zeta} + \omega S = 0. \tag{4-29}$$

The first integral takes the form $S = S_1^{\omega/\omega_1}$. Here it is assumed that the concentration at the boundary $z = z_0$ of particles in the range between ω and $\omega + d\omega$ is $s_0(\omega)d\omega$. The expression for the Kolmogorov number is given by the following relation:

$$\mathrm{Ko} = \frac{e^{\zeta}\int_{\omega_2}^{\omega_1} A(\omega)S_1^{\omega/\omega_1}d\omega}{dU/d\zeta}, \tag{4-30}$$

where $A(\omega) = s_0(\omega)Z_0$. The previous case of the two-point spectrum corresponds to

$$s_0(\omega) = s_{01}\delta(\omega-\omega_1) + s_{02}\delta(\omega-\omega_2), \tag{4-31}$$

where $\delta(\omega)$ is the Dirac delta function. There is no principal distinction in the results obtained for the case of the continuous spectrum or the two-point spectrum.

5. Conclusion and discussion

The modified model of turbulent shear flow of a suspension of small heavy particles in a fluid is presented. The modification is based on the assumption that in the flow there are two sorts of particles. For the particles of the first sort the velocity of free fall a_1 is larger than the characteristic velocity fluctuation, for the particles of the second sort the velocity of free fall a_2 is less than the characteristic velocity of fluctuation. Considering a_2/a_1 as a small parameter allowed an effective asymptotic analysis of the model equations that were obtained. The investigation was simplified by the existence of a first integral found for the system. The numerical computations are in agreement with the asymptotic analysis.

The main result is that a two-layered flow structure is obtained. In the lower layer, which we called the *lubrication layer*, the Kolmogorov number—the ratio of the work spent on the suspension of particles to the turbulent energy influx from the mean flow—is decreasing. In the upper layer, which we called the *suspension layer*, the Kolmogorov number is increasing after reaching a minimum, until it reaches

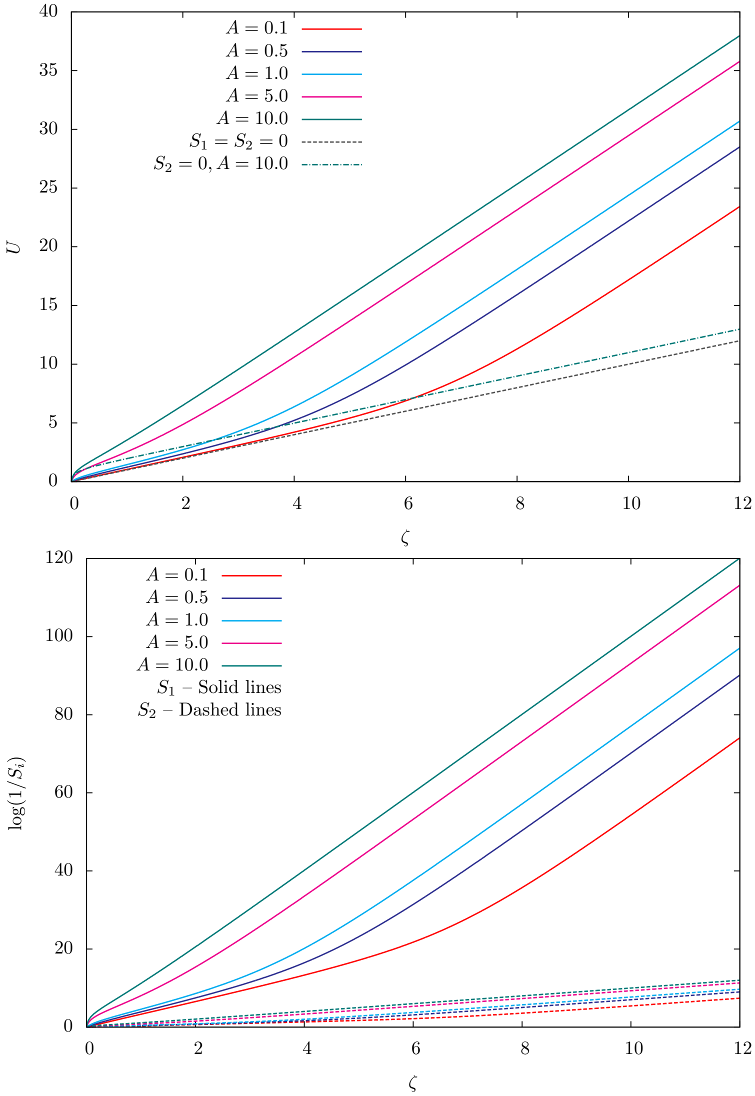


Figure 6. The distributions of dimensionless velocity U and inverse concentrations $1/S_i$ for various values of parameter $A = Z_0 s_0$

the ultimate value at large heights. The basic flow acceleration occurs in the upper layer, where the velocity gradient is small.

Numerical investigation showed that significant laminarization of the flow can be obtained by the addition of heavy particles. What is specifically significant is that the large particles could be of moderate size for reaching high flow speed.

The modified model is applied to the flow in the oceanic spray of a tropical hurricane. It seems that it gives a more realistic structure of the flow than the previously used mono-disperse model.

The modified model can also be applied to dust storms and to big forest and grass fires as well as to other fires when the debris (larger particles) and particles of soot (small particles) are caught by the wind. If the process of combustion is an intensive one so that a sufficiently large amount of small (e.g., soot) particles is produced in the combustion zone, a suspension layer can be formed, and the transition to firestorms—large wind accelerations by intensive fire—can happen, as apparently was the case in the large Chicago fire, 1871. Such firestorms due to intense fires created by large scale bombing (Dresden, February 1945; Hiroshima, August 1945) were also observed.

Acknowledgment

The attention to this work of Professor A.J. Chorin is warmly acknowledged. The friendly help of Dr. C.H. Rycroft who performed the numerical calculations presented in Figures 5, 6 is gratefully appreciated. The comments of the Anonymous Reviewer are also gratefully acknowledged. The author is pleased to express his gratitude to Dr. J. B. Bell for his attention to this article.

References

[1] G. I. Barenblatt. On the motion of suspended particles in a turbulent flow. *Prikl. Mat. Mekh.*, 17(3):261–274, 1953.

[2] G. I. Barenblatt. On the motion of suspended particles in a turbulent flow occupying a half-space or a plane open channel of finite depth. *Prikl. Mat. Mekh.*, 19(1):61–88, 1955.

[3] G. I. Barenblatt. Scaling laws for fully developed turbulent shear flows. Part I. Basic hypotheses and analysis. *J. Fluid Mech.*, 248:513–520, 1993.

[4] G. I. Barenblatt. *Scaling*. Cambridge University press, Cambridge, 2003.

[5] G. I. Barenblatt, A. J. Chorin, and V. M. Prostokishin. Scaling laws in fully developed turbulent pipe flow. *Appl. Mech. Rev.*, 50:413–429, 1997.

[6] G. I. Barenblatt, A. J. Chorin, and V. M. Prostokishin. Self-similar intermediate structures in turbulent boundary layers at large large Reynolds numbers. *J. Fluid Mech.*, 410:263–283, 2000.

[7] G. I. Barenblatt, A. J. Chorin, and V. M. Prostokishin. A note concerning the Lighthill "sandwich model" of tropical cyclones. *Proc. Nat. Ac. Sci.*, 102(32):11148–11150, 2005.

[8] G. I. Barenblatt, N. L. Galerkina, and I. A. Lebedev. Mathematical model of lower quasi-homogeneous oceanic layer: general concepts and scaling-off model. *Izvestiya Bulletin, Russian Ac. Sci., Atmosph. Oceanic Phys.*, 28(1):68–74, 1992.

[9] G. I. Barenblatt, N. L. Galerkina, and I. A. Lebedev. Mathematical model of lower quasi-homogeneous oceanic layer: effects of temperature and salinity stratification and tidal oscillations. *Izvestiya Bulletin, Russian Ac. Sci., Atmosph. Oceanic Phys.*, 29(4):537–542, 1993.

[10] G. I. Barenblatt and G. S. Golitsyn. Local structure of mature dust storms. *J. Atmosph. Sci.*, 31:1917–1933, 1974.

[11] A. J. Chorin. New perspectives in turbulence. *Quart. J. Appl. Math.*, XIV(4):767–785, 1998.

[12] A. N. Kolmogorov. The equations of turbulent motion of incompressible fluids. *Izvestiya, USSR Ac. Sci., Phys.*, 6(1–2):56–58, 1942.

[13] A. N Kolmogorov. On a new version of the gravitational theory of motion of suspended sediment. *Vestnik MGU (Bulletin of Moscow University)*, 3:41–45, 1954.

[14] J. Lighthill. Ocean spray and the thermodynamics of tropical cyclones. *J. Eng. Math.*, 35(1–2):11–42, 1999.

[15] A. S. Monin and A. M. Yaglom. *Statistical fluid mechanics. Mechanics of turbulence*, volume 1. MIT Press, Cambridge, London, 1971.

[16] L. Prandtl. Ueber ein neves Formelsystem für die ausgebildete Turbulenz. *Nachr. Akad. Wiss. Göttingen, Math. Phys. Klasse.*, 6–18, 1945.

[17] M. A. Velikanov. *Dynamics of river-bed flows*. Gidrometeoizdat Press, Leningrad, Moscow, 1946.

[18] M. V. Zagarola. *Mean flow scaling in turbulent pipe flow*. Ph.D. Thesis, Princeton University, 1996.

Received November 19, 2008. Revised October 19, 2009.

GRIGORY ISAAKOVICH BARENBLATT: gibar@math.berkeley.edu
Lawrence Berkeley National Laboratory, 1 Cyclotron Road, Mail Stop 50A/1148,
Berkeley, CA 94720, United States
http://math.lbl.gov/barenblatt/barenblatt.html

COMM. APP. MATH. AND COMP. SCI.
Vol. 4, No. 1, 2009

GLOBAL PATHS OF TIME-PERIODIC SOLUTIONS OF THE BENJAMIN–ONO EQUATION CONNECTING PAIRS OF TRAVELING WAVES

DAVID M. AMBROSE AND JON WILKENING

We classify all bifurcations from traveling waves to nontrivial time-periodic solutions of the Benjamin–Ono equation that are predicted by linearization. We use a spectrally accurate numerical continuation method to study several paths of nontrivial solutions beyond the realm of linear theory. These paths are found to either reconnect with a different traveling wave or to blow up. In the latter case, as the bifurcation parameter approaches a critical value, the amplitude of the initial condition grows without bound and the period approaches zero. We then prove a theorem that gives the mapping from one bifurcation to its counterpart on the other side of the path and exhibits exact formulas for the time-periodic solutions on this path. The Fourier coefficients of these solutions are power sums of a finite number of particle positions whose elementary symmetric functions execute simple orbits (circles or epicycles) in the unit disk of the complex plane. We also find examples of interior bifurcations from these paths of already nontrivial solutions, but we do not attempt to analyze their analytic structure.

1. Introduction

The Benjamin–Ono equation is a nonlocal, nonlinear dispersive equation intended to describe the propagation of internal waves in a deep, stratified fluid [6; 15; 30]. In spite of nonlocality, it is an integrable Hamiltonian system with meromorphic particle solutions [12; 13], N-soliton solutions [24], and N-phase multiperiodic solutions [32; 16; 26]. A bilinear formalism [32] and a Bäcklund transformation [28; 7; 25] have been found to generate special solutions of the equation, and, in the non-periodic setting of rapidly decaying initial conditions, an inverse scattering transform has been developed [18; 20] that exploits an interesting Lax pair structure

MSC2000: 35Q53, 37G15, 37M20, 65K10.

Keywords: periodic solutions, Benjamin–Ono equation, nonlinear waves, solitons, bifurcation, continuation, exact solution, adjoint equation, spectral method.

This work was supported in part by the National Science Foundation through grant DMS-0926378, and by the Director, Office of Science, Computational and Technology Research, U.S. Department of Energy under contract no. DE-AC02-05CH11231.

in which the solution plays the role of a compatibility condition in a Riemann–Hilbert problem.

It is common practice in numerical analysis to test a numerical method using a problem for which exact solutions can be found. Our initial interest in Benjamin–Ono was to serve as such a test problem. Although many of the tools mentioned above can be used to study time-periodic solutions, they do not generalize to problems such as the vortex sheet with surface tension [4; 3] or the true water wave [31; 19], which are not known to be integrable. Our goal in this paper is to develop tools that *will* generalize to these harder problems and use them to study bifurcation and global reconnection in the space of time-periodic solutions of B–O. Specifically, we employ a variant of the numerical continuation method we introduced in [2] for this purpose, which yields solutions that are accurate enough that we are able to recognize their analytic form.

Because we approached the problem from a completely different viewpoint, our description of these exact solutions is very different from previously known representations of multiperiodic solutions. Rather than solve a system of nonlinear algebraic equations at each x to find $u(x,t)$ as was done in [26], we represent $u(x,t)$ in terms of its Fourier coefficients $c_k(t)$, which turn out to be power sums $c_k = 2[\beta_1^k + \cdots + \beta_N^k]$ of a collection of N particles $\beta_j(t)$ evolving in the unit disk of the complex plane as the zeros of a polynomial $z \mapsto P(z,t)$ whose coefficients execute simple orbits (circles or epicycles in $\mathbb{C}$). The connection between the new representation and previous representations will be explored elsewhere [36].

Many of our findings on the structure of bifurcations and reconnections in the manifold of time-periodic solutions of the Benjamin–Ono equation are likely to hold for other systems as well. One interesting pitfall we have identified by applying our method to an integrable problem is that degenerate bifurcations can exist that are not predicted by counting linearly independent, periodic solutions of the linearization about traveling waves. Although it is possible that such degeneracy is a consequence of the symmetries that make this problem integrable, it is also possible that other problems such as the water wave will also possess degenerate bifurcations that are invisible to a linearized analysis. We have also found that one cannot achieve a complete understanding of these manifolds of time-periodic solutions by holding, for example, the mean constant and varying only one parameter. In some of the simulations where we hold the mean fixed, the solution (that is, the L^2 norm of the initial condition) blows up as the parameter approaches a critical value rather than reconnecting with another traveling wave. However, if the mean is simultaneously varied, it is always possible to reconnect. Thus, although numerical continuation with more than one parameter is difficult, it will likely be necessary to explore multidimensional parameter spaces to achieve a thorough understanding of time-periodic solutions of other problems.

On the numerical side, we believe our use of certain Fourier modes of the initial conditions as bifurcation parameters will prove useful in many other problems beyond Benjamin–Ono. We also wish to advocate the use of variational calculus and optimal control for the purpose of finding time-periodic solutions (or solving other two-point boundary value problems). For ODE, a competing method known as orthogonal collocation (for example, as implemented in AUTO [17]) has proved to be a very powerful technique for solving boundary value problems. This approach becomes quite expensive when the dimension of the system increases, and is therefore less competitive for PDE than it is for ODE. For PDE, many authors do not attempt to find exact periodic solutions, and instead point out that typical solutions of certain equations do tend to pass near their initial states at a later time [11]. If true periodic solutions are sought, a more common approach has been to either iterate on a Poincaré map and use stability of the orbit to find time-periodic solutions [10], or use a shooting method [33; 35] to find a fixed point of the Poincaré map.

In a shooting method, we define a functional $F(u_0, T) = [u(\cdot, T) - u_0]$ that maps initial conditions and a supposed period to the deviation from periodicity. The equation $F = 0$ is then solved by Newton's method, where the Jacobian $J = DF$ is either computed using finite differences [34] or by solving the variational equation repeatedly to compute each column of J. We have found that it is much more efficient (by a factor of the number of columns of J) to instead minimize the scalar functional $G = \frac{1}{2}\|F\|^2$ via a quasi-Newton method in which the gradient DG is computed by solving an adjoint PDE.

Bristeau et. al. [8] developed a similar approach for linear (but two- or three-dimensional) scattering problems. Three-dimensional problems are intractable by the standard shooting approach as J could easily have 10^5 columns. However, the gradient of G can be computed by solving a single adjoint PDE. The success of the method then boils down to a question of the number of iterations required for the minimization algorithm to converge. For linear problems, Bristeau et. al. have had success using conjugate gradients to minimize G. We find that BFGS [9] works very well for nonlinear problems like the Benjamin–Ono equation and the vortex sheet with surface tension [3].

To find nontrivial time-periodic solutions in the present work, we use a symmetric variant of the algorithm described in [2]. Although the original method works well, we use the symmetric variant for the simulations in this paper because evolving to $T/2$ requires half the time-steps and yields more accurate answers (as there is less time for numerical round-off error to corrupt the calculation). Moreover, the number of degrees of freedom in the search space of initial conditions is also cut in half and the condition number of the problem improves when we eliminate phase shift degrees of freedom via symmetry rather than including them in the penalty function described in Section 3.1. Although we do not make use of it, there is a procedure known as the Meyer–Marsden–Weinstein reduction [27; 23] that allows one to reduce the dimension of a symplectic manifold on which a group acts symplectically. This allows one to eliminate actions of the group (for example, translations) from the phase space. Equilibria and periodic solutions of the reduced Hamiltonian system correspond to (families of) relative equilibria and relative periodic solutions [39] of the original system.

This paper is organized as follows. In Section 2, we discuss stationary, traveling and particle solutions of B-O, linearize about traveling waves, and classify all bifurcations predicted by linear theory from traveling waves to nontrivial time-periodic solutions. Some of the more technical material from this section is given in Appendix A. In Section 3, we present a collection of numerical experiments using our continuation method to follow several paths of nontrivial solutions beyond the realm of linear theory in order to formulate a theorem that gives the global mapping from one traveling wave bifurcation to its counterpart on the other side of the path. In Section 4, we study the behavior of the Fourier modes of the time-periodic solutions found in Section 3 and state a theorem about the exact form of these solutions, which is proved in Appendix B. Finally, in Section 5, we discuss interior bifurcations from these paths of already nontrivial solutions to still more complicated solutions. Although the existence of such a hierarchy of solutions was already known [32], bifurcation between various levels of the hierarchy has not previously been discussed.

2. Bifurcation from traveling waves

In this section, we study the linearization of the Benjamin–Ono equation about stationary solutions and traveling waves by solving an infinite dimensional eigenvalue problem in closed form. Each eigenvector corresponds to a time-periodic solution of the linearized equation. The traveling case is reduced to the stationary case by requiring that the period of the perturbation (with a suitable spatial phase shift) coincide with the period of the traveling wave. The main goal of this section is to devise a classification scheme of the bifurcations from traveling waves so that in later sections we can describe which (local) bifurcations are connected together by a global path of nontrivial time-periodic solutions.

2.1. ***Stationary, traveling and particle solutions.*** We consider the Benjamin–Ono equation on the periodic interval $\mathbb{R}/2\pi\mathbb{Z}$, namely,

$$u_t = Hu_{xx} - uu_x. \tag{1}$$

Here H is the Hilbert transform, which has the symbol $\hat{H}(k) = -i\,\mathrm{sgn}(k)$. The Benjamin–Ono equation possesses solutions [12; 2] of the form

$$u(x,t) = \alpha_0 + \sum_{l=1}^{N} \phi(x; \beta_l(t)), \tag{2}$$

where α_0 is the mean, $\beta_1(t), \ldots, \beta_N(t)$ are the trajectories of N particles evolving in the unit disk Δ of the complex plane and governed by the ODE

$$\dot{\beta}_l = \sum_{\substack{m=1\\ m\neq l}}^{N} \frac{-2i\beta_l^2}{\beta_l - \beta_m} + \sum_{m=1}^{N} \frac{2i\beta_l^2}{\beta_l - \bar{\beta}_m^{-1}} + i(2N-1-\alpha_0)\beta_l \quad (1 \le l \le N), \tag{3}$$

and $\phi(x; \beta)$ is the function with Fourier representation

$$\hat{\phi}(k;\beta) = \left\{ \begin{array}{ll} 0, & k=0 \\ 2\beta^k, & k>0 \\ 2\bar{\beta}^{|k|}, & k<0 \end{array} \right\}, \qquad \beta \in \Delta = \{z : |z| < 1\}. \tag{4}$$

The function $\phi(x; \beta)$ has a peak centered at $x = \arg(\bar{\beta})$ with amplitude growing to infinity as $|\beta|$ approaches 1. The N-hump traveling waves (with a spatial period of $2\pi/N$) are a special case of the particle solutions given by (2) and (3):

$$\begin{gathered} u_{\text{trav}}(x,t;\alpha_0,N,\beta) = \alpha_0 + \sum_{l=1}^{N} \phi(x;\beta_l(t)), \quad \beta_l(t) = \sqrt[N]{\beta} e^{-ict}, \\ c = \alpha_0 - N\alpha(\beta). \end{gathered} \tag{5}$$

Each β_l is assigned a distinct N-th root of β and $\alpha(\beta)$ is the mean of the one-hump stationary solution, namely,

$$\alpha(\beta) = \frac{1-3|\beta|^2}{1-|\beta|^2}, \quad |\beta|^2 = \frac{1-\alpha(\beta)}{3-\alpha(\beta)}. \tag{6}$$

The solution (5) moves to the right when $c > 0$. Indeed, it may also be written

$$u_{\text{trav}}(x,t;\alpha_0,N,\beta) = u_{\text{stat}}(x-ct;N,\beta) + c, \tag{7}$$

where u_{stat} is the N-hump stationary solution

$$u_{\text{stat}}(x;N,\beta) = N\alpha(\beta) + \sum_{\{\gamma\,:\,\gamma^N=\beta\}} \phi(x;\gamma) \; = \; N\alpha(\beta) + N\phi(Nx;\beta). \tag{8}$$

The Fourier representation of u_{stat} is

$$\hat{u}_{\text{stat}}(k;N,\beta) = \begin{cases} N\alpha(\beta), & k=0, \\ 2N\beta^{k/N}, & k \in N\mathbb{Z},\ k>0, \\ 2N\bar{\beta}^{|k|/N}, & k \in N\mathbb{Z},\ k<0, \\ 0, & \text{otherwise.} \end{cases} \tag{9}$$

Amick and Toland have shown [5] that all traveling waves of the Benjamin–Ono equation have the form (7); see also [36].

2.2. ***Linearization about stationary solutions.*** Let $u(x) = u_{\text{stat}}(x; N, \beta)$ be an N-hump stationary solution. In [2], we solved the linearization of (1) about u, namely,

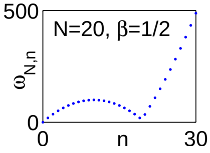
$$v_t = Hv_{xx} - (uv)_x = iBAv, \quad A = H\partial_x - u, \quad B = \frac{1}{i}\partial_x, \tag{10}$$

by substituting the expression $v(x,t) = \text{Re}\{Cz(x)e^{i\omega t}\}$ into (10) and solving the eigenvalue problem

$$BAz = \omega z \tag{11}$$

in closed form. Specifically, we showed that the eigenvalues $\omega_{N,n}$ are given by

$$\omega_{N,n} = \left\{ \begin{array}{ll} -\omega_{N,-n}, & n < 0 \\ 0, & n = 0 \\ (n)(N-n), & 1 \le n \le N-1 \\ (n+1-N)\big(n+1+N(1-\alpha(\beta))\big), & n \ge N \end{array} \right\} \tag{12}$$



The zero eigenvalue $\omega_{N,0} = 0$ has geometric multiplicity two and algebraic multiplicity three. The eigenfunctions in the kernel of BA are

$$z_{N,0}^{(1,0)}(x) = -\frac{\partial}{\partial x}u_{\text{stat}}(x; N, \beta), \quad z_{N,0}^{(2)}(x) = \frac{\partial}{\partial|\beta|}u_{\text{stat}}(x; N, \beta), \tag{13}$$

which correspond to changing the phase or amplitude of β in the underlying stationary solution. There is also a Jordan chain [37] of length two associated with $z_{N,0}^{(1,0)}(x)$, namely,

$$z_{N,0}^{(1,1)}(x) = 1, \quad \big(iBAz_{N,0}^{(1,1)} = z_{N,0}^{(1,0)}\big), \tag{14}$$

which corresponds to the fact that adding a constant to a stationary solution causes it to travel. The fact that all the eigenvalues $i\omega_{N,n}$ in the linearization (10) are purely imaginary is a consequence of the Hamiltonian structure [13] of the Benjamin–Ono equation. For non-Hamiltonian systems, one does not generally expect to find time-periodic perturbations of traveling waves (as periodic solutions of the linearized problem may not even exist).

The eigenfunctions $z_{N,n}(x)$ corresponding to positive eigenvalues $\omega_{N,n}$ (with $n \ge 1$) have the Fourier representation

$$\begin{aligned}
&\hat{z}_{N,n}(k)\Big|_{k=n+jN} = \left\{ \begin{array}{ll} \left(1+\dfrac{N(|j|-1)}{N-n}\right)\bar{\beta}^{|j|-1}, & j<0 \\ C\left(1+\dfrac{Nj}{n}\right)\beta^{j+1}, & j\geq 0 \end{array} \right\} \\
&\qquad\qquad \left(1\leq n\leq N-1,\quad C=\frac{-nN}{(N-n)\big[n+(N-n)|\beta|^2\big]}\right), \\
&\hat{z}_{N,n}(k)\Big|_{k=n+1-N+jN} = \left\{ \begin{array}{ll} 0, & j<0 \\ \dfrac{-\bar{\beta}}{(1-|\beta|^2)^2}\left[1-\left(1-\dfrac{N}{n+1}\right)|\beta|^2\right], & j=0 \\ \left(1+\dfrac{N(j-1)}{n+1}\right)\beta^{j-1}, & j>0 \end{array} \right\} \quad (n\geq N),
\end{aligned} \tag{15}$$

with all other Fourier coefficients equal to zero. The eigenfunctions corresponding to negative eigenvalues $\omega_{N,n}$ (with $n\leq -1$) satisfy $z_{N,n}(x)=\overline{z_{N,-n}(x)}$, so the Fourier coefficients appear in reverse order, conjugated. For $1\leq n\leq N-1$, any linear combination of $z_{N,n}(x)$ and $z_{N,N-n}(x)$ is also an eigenfunction; however, the choices here seem most natural as they simultaneously diagonalize the shift operator (discussed below) and yield directions along which nontrivial solutions exist beyond the linearization. Said differently, we have listed the first $N-1$ positive eigenvalues $\omega_{N,n}$ in an unusual order (rather than enumerating them monotonically and coalescing multiple eigenvalues) because this is the order that leads to the simplest description of the global paths of nontrivial solutions connecting these traveling waves.

2.3. ***Classification of bifurcations from traveling waves.*** Time-periodic solutions of the Benjamin–Ono equation with period T have initial conditions that satisfy $F(u_0,T)=0$, where $F: H^1\times\mathbb{R}\to H^1$ is given by

$$F(u_0,T)=u(\,\cdot\,,T)-u_0, \quad u_t=Hu_{xx}-uu_x, \quad u(\,\cdot\,,0)=u_0. \tag{16}$$

First, we linearize F about an N-hump stationary solution $u_0(x)=u_{\text{stat}}(x;\,N,\beta)$. The Fréchet derivative $DF=(D_1F,D_2F): H^1\times\mathbb{R}\to H^1$ yields directional derivatives

$$\begin{aligned}
D_1F(u_0,T)v_0 &= \frac{\partial}{\partial\varepsilon}\Big|_{\varepsilon=0} F(u_0+\varepsilon v_0,T) = v(\,\cdot\,,T)-v_0=\big[e^{iBAT}-I\big]\,v_0, \\
D_2F(u_0,T)\tau &= \frac{\partial}{\partial\varepsilon}\Big|_{\varepsilon=0} F(u_0,T+\varepsilon\tau)=0.
\end{aligned} \tag{17}$$

Note that $v_0\in\ker D_1F(u,T)$ if and only if the solution $v(x,t)$ of the linearized problem is periodic with period T. As a result, a basis for the kernel $\mathcal{N}=\ker DF(u_0,T)$

consists of $(0; 1)$ together with all pairs $(v_0; 0)$ of the form

$$v_0(x) = \mathrm{Re}\{z_{N,n}(x)\} \quad \text{or} \quad v_0(x) = \mathrm{Im}\{z_{N,n}(x)\}, \tag{18}$$

where n ranges over all integers such that

$$\omega_{N,n} T \in 2\pi\mathbb{Z}, \tag{19}$$

with N and β (in the formula (12) for $\omega_{N,n}$) held fixed. The corresponding periodic solutions of the linearized problem are

$$v(x,t) = \mathrm{Re}\{z_{N,n}(x)e^{i\omega_{N,n}t}\} \quad \text{or} \quad v(x,t) = \mathrm{Im}\{z_{N,n}(x)e^{i\omega_{N,n}t}\}. \tag{20}$$

Negative values of n have already been accounted for in (18) and (20) using $z_{N,-n}(x) = \overline{z_{N,n}(x)}$, and the $n = 0$ case always yields two vectors in the kernel, namely, those in (13). These directions do not cause bifurcations as they lead to other stationary solutions.

Next we wish to linearize F about an arbitrary traveling wave. Suppose $u(x) = u_{\mathrm{stat}}(x; N, \beta)$ is an N-hump stationary solution and $U(x,t) = u(x-ct)+c$ is a traveling wave. Then the solutions v and V of the linearizations about u and U, respectively, satisfy $V(x,t) = v(x-ct, t)$. Note also that

$$F(U_0, T) = 0 \quad \text{if and only if} \quad cT = \frac{2\pi\nu}{N} \quad \text{for some } \nu \in \mathbb{Z}, \tag{21}$$

where $U_0(x) = U(x,0) = u(x)+c$. Note that ν is the number of times the traveling wave turns over itself in one period. Assuming (21) holds, we set $\theta = 2\pi\nu/N$ and compute

$$\begin{aligned} [D_1F(U_0,T)v_0](x) &= v(x-cT,T) - v_0(x) = [(S_\theta e^{iBAT} - I)v_0](x), \\ [D_2F(U_0,T)\tau](x) &= U_t(x,T)\tau = -cu_x(x-cT)\tau = -cu_x(x)\tau, \end{aligned} \tag{22}$$

where v solves (10) and the shift operator S_θ is defined via

$$S_\theta z(x) = z(x-\theta), \quad \hat{S}_{\theta,kl} = e^{-ik\theta}\delta_{kl}. \tag{23}$$

One element of $\mathcal{N} = \ker DF(U_0,T)$ arises from (14), which gives

$$e^{iBAt}1 = 1 - tu_x \quad \Rightarrow \quad D_1F(U_0,T)(-c/T) + D_2F(U_0,T)1 = 0,$$

and implies $(-c/T; 1) \in \mathcal{N}$. This just means that we can change the period T by a small amount τ by adding the constant $-(c/T)\tau$ to U_0 (this also follows from the condition (21) that $cT = \theta = \text{const}$). If we wish to change the period without changing the mean, we need to simultaneously adjust $|\beta|$ in the underlying stationary solution $u(x) = u_{\mathrm{stat}}(x; N, \beta)$. The other elements of $\mathcal{N}$ are of the form $(v_0; 0)$ with

$$v_0(x) = \mathrm{Re}\{z_{N,n}(x)\} \quad \text{or} \quad v_0(x) = \mathrm{Im}\{z_{N,n}(x)\}. \tag{24}$$

The admissible values of n here are found using (22) together with

$$S_\theta e^{iBAT} z_{N,n} = e^{i(\omega_{N,n}T - \theta k_{N,n})} z_{N,n}, \qquad \theta = \frac{2\pi\nu}{N}, \tag{25}$$

where $k_{N,n}$ is the stride offset of the non-zero Fourier coefficients of $z_{N,n}$, i.e.,

$$\hat{z}_{N,n}(k) \neq 0 \quad \Rightarrow \quad k - k_{N,n} \in N\mathbb{Z}. \tag{26}$$

Thus, instead of (19), n ranges over all integers such that

$$\omega_{N,n}T \in 2\pi\left(\frac{\nu k_{N,n}}{N} + \mathbb{Z}\right), \quad k_{N,n} = \begin{cases} -k_{N,-n}, & n < 0, \\ 0, & n = 0, \\ n, & 1 \le n \le N-1, \\ \operatorname{mod}(n+1, N), & n \ge N. \end{cases} \tag{27}$$

As before, negative values of n need not be considered once we take real and imaginary parts in (24), and the $n = 0$ case always gives the two vectors $(z_{N,0}^{(1,0)}; 0)$ and $(z_{N,0}^{(2)}; 0)$ in $\mathcal{N}$, which lead to other traveling waves rather than bifurcations to nontrivial solutions.

Our numerical experiments have led us to the following conjecture, which we prove as part of Theorem 3 in Section 4:

Conjecture 1. For every $\beta \in \Delta$ and $(N, \nu, n, m) \in \mathbb{Z}^4$ satisfying

$$N \ge 1, \quad \nu \in \mathbb{Z}, \quad n \ge 1, \quad m \ge 1, \quad m \in \nu k_{N,n} + N\mathbb{Z}, \tag{28}$$

there is a four parameter sheet of nontrivial time-periodic solutions bifurcating from the N-hump traveling wave with speed index ν, $(cT = 2\pi\nu/N)$, bifurcation index n, and oscillation index m, $(\omega_{N,n}T = 2\pi m/N)$. The phase and amplitude of the traveling wave are determined by β.

The main content of this conjecture is that we do not have to consider linear combinations of the $z_{N,n}$ with different values of n to find periodic solutions of the nonlinear problem — this basis is already "diagonal" with respect to these bifurcations. This is true in spite of a small divisor problem preventing $DF(U_0, T)$ from being Fredholm. The decision to number the first $N - 1$ eigenvalues $\omega_{N,n}$ nonmonotonically in (12) and to simultaneously diagonalize the shift operator S_θ when choosing eigenvectors $z_{N,n}$ in (15) was essential to make this work. Formulas relating the period, T, the mean, α_0, and the decay parameter, $|\beta|$, for each of these bifurcations are given in Appendix A along with a list of bifurcation rules governing "legal" values of the mean.

A canonical way to generate one of these bifurcations is to take β real and perturb the initial condition in the direction $v_0(x) = \operatorname{Re}\{z_{N,n}(x)\}$. This leads to nontrivial solutions with even symmetry at $t = 0$. Perturbation in the $\operatorname{Im}\{z_{N,n}(x)\}$ direction yields the same set of nontrivial solutions, but with a spatial and temporal phase shift:

$$\operatorname{Im}\{z_{N,n}(x-ct)e^{i\omega t}\} = \operatorname{Re}\left\{z_{N,n}\left(\left(x-\frac{c\pi}{2\omega}\right)-c\left(t-\frac{\pi}{2\omega}\right)\right)e^{i\omega(t-(\pi/2\omega))}\right\}, \quad (29)$$

where $\omega = \omega_{N,n}$. The manifold of nontrivial solutions is four dimensional with two essential parameters (for example, the mean α_0 and a parameter governing the distance from the traveling wave) and two inessential parameters (the spatial and temporal phase). In our numerical studies, we use the real part of a Fourier coefficient c_k of the initial condition (with k such that $\hat{z}_{N,n}(k) \neq 0$) for the second essential bifurcation parameter. When we discuss exact solutions in Section 4, a different parameter will be used.

We remark that this enumeration of bifurcations accounts for all time-periodic solutions of the linearization about traveling waves; therefore, the heuristic that each bifurcation of the nonlinear problem gives rise to a linearly independent vector in the kernel $\mathcal{N}$ of the linearized problem suggests that we have found all bifurcations from traveling waves. Interestingly, this turns out not to be the case; the interior bifurcations we discuss in Section 5 can occur at the endpoints of the path, allowing for degenerate bifurcations directly from traveling waves to higher levels in the infinite hierarchy of time-periodic solutions. Only the transition from the first level of the hierarchy to the second is "visible" to a linearized analysis about traveling waves. The other transitions become linearly dependent on these in the limit as the traveling wave is approached; they will be analyzed in [36].

3. Numerical experiments

In this section we present a collection of numerical experiments in which we start with a given bifurcation (N, ν, n, m, β) and use a symmetric variant of the method we described in [2] for finding periodic solutions of nonlinear PDE to continue these solutions until another traveling wave is found, or until the solution blows up as the bifurcation parameter approaches a critical value. We determine the bifurcation indices (N', ν', n', m') at the other end of the path of nontrivial solutions by fitting the data to the formulas of the previous section. By trial and error, we are then able to guess a formula relating (N', ν', n', m') to (N, ν, n, m) that we use in Section 4 to construct exact solutions.

3.1. ***Numerical method.*** As mentioned in Section 2.3, a natural choice of spatial and temporal phase can be achieved by choosing the parameter β of the traveling wave to be real and perturbing the initial condition in the direction $v_0(x) = \operatorname{Re}\{z_{N,n}(x)\}$. For reasons of efficiency and accuracy (explained in the introduction), we now restrict our search for time-periodic solutions of (1) to functions $u(x,t)$ that possess even spatial symmetry at $t = 0$. If we succeed in finding solutions with this symmetry, then they — together with their phase-shifted counterparts analogous to (29) — span the nullspace $\mathcal{N} = \ker DF(U_0, T)$ in the limit that the perturbation goes to zero. Thus, we do not expect symmetry breaking bifurcations from traveling waves that cannot be phase shifted to have even symmetry at $t = 0$.

The Benjamin–Ono equation has the property that if $u(x,t)$ is a solution of (1), then so is $U(x,t) = u(-x,-t)$. As a result, if u is a solution such that $u(x, T/2) = U(x, -T/2)$, then $u(x,T) = U(x,0)$, i.e., u is time-periodic if the initial condition has even symmetry. Thus, we seek initial conditions u_0 with even symmetry and a period T to minimize the functional

$$G_{\text{tot}}(u_0, T) = G(u_0, T) + G_{\text{penalty}}(u_0, T), \tag{30}$$

where

$$G(u_0, T) = \frac{1}{2}\int_0^{2\pi} [u(x, T/2) - u(2\pi - x, T/2)]^2 \, dx, \tag{31}$$

and $G_{\text{penalty}}(u_0, T)$ is a non-negative penalty function to impose the mean and set the bifurcation parameter. To compute the gradient of G with respect to variation of the initial conditions, we use

$$\frac{d}{d\varepsilon}\bigg|_{\varepsilon=0} G(u_0 + \varepsilon v_0, T) = \int_0^{2\pi} \frac{\delta G}{\delta u_0}(x) v_0(x)\, dx, \tag{32}$$

where the variational derivative

$$\frac{\delta G}{\delta u_0}(x) = 2w(x, T/2), \qquad w_0(x) = u(x, T/2) - u(2\pi - x, T/2) \tag{33}$$

is found by solving the following adjoint equation from $s = 0$ to $s = T/2$:

$$w_s(x,s) = -Hw_{xx}(x,s) + u(x, T/2 - s) w_x(x,s), \qquad w(\,\cdot\,, 0) = w_0. \tag{34}$$

Since v_0 is assumed symmetric in this formulation, (33) is equivalent to

$$\frac{\delta G}{\delta u_0}(x) = w(x, T/2) + w(2\pi - x, T/2). \tag{35}$$

The Benjamin–Ono and adjoint equations are solved using a pseudo-spectral collocation method employing a fourth order semi-implicit additive Runge–Kutta method [14; 21; 38] to advance the solution in time. The BFGS method [9; 29] is then used to minimize G_{tot} (varying the period and the Fourier coefficients of the initial conditions). We use the penalty function

$$G_{\text{penalty}}(u_0, T) = 1/2([a_0(0) - \alpha_0]^2 + [a_K(0) - \rho]^2) \tag{36}$$

to specify the mean α_0 and the real part ρ of the K-th Fourier coefficient of the initial condition

$$u_0(x) = \sum_{k=-M/2+1}^{M/2} c_k(0)e^{ikx}, \quad c_k(t) = a_k(t) + ib_k(t). \tag{37}$$

The parameters α_0 and ρ serve as the bifurcation parameters while the phases are determined by requiring that the solution have even symmetry at $t = 0$. We generally choose K to be the first $k \geq 1$ such that $\hat{z}_{N,n}(k) \neq 0$.

Our continuation method consists of three stages. First, we choose a traveling wave and a set of bifurcation indices to begin the path of nontrivial solutions. We also choose a direction in which to vary the bifurcation parameter ρ and the mean α_0. In most of our numerical experiments, we hold α_0 fixed; however, in the example of Figure 6 below, we vary ρ and α_0 simultaneously. The traveling wave serves as the zeroth point on the path. The initial guess for the first point on the path is obtained by perturbing the initial condition of the traveling wave in the direction $\text{Re}\{z_{N,n}(x)\}$. We use the period T given in (A.1) in Appendix A as a starting guess. We then use the minimization algorithm to descend from the starting guess predicted by linear theory to an actual time-periodic solution. The second stage of the continuation algorithm consists of varying ρ (and possibly α_0), using linear extrapolation for the starting guess (for u_0 and T) of the next solution, and then minimizing G_{tot} to find an actual time-periodic solution with these values of ρ and α_0. If the initial value of G_{tot} from the extrapolation step is too large, we discard the step and try again with a smaller change in ρ and α_0. The final stage of the algorithm consists of identifying the reconnection on the other side of the path. We do this by blindly overshooting the target values of ρ and α_0 (which we do not know in advance). Invariably, the algorithm will lock onto a family of traveling waves once we reach the end of the path of nontrivial solutions. We look at the Fourier coefficients of the last nontrivial solution before the traveling waves are reached and match them with the formulas for $\hat{z}_{N',n'}(k)$ to determine the correct bifurcation indices on this side of the path. (A prime indicates indices for the bifurcation at the other end of the path.) We then recompute the last several solutions on the path of nontrivial solutions with appropriate values of ρ and α_0 to arrive exactly at the traveling wave on the last iteration. We sometimes change K in (36) to compute this reconnection to avoid $\hat{z}_{N',n'}(K) = 0$.

The running time of our algorithm (on a 2.4 GHz desktop machine) varies from a few hours to compute one of the paths labeled *a*–*l* in (38)–(41) below, to a few days to compute a path in which the solution blows up, such as the one shown in Figure 5 (page 196). We always refine the mesh and timestep enough so that the solutions are essentially exact (with $G_{\text{tot}} \leq 10^{-26}$ in the easy cases and 10^{-20} in the hard cases).

3.2. ***Global paths of nontrivial solutions.*** We now investigate the global behavior of nontrivial solutions that bifurcate from arbitrary stationary or traveling waves. We find that these nontrivial solutions act as rungs in a ladder, connecting stationary and traveling solutions with different speeds and wavelengths by creating or annihilating oscillatory humps that grow or shrink in amplitude until they become part of the stationary or traveling wave on the other side of the rung. In some cases, rather than reconnecting with another traveling wave, the solution blows up (the L^2 norm of the initial condition grows without bound) as the bifurcation parameter ρ approaches a critical value. However, even in these cases a reconnection with another traveling wave does occur if, in addition to ρ, we vary the mean, α_0, appropriately.

Recall from Section 2.3 that we can enumerate all such bifurcations by specifying a complex parameter β in the unit disk Δ along with four integers (N, ν, n, m) satisfying (28), and in most cases we can solve for $|\beta|$ in terms of the mean, α_0, using (A.4) in Appendix A. In [2], we presented a detailed study of the solutions on the path connecting a one-hump stationary solution to a two-hump traveling wave moving left. We denote this path by

$$a: \quad (1, 0, 1, 1) \quad \longleftrightarrow \quad (2, -1, 1, 1), \tag{38}$$

where the label a refers to the bifurcation diagram in Figure 1.

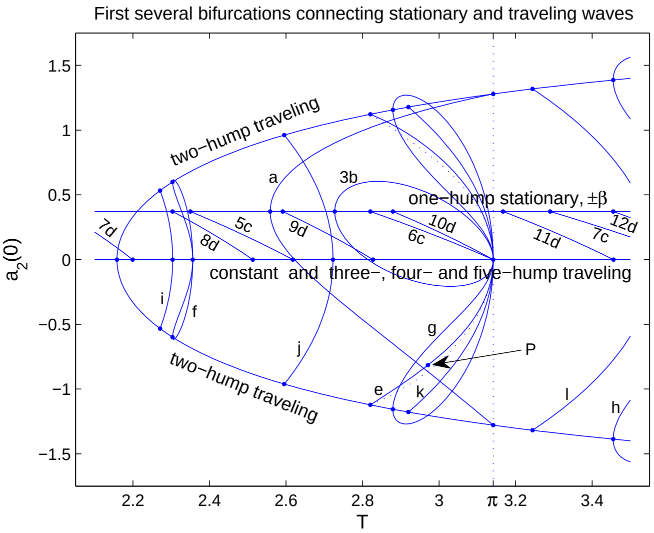


Figure 1. Paths of nontrivial solutions listed in (38)–(41). The second Fourier mode of the eigenvector $z_{N,n}(x)$ in the linearization is nonzero for the pitchfork bifurcations and is zero for the one-sided, oblique-angle bifurcations. The point labeled P corresponds to the solution in Figure 3 below.

We have also computed the next several bifurcations ($n = 2, 3, 4$) from the one-hump stationary solution and found that they connect up with a traveling wave with $N' = n + 1$ humps moving left with speed index $\nu' = -1$, where we denote the bifurcation on the other side of the path by (N', ν', n', m'). By comparing the Fourier coefficients of the last few nontrivial solutions on these paths to those of the linearization about the N'-hump traveling wave, we determined that the bifurcation and oscillation indices satisfy $n' = n$ and $m' = 1$, respectively. Studying these reconnections revealed that the correct way to number the eigenvalues $\omega_{N',n'}$ was to split the double eigenvalues with $n' < N'$ apart as we did in (12) by simultaneously diagonalizing the shift operator and ordering the $\omega_{N',n'}$ via the stride offset of the corresponding eigenvectors (rather than monotonically). Using this ordering, the nontrivial solutions connect up with the N'-hump traveling wave along the $z_{N',n'}$ direction (without involving $z_{N',N'-n'}$). These results are summarized as

$$\begin{aligned} b:&\quad (1,0,2,1) &\longleftrightarrow&\quad (3,-1,2,1),\\ c:&\quad (1,0,3,1) &\longleftrightarrow&\quad (4,-1,3,1),\\ d:&\quad (1,0,4,1) &\longleftrightarrow&\quad (5,-1,4,1). \end{aligned} \tag{39}$$

The labels a, b, c, d in (38) and (39) correspond to the paths labeled $7d$, $8d$, $5c$, a, etc. in the bifurcation diagram. When an integer p precedes a label, it means that the period T that is plotted is p times larger than the fundamental period of the solution represented. Thus, curve $7d$ is the image of curve d (not shown) under the linear transformation $(T, a_2) \mapsto (7T, a_2)$. In our labeling scheme, we just need to multiply ν, m, ν', m' by p to obtain the new path, for example,

$$7d:\quad (1,0,4,7) \quad\longleftrightarrow\quad (5,-7,4,7). \tag{40}$$

In this diagram, we plot $a_2(0)$ versus T with the spatial and temporal phases chosen so the solution is even at $t = 0$. For example, on path d, as we decrease $\rho = a_2(0)$ from 0.371087 to 0, the solution transitions from the one-hump stationary solution to the five-hump left-traveling wave as shown in Figure 2.

It is interesting that the paths labeled a and $3b$ in Figure 1 meet the one-hump stationary solutions in a pitchfork, while the other paths (such as $5c$ and $8d$) meet at an oblique angle from one side only. This is because the second Fourier mode of the eigenvector $z_{1,n}(x)$ in the linearization about the stationary solution is zero in these latter cases, so the change in $a_2(0)$ from that of the stationary solution (namely, 0.371087) is a higher-order effect, (as is the change in T). This explains the oblique angle. We now explain why these bifurcations occur from one side only. When we go beyond the linearization as we have here, we find that $c_2(t) = a_2(t) + ib_2(t)$ has a nearly circular (epitrochoidal) orbit in case a, a circular orbit in case b, and remains constant in time in cases c and d (see Section 4). If one branch of the pitchfork

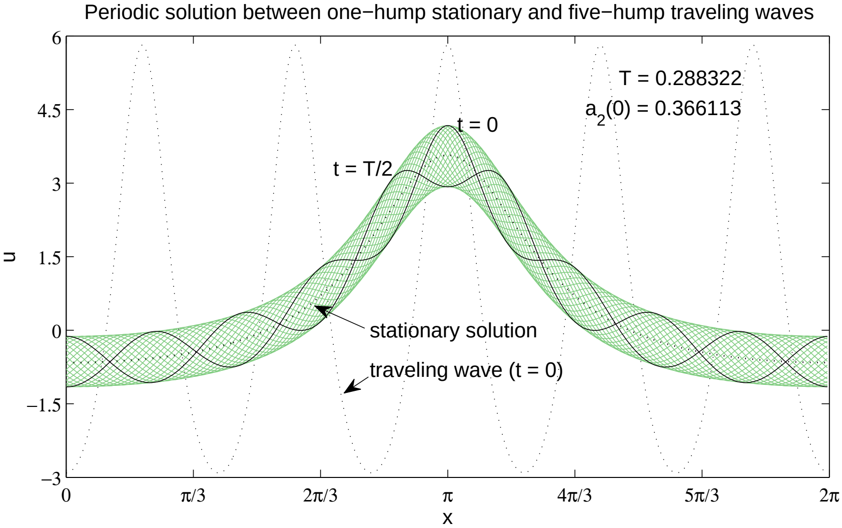


Figure 2. Periodic solution on path d connecting the one-hump stationary solution to the five-hump left-traveling wave ($\alpha_0 = 0.544375$). The second Fourier mode of $z_{1,4}(x)$ is zero, which explains why $a_2(0) = 0.366113$ for this solution is only 1.35% of the way between the stationary solution $a_2(0) = 0.371087$ and the five-hump traveling wave $a_2(0) = 0$.

corresponds to $a_2(0)$, the other is $a_2(T/2)$ since the function $u(\,\cdot\,, T/2)$ also has even symmetry. But in cases c and d, $a_2(0)$ is equal to $a_2(T/2)$ even though the functions $u(\,\cdot\,, 0)$ and $u(\,\cdot\,, T/2)$ are different. These cases also become pitchforks when a different Fourier coefficient $a_K(0)$ is used as the bifurcation parameter.

Next we compute the first several bifurcations from the two-hump traveling waves with mean $\alpha_0 = 0.544375$ and speed index $\nu = -1$. We set $N = 2$, $\nu = -1$, $n \in \{1, 2, 3, 4\}$ and choose the first several legal m values, i.e., values of m that satisfy the bifurcation rules of Table 1 on page 210. For example, the curves labeled i, j, k and l in Figure 1 correspond to the bifurcations $(2, -1, 4, m)$ with $m = 11, 13, 15, 17$; smaller values (and even values) of m are not allowed. In addition to the path a in (38) above, we obtain the paths

$$\begin{aligned}
&e : (2,-1,2,3) \leftrightarrow (3,-3,1,3), && i : (2,-1,4,11) \leftrightarrow (5,-8,3,11),\\
&f : (2,-1,3,6) \leftrightarrow (4,-5,2,6), && j : (2,-1,4,13) \leftrightarrow (5,-9,3,13),\\
&g : (2,-1,3,8) \leftrightarrow (4,-6,2,8), && k : (2,-1,4,15) \leftrightarrow (5,-10,3,15),\\
&h : (2,-1,3,10) \leftrightarrow (4,-7,2,10), && l : (2,-1,4,17) \leftrightarrow (5,-11,3,17).
\end{aligned} \tag{41}$$

The paths f, g and h meet the curve representing the two-hump traveling waves in a pitchfork bifurcation while the others meet obliquely from one side. This, again, is an anomaly of having chosen the second Fourier mode for the bifurcation parameter. The dotted line near the path e is the curve obtained when e is reflected across the T-axis. Solutions on this dotted line correspond to solutions on path e shifted by $\pi/2$ in space, which changes the sign of $\rho = a_2(0)$ but also breaks the even symmetry of the solution at $t = 0$. The paths labeled i, j, k and l are exactly symmetric when reflected about the T-axis because $c_2(t)$ has a circular orbit centered at zero in these cases. It is interesting that so many of the paths in this bifurcation diagram terminate when $T = \pi$ (or a simple rational multiple of π). This is due to the fact that T in (A.1) in Appendix A is independent of α when $n < N$.

The solutions $u(x, t)$ corresponding to points along the paths b, c and d are qualitatively similar to each other. As shown in Figure 2, these solutions look like N'-hump waves traveling over a stationary one-hump carrier signal. At one end of the path the high frequency wave may be viewed as a perturbation of the one-hump stationary solution, while at the other end of the path it is more appropriate to regard the stationary solution as the perturbation, causing the traveling wave to bulge upward as it passes near $x = \pi$ and downward near $x = 0$ and $x = 2\pi$. In all these cases, the solution repeats itself when one of the high frequency waves has moved left one slot to assume the shape of its left neighbor at $t = 0$.

By contrast, the solutions that bifurcate from the two-hump traveling waves, that is, those on the paths listed in (41), have the property that when a wave has moved left one slot to the location that its neighbor occupied at $t = 0$, it has acquired a different shape and must keep progressing a number of slots before it finally lines up with one of the initial waves. This is illustrated in Figure 3 for the solution labeled P in Figure 1 on the path

$$e: \quad (2, -1, 2, 3) \quad \longleftrightarrow \quad (3, -3, 1, 3). \tag{42}$$

This solution is qualitatively similar to the linearized solution $(3, -3, 1, 3)$. There are $N' = 3$ humps oscillating with the same amplitude but with different phases as they travel left. They do not line up with the initial condition again until they have traveled three slots ($\nu' = -3$) and progressed through one cycle ($m'/N' = 3/3$), which leads to a braided effect when the time history of the solution is plotted on one graph. All the solutions on path e are *irreducible* in the sense that there is no smaller time T in which they are periodic (unlike the cases labeled $3b$, $5c$, $7d$, etc. in Figure 1, which are reducible to b, c and d, respectively). Note that although $\nu' = -3$ and $m' = 3$ are both divisible by 3, we cannot reduce $(3, -3, 1, 3)$ to $(3, -1, 1, 1)$ as the latter indices violate the bifurcation rules of Table 1 (page 210). We also mention that at the beginning of the path, near $(2, -1, 2, 3)$, the braiding effect is not present; instead, the solution can be described as two humps bouncing out of phase as they travel left. In one period, they each travel left one slot ($\nu = -1$)

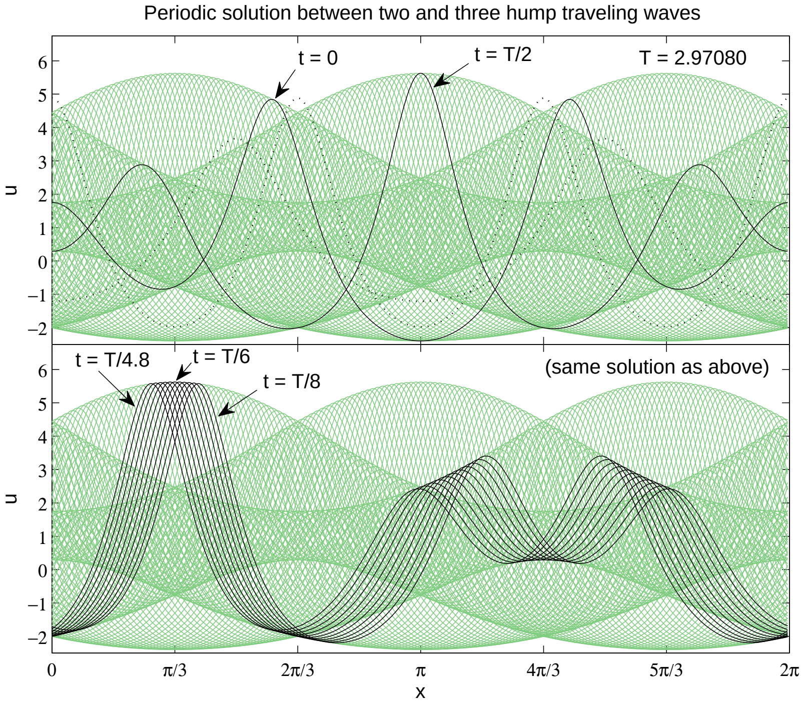


Figure 3. Time-periodic solution (labeled P in Figure 1) on path e connecting two- and three-hump traveling waves. The amplitude of each hump oscillates as it travels left. The dotted curves in the top panel represent the traveling waves at each end of the path at $t = 0$.

and bounce 1.5 times ($m/N = 3/2$) to assume the shape of the other hump at $t = 0$. The transition from this behavior to the braided behavior occurs at the point on path e that a third hump becomes recognizable in the wave profile. The solutions on the paths f, g, h, i, j, k and l are similar to those on path e, but the braiding patterns are more complicated near the right end-points of these paths.

All the traveling waves we have described until now move left. To see what happens to a right-moving wave, we computed the first bifurcation from the simplest such case and obtained the path

$$(1, 1, 1, 2) \quad \longleftrightarrow \quad (2, 0, 1, 2). \tag{43}$$

Thus, the one-hump right-traveling wave is connected to the two-hump stationary solution. Solutions near the left end of this path consist of a large-amplitude, right-moving soliton traveling over a small-amplitude, left-moving soliton. As we progress along the path, the amplitude of the left-moving soliton increases until the solitons cease to fully merge at $t = T/4$ and $t = 3T/4$. Instead, a dimple forms in the

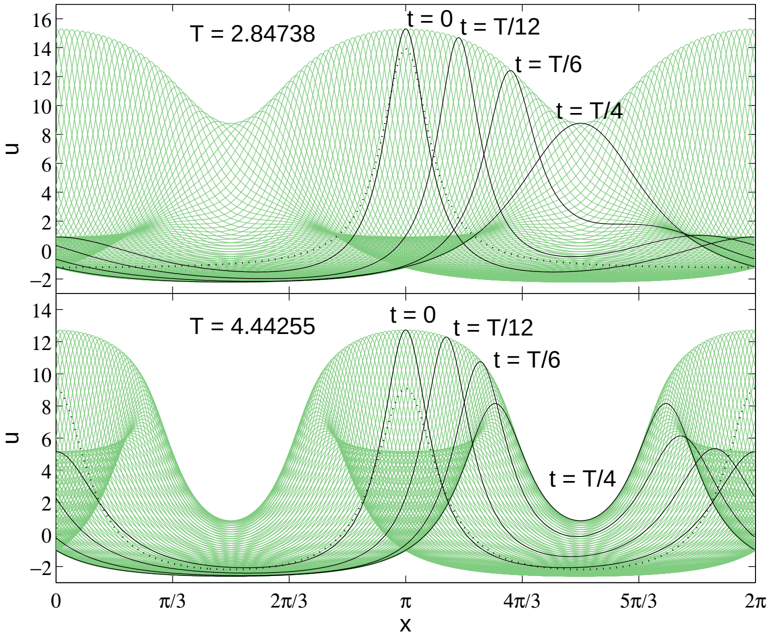


Figure 4. Periodic solutions with mean $\alpha_0 = 0.544375$ between the one-hump right-traveling wave (dotted curve, top panel) and the two-hump stationary solution (dotted curve, bottom panel). Top: a large, right-traveling soliton temporarily merges with a small, left traveling soliton at $t = \frac{1}{4}T$ and $t = \frac{3}{4}T$. Bottom: two solitons traveling in opposite directions bounce off each other at $\frac{1}{4}T$ and $\frac{3}{4}T$ and change direction.

wave profile at these times and the solitons begin to bounce off each other, trading amplitude so the right-moving wave is larger than the left-moving wave. This type of behavior has also been observed by Leveque [22] for the KdV equation for solitons of nearly equal amplitude. Both types of behavior (merging and bouncing off one another) are illustrated in Figure 4. As we proceed further along this path, the solitons settle into a synchronized dancing motion without changing their shape or deviating far from their initial positions. Eventually the "dancing amplitude" becomes small and the nontrivial solution turns into a stationary two-hump solution.

In order to guess a general formula for the relationship between two traveling waves that are connected by a path of nontrivial solutions, we generated two additional paths, namely,

$$\begin{aligned} (2,0,2,2) &\quad\longleftrightarrow\quad (3,-1,1,2),\\ (3,0,3,3) &\quad\longleftrightarrow\quad (4,-1,1,3). \end{aligned} \tag{44}$$

After studying all the paths listed in (38)–(44), we propose the following conjecture, which we prove as part of Theorem 3 in Section 4:

Conjecture 2. The four-parameter sheet of nontrivial solutions with bifurcation parameters (N, ν, n, m) coincides with the sheet with parameters (N', ν', n', m') if and only if

$$\text{if } n < N: \quad N' = N - n, \quad \nu' = \frac{(N-n)\nu + m}{N}, \quad n' = N - 1, \qquad m' = m, \tag{45}$$

$$\text{if } n \geq N: \quad N' = n + 1, \quad \nu' = \frac{(n+1)\nu - m}{N}, \quad n' = n + 1 - N, \quad m' = m. \tag{46}$$

By symmetry, we may interchange the primed and unprimed indices in either formula; thus, $N' > N \Leftrightarrow n < N \Leftrightarrow n' \geq N'$. In most of our numerical calculations, N' turned out to be larger than N. In the exact formulas of Section 4, we find it more convenient to adopt the convention that $N' < N$ since, in that case, all the solutions on the path connecting these traveling waves turn out to be N-particle solutions as described in Section 2.1.

Equations (45) and (46) are consistent with the bifurcation rules of Appendix A in that

$$n < N, \;\; m \in n\nu + N\mathbb{Z} \quad \Rightarrow \quad \nu' \in \mathbb{Z}, \;\; m' \in (n'+1)\nu' + N'\mathbb{Z}, \tag{47}$$

$$n \geq N, \;\; m \in (n+1)\nu + N\mathbb{Z} \quad \Rightarrow \quad \nu' \in \mathbb{Z}, \;\; m' \in n'\nu' + N'\mathbb{Z}. \tag{48}$$

However, if the mean is held constant, they do not necessarily respect the requirements on α_0 listed in Table 1 (page 210). For example, if $\alpha_0 \leq 3$, then (2, 1, 1, 1) is a valid bifurcation, but the reconnection (1, 1, 1, 1) predicted by (45) is legal only if $\alpha_0 = 3$. Interestingly, when we use our numerical method to follow the path of nontrivial solutions that bifurcates from (2, 1, 1, 1) with the mean $\alpha_0 = 1.2$ held constant, it does not connect up with another traveling wave. Instead, as illustrated in Figure 5, as we vary the bifurcation parameter, the two humps (of the solutions labeled A,B,C) grow in amplitude and merge together until they become a single soliton traveling very rapidly on top of a small amplitude stationary hump. As the bifurcation parameter $\rho = a_1(0)$ approaches a critical value, the period T approaches zero and the solution blows up in $L^2(0, 2\pi)$ with the Fourier coefficients of any time-slice decaying more and more slowly.

As another example, the bifurcation (3, 1, 1, 1) is valid when $\alpha_0 \leq 5$ but the reconnection $(2, 1, 2, 1)$ is only valid if $\alpha_0 = 5$. If we hold $\alpha_0 < 5$ constant, the solution blows up as we vary $\rho = a_2(0)$ from 0 to a critical value. However, *if we simultaneously vary the mean* so that it approaches 5, we do indeed reach a traveling wave with bifurcation indices $(2, 1, 2, 1)$. To check this numerically, we started at (3, 1, 1, 1) with $\alpha_0 = 4.8$ (which has $\alpha = \frac{14}{15}$, $|\beta| = 1/\sqrt{31}$) and computed 40 solutions varying ρ from 0 to 0.1 and setting $\alpha_0 = 4.8 + 2\rho$. The

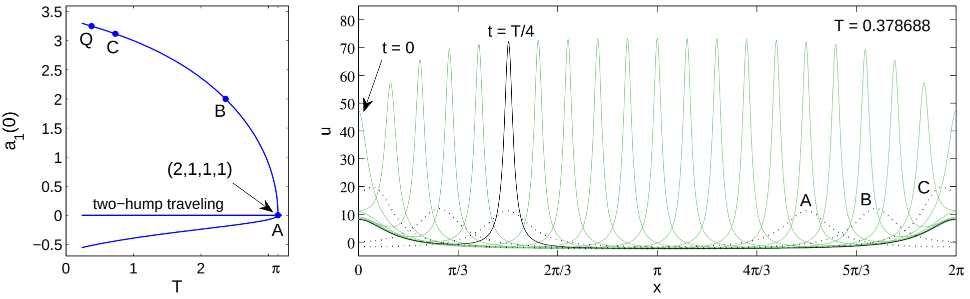


Figure 5. Left: path of nontrivial solutions with mean $\alpha_0 = 1.2$ that bifurcates with indices (2, 1, 1, 1) from the two-hump traveling wave. These solutions do not reconnect with another traveling wave, but instead blow up as $T \to 0$. The solution Q is shown at right, where a large, right-moving soliton travels rapidly over a small, stationary hump. The dotted curves are initial conditions for the points labeled A, B, C at left.

bifurcation at the other end turned out to be (2, 1, 2, 1) with $\alpha_0 = 5$, $\beta = \frac{1}{4}\rho = 0.025$, $\alpha = (1-3\beta^2)/(1-\beta^2)$, $T = \pi/(5-2\alpha)$, as predicted by Conjecture 2. The solutions on this path have the interesting property that the envelope of the solution pinches off into a football shape at one point in the transition from the three-hump traveling wave to the two-hump traveling wave. Using a bracketing technique, we were able to find a solution such that the value of $u(0, t)$ remained constant in time to 8 digits of accuracy. The result is shown in Figure 6.

In summary, it appears that the family of bifurcations with indices (N, ν, n, m) is always connected to the family with indices (N', ν', n', m') given by (45) and (46) by a sheet of nontrivial solutions, but we often have to vary both the mean and

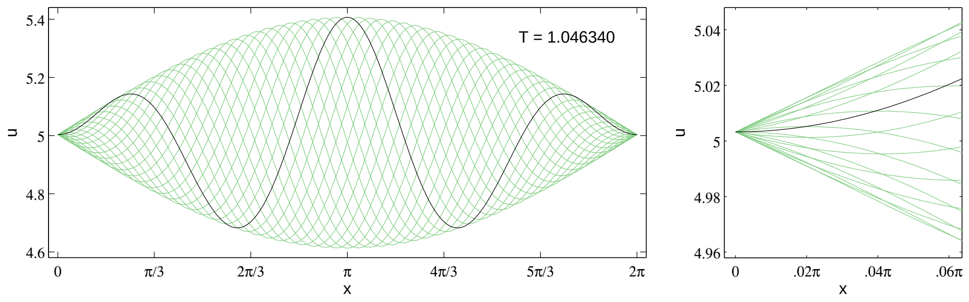


Figure 6. Left: one of the solutions on the path from $\{(3, 1, 1, 1), \beta = -\sqrt{1/31}\}$ to $\{(2, 1, 2, 1), \beta = \frac{1}{40}\}$ consists of a traveling wave inside a football-shaped envelope. The exact solution appears to be of the form $u(x, t) = A + B\left(\sin\frac{x}{2}\right)\sin\left(\frac{5}{2}x - \frac{2\pi}{T}t\right)$.

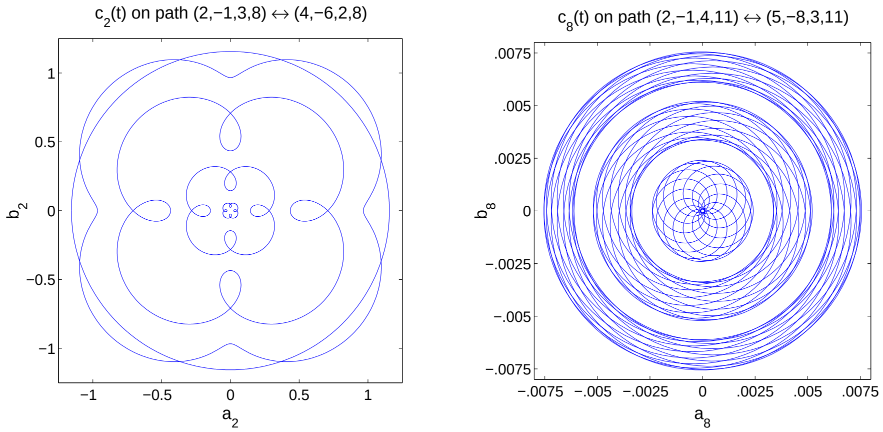
a Fourier coefficient of the initial condition to achieve a reconnection. Thus, the manifold of nontrivial solutions is genuinely two-dimensional (or four dimensional if phase shifts are included). Some of its important properties cannot be seen if we hold the mean α_0 constant.

4. Exact solutions

In this section we use data fitting techniques to determine the analytic form of the numerical solutions of Section 3. We then state a theorem that confirms our numerical predictions and explains why some paths of solutions reconnect with traveling waves when the mean is held fixed while others lead to blow-up. The theorem is proved in Appendix B.

4.1. ***Fourier coefficients and lattice sums.*** One striking feature of the time-periodic solutions we have found numerically is that the trajectories of the Fourier modes $c_k(t)$ are often circular or nearly circular. Other Fourier modes have more complicated trajectories resembling cartioids, flowers and many other familiar “spirograph” patterns (see Figure 7). This led us to experiment with data fitting to try to guess the analytic form of these solutions. The first thing we noticed was that the trajectories of the spatial Fourier coefficients are band-limited in time, with the width of the band growing linearly with the wave number:

$$u(x,t)=\sum_{k=-\infty}^{\infty} c_k(t)e^{ikx}, \quad c_k(t)=\sum_{j=-\infty}^{\infty} c_{kj}e^{-ij\frac{2\pi}{T}t}, \quad c_{kj}=0 \text{ if } |j|>r|k|. \tag{49}$$


$c_2(t)$ on path (2,−1,3,8) ↔ (4,−6,2,8)

$c_8(t)$ on path (2,−1,4,11) ↔ (5,−8,3,11)

Figure 7. Left: trajectories $c_2(t)$ for five solutions on path g in (41). The evolution of $c_2(t)$ on paths f and h in (41) are similar, but with three- and five-fold symmetry rather than four. Right: trajectories $c_8(t)$ for three solutions on path i in (41).

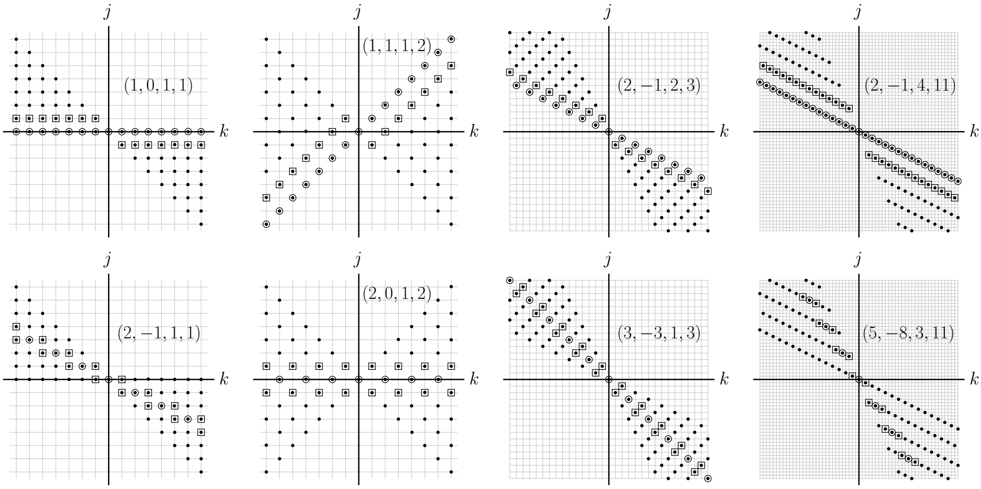


Figure 8. Each pair (aligned vertically) corresponds to a path of nontrivial solutions connecting two traveling waves. Solid dots represent the nonzero entries c_{kj} in (49) of the exact solutions along this path; open circles represent a traveling wave; and open squares represent the nonzero entries d_{kj} in the linearization about the traveling wave.

Here r is a fixed positive integer (depending on which path of nontrivial solutions u belongs to) and the c_{kj} are real numbers when a suitable choice of spatial and temporal phase is made. Since u is real, these coefficients satisfy $c_{-k,-j} = c_{kj}$.

Each path of nontrivial time-periodic solutions has a lattice pattern of nonzero Fourier coefficients c_{kj} associated with it. In Figure 8, we show the lattice of integers (k, j) such that $c_{kj} \neq 0$ for solutions on the paths

$$\begin{aligned} &(1,0,1,1) \longleftrightarrow (2,-1,1,1), \qquad (2,-1,2,3) \longleftrightarrow (3,-3,1,3), \\ &(1,1,1,2) \longleftrightarrow (2,0,1,2), \qquad (2,-1,4,11) \longleftrightarrow (5,-8,3,11). \end{aligned} \tag{50}$$

All solutions on a given path have the same lattice pattern (of solid dots), but different paths have different patterns. One may show that if $u(x,t)$ is of the form (49) and

$$\frac{k}{2}\sum_{l,p} c_{lp}c_{k-l,j-p} = \Big(k|k| + \frac{2\pi}{T}j\Big)c_{kj}, \quad (k > 0,\ j \in \mathbb{Z}), \tag{51}$$

then $u(x,t)$ satisfies the Benjamin–Ono equation, $uu_x = Hu_{xx} - u_t$. The traveling waves at each end of the path have fewer nonzero entries, namely,

$$\tilde{c}_{kj} = \left\{\begin{array}{ll} N\alpha + \dfrac{2\pi\nu}{NT}, & k = j = 0, \\ 2N\beta^{|k|/N}, & k \in N\mathbb{Z}\setminus\{0\},\ j = \dfrac{\nu k}{N} \\ 0, & \text{otherwise.} \end{array}\right\} \qquad \left(\alpha = \frac{1-3\beta^2}{1-\beta^2}\right). \tag{52}$$

Here a tilde is used to indicate a solution about which we linearize. Substitution of $c_{kj} = \tilde{c}_{kj} + \varepsilon d_{kj}$ into (51) and matching terms of order ε leads to an eigenvalue problem with solution

$$d_{kj} = \begin{cases} \hat{z}_{N,n}(k), & k \in k_{N,n} + N\mathbb{Z}, \quad j = (k\nu - m)/N, \\ \hat{z}_{N,n}(-k), & k \in -k_{N,n} + N\mathbb{Z}, \quad j = (k\nu + m)/N, \\ 0, & \text{otherwise,} \end{cases} \tag{53}$$

with $\hat{z}_{N,n}(k)$ as in (15). The nonzero coefficients d_{kj} in this linearization are represented by open squares in Figure 8. Recall from (15) that if $n \geq N$ and $k \leq n - N$ then $\hat{z}_{N,n}(k) = 0$, but if $n < N$, the nonzero entries of $\hat{z}_{N,n}(k)$ continue in both directions (with k approaching $+\infty$ or $-\infty$). This is why the rows of open squares terminate in the graphs in the top row of Figure 8 rather than continuing past the origin as in the graphs in the bottom row.

4.2. ***Elementary symmetric functions.*** It is interesting that the lattice patterns that arise for the exact solutions (beyond the linearization) contain only positive integer combinations of the lattice points of the linearization and of the traveling wave (treating the left and right half-planes separately). Somehow the double convolution in (51) leads to exact cancellation at all other lattice sites! This suggests that the c_{kj} have a highly regular structure that generalizes the simple power law decay rate of the Fourier coefficients $\hat{u}_{\text{stat}}(k; N, \beta)$ of the N-hump stationary solution.

The first step to understand this is to grasp that there is a close connection between the trajectories of the Fourier coefficients and the trajectories of the elementary symmetric functions of the particles $\beta_1, \dots, \beta_N$ in (2). Specifically, because the Fourier coefficients of $\phi(x; \beta)$ in (4) are of the form $2\beta^k$ for $k \geq 1$, we have

$$\beta_1^k(t) + \cdots + \beta_N^k(t) = (1/2)c_k(t), \quad \left(k \geq 1,\ c_k(t) = \frac{1}{2\pi}\int_0^{2\pi} u(x,t)e^{-ikx}\,dx\right). \tag{54}$$

Next we define the elementary symmetric functions σ_j via

$$\sigma_0 = 1, \quad \sigma_j = \sum_{l_1 < \cdots < l_j} \beta_{l_1} \cdots \beta_{l_j}, \quad (j = 1, \dots, N), \tag{55}$$

so that

$$P(z) := \prod_{l=1}^{N} (z - \beta_l) = \sum_{j=0}^{N} (-1)^j \sigma_j z^{N-j}. \tag{56}$$

It is well known [38] that the companion matrix Σ of P has the Jordan canonical form

$$\Sigma = \begin{pmatrix} 0 & 1 & & \\ & \ddots & \ddots & \\ 0 & \cdots & 0 & 1 \\ \pm\sigma_N & \cdots & -\sigma_2 & \sigma_1 \end{pmatrix}, \quad V^{-1}\Sigma V = \begin{pmatrix} J_1 & & \\ & \ddots & \\ & & J_m \end{pmatrix}, \quad J_r = \begin{pmatrix} \beta_{l(r)} & 1 & 0 \\ 0 & \ddots & 1 \\ 0 & 0 & \beta_{l(r)} \end{pmatrix},$$

where $l:\{1,\dots,m\}\to\{1,\dots,N\}$ is an enumeration of the *distinct* roots of $P(z)=0$ and the size of the Jordan block J_r is equal to the multiplicity of $\beta_{l(r)}$. As a result, the trace of powers of Σ will give the power sums of the β_l, and hence the Fourier coefficients:

$$c_k = 2\,\mathrm{tr}(\Sigma^k), \qquad (k \ge 1). \tag{57}$$

Thus, if the elementary symmetric functions are finite sums of circular orbits, then the Fourier coefficients will be as well, and we expect higher Fourier modes to involve more terms, in accordance with our findings above.

Before presenting our main result, we note that once the mapping (45) from (N, ν, n, m) to (N', ν', n', m') is known, we can choose N, ν, N' and ν' independently, subject to the conditions

$$N' < N, \qquad \nu' > \frac{N'}{N}\nu. \tag{58}$$

The first condition is merely a labeling convention while the second is an actual restriction on which traveling waves are connected together by a path of nontrivial solutions. The formulas of Conjecture 2 then imply that

$$m = m' = N\nu' - N'\nu > 0, \qquad n = N - N', \qquad n' = N - 1. \tag{59}$$

After extensive experimentation with data fitting on the numerical simulations described in Section 3, we arrived at the form (61) below for the polynomial P. We then substituted the ansatz (60) into (1) to obtain algebraic relationships between A, B, C, α_0, ω, N, N', ν and ν', namely, (B.9)–(B.11) in Appendix B. We solved these using Mathematica to obtain formulas for A, B and ω in terms of C, α_0, N, N', ν and ν'. We had to break the analysis into three cases depending on whether ν is less than, equal to, or greater than ν'. By comparing our exact solutions with previously known representations of multiperiodic solutions [26], we found that all three cases could be unified by replacing C and α_0 by two new parameters, ρ and ρ', related to C and α_0 by (62) below. We give a direct proof of the following theorem in Appendix B.

Theorem 3. *Let N, N', ν and ν' be integers with $N > N' > 0$ and $N\nu' - N'\nu > 0$. There is a four-parameter family of time-periodic solutions connecting the traveling* *wave bifurcations* $(N', \nu', N-1, m)$ *and* $(N, \nu, N-N', m)$, *where* $m = N\nu' - N'\nu$. *These solutions are of the form*

$$u(x,t) = \alpha_0 + \sum_{l=1}^{N} \phi(x; \beta_l(t)), \qquad \hat{\phi}(k;\beta) = \begin{cases} 2\bar{\beta}^{|k|}, & k < 0, \\ 0, & k = 0, \\ 2\beta^k, & k > 0, \end{cases} \tag{60}$$

where $\beta_1(t), \dots, \beta_N(t)$ *are the roots of the polynomial*

$$P(z) = z^N + Ae^{-i\nu'\omega t} z^{N-N'} + Be^{-i(\nu-\nu')\omega t} z^{N'} + Ce^{-i\nu\omega t}, \tag{61}$$

with

$$\begin{aligned} A &= e^{i\nu'\omega t_0} e^{-iN'x_0} \sqrt{\frac{N-N'+\rho+\rho'}{N+\rho+\rho'}} \sqrt{\frac{(N+\rho')\rho'}{N'(N-N')+(N+\rho')\rho'}}, \\ B &= e^{i(\nu-\nu')\omega t_0} e^{-i(N-N')x_0} \sqrt{\frac{(N+\rho')\rho'}{N'(N-N')+(N+\rho')\rho'}} \sqrt{\frac{\rho}{N-N'+\rho}}, \\ C &= e^{i\nu\omega t_0} e^{-iNx_0} \sqrt{\frac{\rho}{N-N'+\rho}} \sqrt{\frac{N-N'+\rho+\rho'}{N+\rho+\rho'}}, \\ \alpha_0 &= \frac{N^2\nu' - (N')^2\nu}{m} - 2\rho - \frac{2N'(\nu'-\nu)}{m}\rho', \\ \omega &= \frac{2\pi}{T} = \frac{N'(N-N')(N+2\rho')}{m}. \end{aligned} \tag{62}$$

The four parameters are $\rho \geq 0$, $\rho' \geq 0$, $x_0 \in \mathbb{R}$ and $t_0 \in \mathbb{R}$. The N- and N'-hump traveling waves occur when $\rho' = 0$ and $\rho = 0$, respectively. When both are zero, we obtain the constant solution $u(x,t) \equiv (N^2\nu' - (N')^2\nu)/m$.

Remark 4. The parameters x_0 and t_0 are spatial and temporal phase shifts. A straightforward calculation shows that if u has parameters ρ, ρ', x_0 and t_0 in Theorem 3 while $\tilde{u}$ has parameters ρ, ρ', 0 and 0, then $u(x,t) = \tilde{u}(x - x_0, t - t_0)$.

There are two features of this theorem that are new. First, it had not previously been observed that the dynamics of the Fourier modes of multiperiodic solutions was so simple. And second, in our representation, it is clear that these solutions reduce to traveling waves in the limit as ρ or ρ' approaches zero. By contrast, other representations become indeterminate in the equivalent limit, and are missing a key degree of freedom (the mean) to allow bifurcation between levels of the hierarchy of multiperiodic solutions.

4.3. ***Three types of reconnection.*** We now wish to explain why following a path of nontrivial solutions with the mean α_0 held fixed sometimes leads to reconnection with a different traveling wave and sometimes leads to blow-up of the initial condition. By Theorem 3, α_0 depends on the parameters ρ and ρ' via

$$\alpha_0 = \alpha_0^* - 2\rho - \frac{2N'(\nu' - \nu)}{m}\rho', \quad \alpha_0^* := \frac{N^2\nu' - (N')^2\nu}{m}. \tag{63}$$

If we hold α_0 fixed, then ρ and ρ' must satisfy

$$2\rho + \frac{2N'(\nu' - \nu)}{m}\rho' = (\alpha_0^* - \alpha_0). \tag{64}$$

This is a line in the ρ-ρ'-plane whose intersection with the first quadrant gives the set of legal parameters for a time-periodic solution to exist. We assume the mean is chosen so that this intersection is nonempty. If the ρ- or ρ'-intercept of this line is positive, the corresponding traveling wave bifurcation exists. There are three cases to consider.

Case 1. ($\nu < \nu'$) Both intercepts will be positive as long as $\alpha_0 < \alpha_0^*$. Thus, a reconnection occurs regardless of which side of the path we start on.

Case 2. ($\nu = \nu'$) The line (64) is vertical in this case, so $\rho = (\alpha_0^* - \alpha_0)/2$ remains constant as we vary ρ' from 0 to ∞. As $\rho' \to \infty$, we see from (62) that $T \to 0$, $A \to 1$, and B and C both approach $\sqrt{\rho/(N - N' + \rho)}$. In this limit, N' of the roots β_l lie on the unit circle at $t = 0$, indicating that the norm of the initial condition blows up as $\rho' \to \infty$.

Case 3. ($\nu > \nu'$) The line (64) has positive slope in this case. If $\alpha_0 < \alpha_0^*$, a bifurcation from the N'-hump traveling wave exists. If $\alpha_0 > \alpha_0^*$, a bifurcation from the N-hump traveling wave exists. And if $\alpha_0 = \alpha_0^*$, a bifurcation directly from the constant solution $u = \alpha_0^*$ to a nontrivial time periodic solution exists. In any of these cases, another traveling wave is not reached as we increase ρ and ρ' to ∞. Instead, $T \to 0$ and A, B and C all approach 1. As a result, all the roots β_l approach the unit circle, indicating that the norm of the initial condition blows up as $\rho, \rho' \to \infty$.

Example 5. Consider the three-particle solutions on the path $e : (2, -1, 2, 3) \leftrightarrow (3, -3, 1, 3)$ in Figures 1 and 3. Since $-3 = \nu < \nu' = -1$, we do not need to vary the mean in order to reconnect with a traveling wave on the other side of the path. Suppose $\alpha_0 < \alpha_0^* = 1$ is held fixed. Then the parameters ρ and ρ' in Theorem 3 satisfy

$$\rho = \frac{1}{2}\Big(1 - \alpha_0 - \frac{8}{3}\rho'\Big), \quad 0 \le \rho' \le \frac{3(1 - \alpha_0)}{8}. \tag{65}$$

The solutions $u(x, t)$ on this path are of the form (60) with particles $\beta_l(t)$ evolving as the roots of the polynomial

$$P(z) = z^3 + Ae^{i\omega t}z + Be^{2i\omega t}z^2 + Ce^{3i\omega t}, \tag{66}$$

where

$$\begin{aligned} A &= \sqrt{\frac{(9-3\alpha_0-2\rho')(3+\rho')\rho'}{(21-3\alpha_0-2\rho')(2+\rho')(1+\rho')}}, \\ B &= \sqrt{\frac{(3-3\alpha_0-8\rho')(3+\rho')\rho'}{(9-3\alpha_0-8\rho')(2+\rho')(1+\rho')}}, \\ C &= \sqrt{\frac{(9-3\alpha_0-2\rho')(3-3\alpha_0-8\rho')}{(21-3\alpha_0-2\rho')(9-3\alpha_0-8\rho')}}, \\ \omega &= \frac{2\pi}{T} = \frac{2(3+2\rho')}{3}. \end{aligned} \tag{67}$$

The transition from the two- to three-hump traveling wave occurs as we decrease the bifurcation parameter ρ' from $3(1-\alpha_0)/8$ to 0. This causes C to increase from 0 to $\sqrt{(1-\alpha_0)/(7-\alpha_0)}$ and A to decrease from $\sqrt{(3-3\alpha_0)/(19-3\alpha_0)}$ to 0. B is zero at both ends of the path.

The trajectories $\beta_1(t)$, $\beta_2(t)$ and $\beta_3(t)$ for $\alpha_0 = 0.544375$ and four choices of ρ' are shown in Figure 9. For this value of the mean, ρ' varies from 0.17086 to 0. Note that the bifurcation from the two-hump traveling wave causes a new particle to nucleate at the origin. As ρ' decreases, the new particle's trajectory grows in amplitude until it joins up with the orbits of the outer particles. There is a critical value of ρ' at which the particles collide and the solution of the ODE (3) ceases to exist for all time; nevertheless, the representation of u in terms of P in (B.1) in Appendix B remains well-behaved and does satisfy (1) for all time. Thus, a change in topology of the orbits does not manifest itself as a singularity in the solution of the

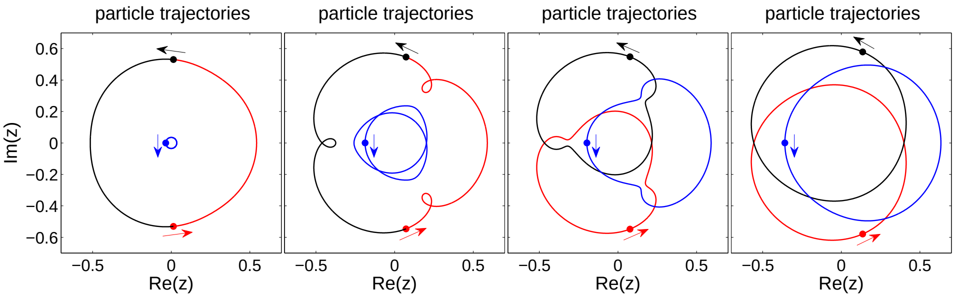


Figure 9. Trajectories $\beta_l(t)$ for four solutions on the path $(2,-1,2,3) \leftrightarrow (3,-3,1,3)$ with mean $\alpha_0 = 0.544375$. The markers give the position of the β_l at $t = 0$. The value of ρ' in (65) is, from left to right: 0.1707, 0.1642, 0.1634 and 0.1369. In Figure 3, $\rho' = 0.0862$.

PDE. As ρ' decreases further, the three orbits become nearly circular and eventually coalesce into a single circular orbit (with $\nu = -3$) at the three-hump traveling wave. The "braided" effect of the solution shown in Figure 3 is recognizable for $\rho' \leq 0.15$ or so for this value of the mean.

5. Interior bifurcations

We conclude this work by mentioning that our numerical method for following paths of nontrivial solutions from one traveling wave to another occasionally wanders off course, following an interior bifurcation rather than reaching the traveling wave on the other side of the original path. These interior bifurcations lead to new paths of nontrivial solutions that are more complicated than those on the original path. For example, on the path

$$(1,1,1,2) \longleftrightarrow (2,0,1,2), \tag{68}$$

Theorem 3 tells us that the exact solution is a two-particle solution with elementary symmetric functions of the form

$$\sigma_1(t) = -(Ae^{-i\omega t} + Be^{i\omega t}), \quad \sigma_2(t) = C. \tag{69}$$

We freeze $\alpha_0 < \alpha_0^* = 2$, set $\rho = \frac{1}{2}(2 - \alpha_0 - \rho')$, and determine that

$$\begin{aligned}
A &= e^{-i(x_0 - \omega t_0)} \sqrt{\frac{(4 - \alpha_0 + \rho')(2 + \rho')\rho'}{(6 - \alpha_0 + \rho')(1 + \rho')^2}}, \\
B &= e^{-i(x_0 + \omega t_0)} \sqrt{\frac{(2 - \alpha_0 - \rho')(2 + \rho')\rho'}{(4 - \alpha_0 - \rho')(1 + \rho')^2}}, \\
C &= e^{-i(2x_0)} \sqrt{\frac{(4 - \alpha_0 + \rho')(2 - \alpha_0 - \rho')}{(6 - \alpha_0 + \rho')(4 - \alpha_0 - \rho')}}, \\
\omega &= \frac{2\pi}{T} = 1 + \rho'.
\end{aligned} \tag{70}$$

In Figure 10, we show the bifurcation diagram for the transition from the one-hump right-traveling wave (labeled P) to the two-hump stationary solution (labeled Q). This diagram was computed numerically before we had any idea that exact solutions for this problem exist; therefore, we used the real part of the first Fourier mode at $t = 0$ for the bifurcation parameter rather than ρ'. We can obtain the same curves analytically as follows. The upper curve from P to Q (containing A1–A5) can be plotted parametrically by setting $x_0 = \pi/2$ and $t_0 = \pi/2\omega$ in (70), varying ρ' from $2 - \alpha_0$ to 0, holding $\alpha_0 = 0.544375$ fixed, and plotting $-2(A + B)$ versus

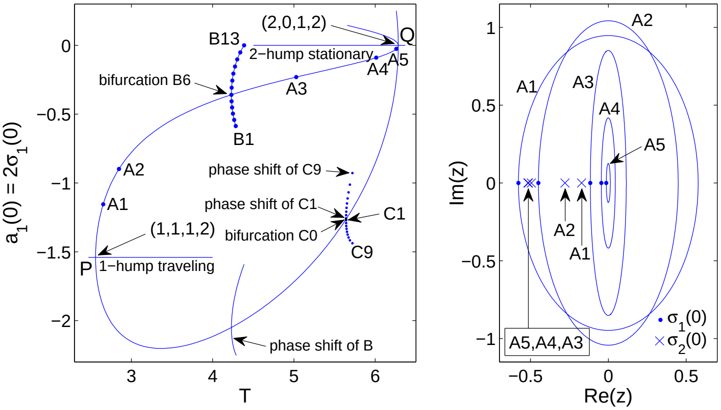


Figure 10. Left: bifurcation diagram showing several interior bifurcations on the path (1, 1, 1, 2) → (2, 0, 1, 2). Right: trajectories of the elementary symmetric functions $\sigma_1(t)$, which have elliptical, clockwise orbits, and $\sigma_2(t)$, which remain stationary in time, for the solutions labeled A1–A5 in the bifurcation diagram.

$T = 2\pi/(1+\rho')$. The lower curve from P to Q is obtained in the same fashion if we instead set $x_0 = t_0 = 0$.

As illustrated in Figure 10, solutions such as A1–A5 on the upper path have $\sigma_1(t)$ executing elliptical, clockwise orbits that start out circular at the one-hump traveling wave but become more eccentric and collapse to a point as we progress toward the two-hump stationary solution Q. Meanwhile, $\sigma_2(t)$ remains constant in time, nucleating from the origin at the one-hump traveling wave and terminating with $\sigma_2 \equiv -\sqrt{(2-\alpha_0)/(6-\alpha_0)}$ at the two-hump stationary solution. On the lower path, the major axis of the orbit of σ_1 is horizontal rather than vertical and σ_2 moves right rather than left as we move from P to Q.

When computing these paths from P to Q, we encountered two interior bifurcations. In the bifurcation labeled B6 in Figure 10, an additional elementary symmetric function nucleates at the origin and the trajectories of σ_1 and σ_2 become more complicated. Through data fitting, we find that

$$\sigma_1(t) = -(Ae^{-i\omega t} + Be^{i\omega t} + C_1e^{3i\omega t}), \tag{71}$$

$$\sigma_2(t) = C + C_2e^{2i\omega t} + C_3e^{4i\omega t}, \tag{72}$$

$$\sigma_3(t) = -C_4e^{3i\omega t}, \tag{73}$$

where the new coefficients C_j are all real parameters. We have not attempted to derive algebraic relationships among these parameters to obtain exact solutions. These trajectories are shown in Figure 11 for the solutions labeled B1–B13 in the bifurcation diagram. The additional term in (71) causes the elliptical orbit of $\sigma_1(t)$

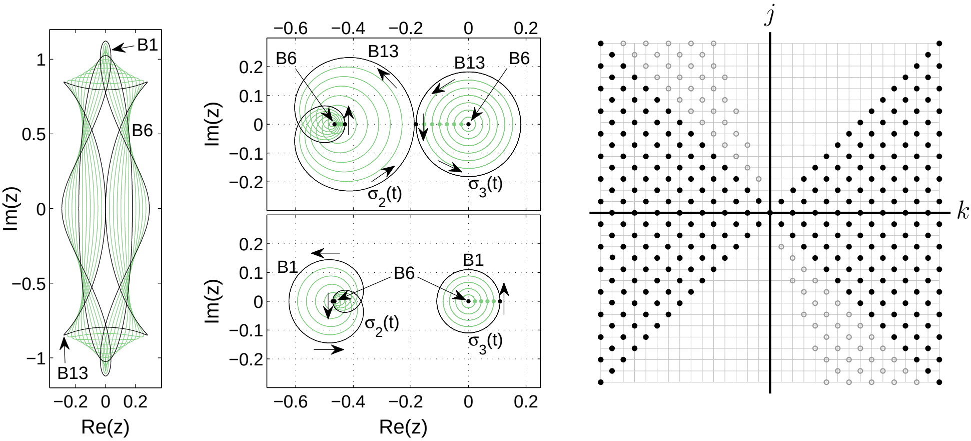


Figure 11. Left: trajectories of $\sigma_1(t)$ for solutions labeled B1–B13 in Figure 10. Center: trajectories of $\sigma_2(t)$ and $\sigma_3(t)$. Since B6 is on the original path from P to Q, $\sigma_2(t)$ is constant and $\sigma_3(t) \equiv 0$ for this solution. Right: the interior bifurcation causes additional lattice coefficients c_{kj} to become nonzero; grey circles represent the new terms.

to deform by bulging out in the vertical and horizontal directions while pulling in along the diagonal directions (or vice versa, depending on which direction we follow the bifurcation). Meanwhile, $\sigma_2(t)$ ceases to be constant and $\sigma_3(t)$ ceases to be zero. To avoid clutter, we plotted the trajectories $\sigma_2(t)$ and $\sigma_3(t)$ for B1–B6 separately from B6–B13, illustrating the effect of following the bifurcation in one direction or the other. The additional terms in (71)–(73) cause the lattice pattern of nonzero entries $c_{kj} = \frac{1}{T}\int_0^T c_k(t)e^{ij\omega t}\,dt$ to become more complicated, where we recall that in this case,

$$c_k(t) = \frac{1}{2\pi}\int_0^{2\pi} u(x,t)e^{-ikx}\,dx = 2\,\mathrm{tr}\left[\begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ \sigma_3(t) & -\sigma_2(t) & \sigma_1(t) \end{pmatrix}^k\right].$$

The solid dots in Figure 11 represent the nonzero entries of solutions on the original path from P to Q while grey circles show the additional terms that are nonzero after the bifurcation at B6. Although this bifurcation causes some of the unoccupied lattice sites to be filled in, the new lattice pattern is rather similar to the original pattern and maintains its checkerboard structure. Also, this bifurcation leads to symmetric perturbations of the Fourier mode trajectories, and is also present (in a phase shifted form) along the lower path from P to Q.

In the bifurcation labeled C0 in Figure 10, the fill-in pattern of the lattice representation is much more complicated, and in fact the checkerboard structure of the nonzero coefficients c_{kj} is destroyed; see Figure 12. But actually, the elementary symmetric functions behave similarly to the previous case: By fitting our numerical data, we find that

$$\sigma_1(t) = -(Ae^{-i\omega t} + Be^{i\omega t} + C_1 e^{4i\omega t}), \tag{74}$$

$$\sigma_2(t) = C + C_2 e^{3i\omega t} + C_3 e^{5i\omega t}, \tag{75}$$

$$\sigma_3(t) = -C_4 e^{4i\omega t}, \tag{76}$$

so each of the new terms executes one additional loop per cycle of the periodic solution in comparison to the corresponding term in (71)–(73). This extra loop causes a star-shaped perturbation of the σ_1 ellipse instead of the rectangular and diamond shaped perturbations seen previously in Figure 11. As a result, this bifurcation is not present on the upper path from P to Q because the symmetry of the perturbation does not respect the 90 degree rotation of the orbit $\sigma_1(t)$ associated with the $\frac{\pi}{2}$-spatial and $\frac{T}{4}$-temporal phase shifts that relate solutions on the upper and lower paths from P to Q.

To follow the bifurcation at C0 in the other direction, we can use the same numerical values for A, B, C, C_1, C_2, C_3, C_4 in (74)–(76) after changing the signs of the latter four parameters. This causes the trajectories of σ_1 in Figure 12 to be rotated 180° with a corresponding $T/2$ phase-shift in time so that the initial position $\sigma_1(0)$ remains on the left side of the figure. Meanwhile, the trajectory of $\sigma_2(t)$

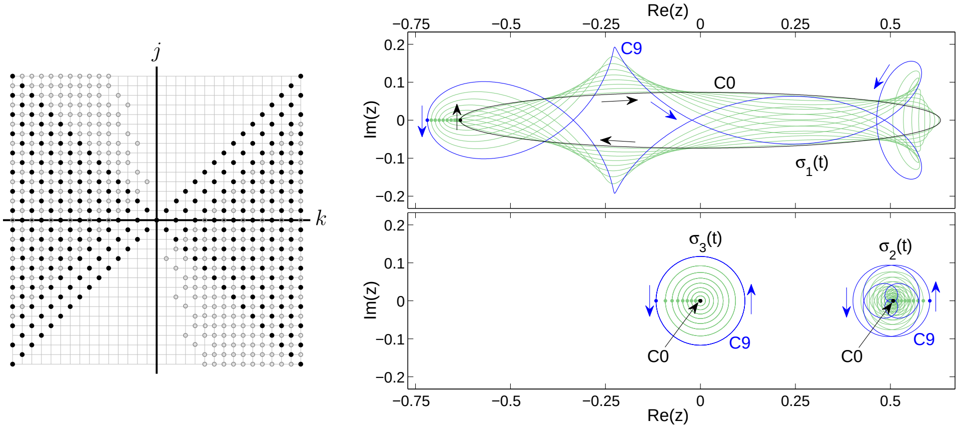


Figure 12. Left: this interior bifurcation causes more lattice coefficients to become nonzero than the interior bifurcation of Figure 11. Right: trajectories of $\sigma_1(t)$, $\sigma_2(t)$, and $\sigma_3(t)$ for the solutions labeled C0-C9 in Figure 10. The long axis of the ellipse C0 is horizontal because we start from the bottom branch connecting P to Q in Figure 10.

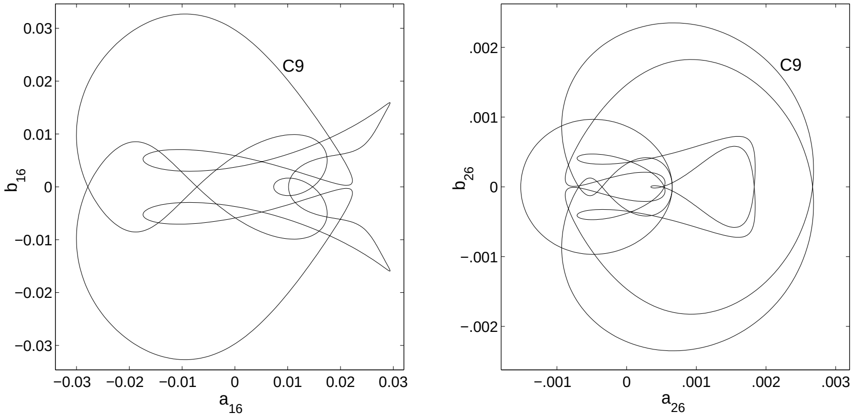


Figure 13. The trajectories of the Fourier modes become very complicated after the interior bifurcation occurs. Here we show the 16th (left) and 26th (right) Fourier modes $c_k(t) = a_k(t) + ib_k(t)$ over one period. It was clearly essential to use a high order (in fact, spectrally accurate) numerical method to resolve these dynamics when computing time-periodic solutions.

experiences a $T/2$ phase-shift in time with no change in the location of the orbit, and $\sigma_3(t)$ starts on the opposite side of its circular trajectory about the origin.

In Figure 13, we show the orbits of the 16th and 26th Fourier modes for the solution labeled C9 in the bifurcation diagram of Figure 10. As the index of the Fourier mode increases, these trajectories become increasingly complicated (involving more nonzero terms c_{kj} in the lattice representation), but also decay exponentially so that the amplitude of the orbit is eventually smaller than can be resolved using floating point arithmetic. We emphasize that these trajectories were resolved to full machine precision by our general purpose numerical method for finding periodic solutions of nonlinear PDE (without any knowledge of the solitonic structure of the solutions). Everything we learned about the form of the exact solutions came about from studying these numerical solutions, which was possible only because our numerical results are correct to 10-15 digits of accuracy.

Appendix A. Bifurcation formulas and rules

In this section we collect formulas relating the period, mean and decay parameter at a bifurcation. We also identify bifurcation rules governing the legal values of α_0 for a given set of bifurcation indices.

In computing the nullspace $\mathcal{N} = \ker DF(U_0, T)$ in Section 2.3, we considered N, ν, β, T (and hence α_0) to be given and searched for compatible indices n and m. The decay parameter $|\beta|$, the mean α_0, and the period T cannot be specified independently; any two of them determines the third. We now derive formulas for the period and mean in terms of (N, ν, n, m) and β. To simplify the formulas, we work with $\alpha = (1-3|\beta|^2)/(1-|\beta|^2)$ instead of β. Note that as we increase $|\beta|$ from 0 to 1, α decreases from 1 to $-\infty$. For the period, we have

$$T = \frac{2\pi m}{N\omega_{N,n}} = \begin{cases} \dfrac{2\pi m}{Nn(N-n)} & n < N, \\ \dfrac{2\pi m}{N(n+1-N)(n+1+N(1-\alpha))} & n \geq N, \end{cases} \tag{A.1}$$

so the period is independent of β when $n < N$, and otherwise decreases to zero as $|\beta|$ varies from 0 to 1. For the mean, α_0, we note that

$$cT = \frac{2\pi\nu}{N}, \quad c = \alpha_0 - N\alpha \quad \Rightarrow \quad \alpha_0 = N\alpha + \frac{2\pi\nu}{NT}. \tag{A.2}$$

Hence, using $(2\pi/NT) = (\omega_{N,n}/m)$, we obtain

$$\alpha_0 = \begin{cases} N + \dfrac{n(N-n)}{m}\nu - (1-\alpha)N, & n < N, \\ N + \dfrac{(n+1-N)(n+1)}{m}\nu - \left(1 - \dfrac{n+1-N}{m}\nu\right)N(1-\alpha), & n \geq N. \end{cases} \tag{A.3}$$

Thus, as $|\beta|$ varies from 0 to 1, the mean α_0 decreases to $-\infty$ if $n < N$, and otherwise either decreases to $-\infty$, increases to $+\infty$, or is independent of β, depending on the sign of $[m - (n+1-N)\nu]$.

In practice, we often wish to start with N, ν, n, m and α_0 and determine T and $|\beta|$ from these. However, not all values of α_0 are compatible with a given set of indices. The bifurcation rules are summarized in Table 1.

Solving (A.3) for α yields

$$\alpha = \begin{cases} 1 - \dfrac{(N-\alpha_0)m + n(N-n)\nu}{Nm}, & n < N, \\ 1 - \dfrac{(N-\alpha_0)m + (n+1-N)(n+1)\nu}{[m-(n+1-N)\nu]N}, & n \geq N. \end{cases} \tag{A.4}$$

The corresponding period is given by

$$T = \begin{cases} \dfrac{2\pi m}{Nn(N-n)}, & n < N, \\ \dfrac{2\pi\left(\frac{m}{n+1-N} - \nu\right)}{N(n+1+N-\alpha_0)}, & n \geq N. \end{cases} \tag{A.5}$$

In the indeterminate cases $\{n \ge N,\ m = (n+1-N)\nu,\ \alpha_0 = n+1+N\}$, any $\alpha \le 1$ is allowed and formula (A.1) should be used to determine T.

If we express n, n', m and m' in terms of N, ν, N', ν', then (A.1) and (A.3) give

$$\begin{aligned} T &= \frac{2\pi(N\nu' - N'\nu)}{N'(N-N')N}, & \alpha_0 &= \alpha_0^* - (1-\alpha)N, \\ T' &= \frac{2\pi(N\nu' - N'\nu)}{N'(N-N')[N+(1-\alpha')N']}, & \alpha_0' &= \alpha_0^* - \frac{\nu'-\nu}{N\nu' - N'\nu}(N')^2(1-\alpha'), \end{aligned} \tag{A.6}$$

where

$$\alpha_0^* = \frac{N^2\nu' - (N')^2\nu}{N\nu' - N'\nu}, \qquad \alpha = \frac{1-3|\beta|^2}{1-|\beta|^2}, \qquad \alpha' = \frac{1-3|\beta'|^2}{1-|\beta'|^2}.$$

We note that the two traveling waves reduce to the same constant function when $\beta \to 0$ and $\beta' \to 0$, which is further evidence that a single sheet of nontrivial solutions connects these two families of traveling waves.

Appendix B. Proof of Theorem 3

As explained in Remark 4, x_0 and t_0 are spatial and temporal phase shifts, so we may set them to zero without loss of generality. We can express the solution directly in terms of the elementary symmetric functions via

$$\begin{aligned} u(x,t) &= \alpha_0 + \sum_{l=1}^{N} \phi(x;\beta_l(t)) = \alpha_0 + \sum_{l=1}^{N} 4\,\mathrm{Re}\left\{\sum_{k=1}^{\infty} \beta_l(t)^k e^{ikx}\right\} \\ &= \alpha_0 + \sum_{l=1}^{N} 4\,\mathrm{Re}\left\{\frac{z}{z-\beta_l(t)} - 1\right\} = \alpha_0 + 4\,\mathrm{Re}\left\{\frac{z\partial_z P(z)}{P(z)} - N\right\}, \ (z = e^{-ix}). \end{aligned} \tag{B.1}$$

(1) $N \ge 1,\ \nu \in \mathbb{Z},\ n \ge 1,\ m \ge 1$

(2) if $n < N$ then

- $m \in n\nu + N\mathbb{Z}$
- $\alpha_0 \le N + n(N-n)\nu/m$

(3) if $n \ge N$ then

- $m \in (n+1)\nu + N\mathbb{Z}$
- if $m > (n+1-N)\nu$ then $\alpha_0 \le N + (n+1-N)(n+1)\nu/m$
- if $m < (n+1-N)\nu$ then $\alpha_0 \ge N + (n+1-N)(n+1)\nu/m$
- if $m = (n+1-N)\nu$ then $\alpha_0 = n+1+N$

Table 1. Bifurcation rules governing which values of α_0 are compatible with the bifurcation indices (N, ν, n, m).

Next we derive algebraic expressions relating A, B, C, α_0, ω, N, N', ν and ν' by substituting (B.1) into the Benjamin–Ono equation (1). To this end, we include the time dependence of P in the notation and write (B.1) in the form

$$u(x,t)=\alpha_0+2\Big(\frac{i\partial_x g}{g}-N\Big)+2\Big(\frac{-i\partial_x h}{h}-N\Big), \tag{B.2}$$

where

$$g(x,t)=P(e^{-ix},e^{-i\omega t}), \qquad h(x,t)=\overline{g(x,t)}, \tag{B.3}$$

$$P(z,\lambda)=z^N+A\lambda^{\nu'}z^{N-N'}+B\lambda^{\nu-\nu'}z^{N'}+C\lambda^{\nu}. \tag{B.4}$$

Note that P is a polynomial in z and a Laurent polynomial in λ (as ν and ν' may be negative). We may assume $\omega>0$; if not, we can change the sign of ω without changing the solution by replacing (A,B,ν,ν',N') by $(B,A,-\nu,\nu'-\nu,N-N')$. Assuming the roots $\beta_l(t)$ of $z\mapsto P(z,e^{-i\omega t})$ remain inside the unit disk Δ for all t, we have

$$\Big(\frac{i\partial_x g}{g}-N\Big)=\sum_{l=1}^{N}\sum_{k=1}^{\infty}\beta_l(t)^k e^{ikx} \quad\Rightarrow\quad Hu=2\Big(\frac{\partial_x g}{g}+Ni\Big)+2\Big(\frac{\partial_x h}{h}-Ni\Big). \tag{B.5}$$

Using (B.2) and $\partial_t(\partial_x g/g)=\partial_x(\partial_t g/g)$, (a technique we learned by studying the bilinear formalism approach of [32; 26]), the equation $1/2\,(u_t-Hu_{xx}+uu_x)=0$ becomes

$$\partial_x\left[i\Big(\frac{\partial_t g}{g}-\frac{\partial_t h}{h}\Big)-\partial_x\Big(\frac{\partial_x g}{g}+\frac{\partial_x h}{h}\Big)+\frac{1}{4}\Big((\alpha_0-4N)+2i\Big(\frac{\partial_x g}{g}-\frac{\partial_x h}{h}\Big)\Big)^2\right]=0. \tag{B.6}$$

The expression in brackets must be a constant, which we denote by γ. We now write

$$P_{jk}=(z\partial_z)^j(\lambda\partial_\lambda)^k P(z,\lambda)\Big|_{\substack{z=e^{-ix}\\ \lambda=e^{-i\omega t}}} \tag{B.7}$$

so that, for example, $\partial_t g=-i\omega P_{01}$ and $\partial_x h=i\bar{P}_{10}$. Equation (B.6) then becomes

$$\gamma\, P_{00}\bar{P}_{00}+\bar{P}_{00}[P_{20}+\omega P_{01}+(\alpha_0-4N)P_{10}] \\ +P_{00}[\bar{P}_{20}+\omega\bar{P}_{01}+(\alpha_0-4N)\bar{P}_{10}]+2P_{10}\bar{P}_{10}=0, \tag{B.8}$$

where we have absorbed $\frac{1}{4}(\alpha_0-4N)^2$ into γ. This equation may be written

$$e_1[\![z^N\lambda^{-\nu}]\!]+e_2[\![z^{N-2N'}\lambda^{2\nu'-\nu}]\!]+e_3[\![z^{N-N'}\lambda^{\nu'-\nu}]\!]+e_4[\![z^{N'}\lambda^{-\nu'}]\!]+e_5=0,$$

where $[\![a]\!]=a+\bar{a}=2\,\mathrm{Re}\{a\}$,

$$e_1=[\gamma+\nu\omega+N^2+(\alpha_0-4N)N]C, \qquad e_2=[\gamma+\nu\omega+N^2+(\alpha_0-4N)N]AB,$$

and, after setting $\gamma = (3N-\alpha_0)N - \nu\omega$ to achieve $e_1 = e_2 = 0$,

$$e_3 = [(N')^2 - 2NN' + N'\alpha_0 - \nu'\omega]B + [(N')^2 + 2NN' - N'\alpha_0 + \nu'\omega]AC = 0, \quad \text{(B.9)}$$

$$\begin{aligned} e_4 = [3N^2 - 4NN' + (N')^2 - (N-N')\alpha_0 + (\nu-\nu')\omega]BC \\ -[N^2 - (N')^2 - (N-N')\alpha_0 + (\nu-\nu')\omega]A = 0, \end{aligned} \quad \text{(B.10)}$$

$$\begin{aligned} e_5 = (N\alpha_0 - \nu\omega - N^2) + [(2N'-N)\alpha_0 + (\nu - 2\nu')\omega + 3N^2 - 8NN' + 4(N')^2]B^2 \\ +[(N-2N')\alpha_0 + 4(N')^2 - N^2 + (2\nu'-\nu)\omega]A^2 + [(3N-\alpha_0)N + \nu\omega]C^2 = 0. \end{aligned} \quad \text{(B.11)}$$

Using a computer algebra system, it is easy to check that (B.9)–(B.11) hold when A, B, C, α_0 and ω are defined as in (62). When $\rho' = 0$, we have $A = B = 0$ and $C = \sqrt{\rho/(N+\rho)}$ so that

$$\beta_l(t) = \sqrt[N]{-C\lambda^\nu} = \sqrt[N]{-C}e^{-ict}, \qquad c = \frac{\omega\nu}{N} = \frac{N'(N-N')\nu}{m} = \alpha_0 - N\frac{1-3C^2}{1-C^2},$$

where each β_l is assigned a distinct N-th root of $-C$. By (5), this is an N-hump traveling wave with speed index ν and period $T = 2\pi/\omega$. Similarly, when $\rho = 0$, we have $B = C = 0$ and $A = \sqrt{\rho'/(N'+\rho')}$ so that

$$\beta_l(t) = \left\{ \begin{array}{cc} \sqrt[N']{-A}e^{-ict}, & l \le N' \\ 0, & l > N' \end{array} \right\},$$
$$c = \frac{\omega\nu'}{N'} = \frac{(N-N')(N+2\rho')\nu'}{m} = \alpha_0 - N'\frac{1-3A^2}{1-A^2},$$

which is an N'-hump traveling wave with speed index ν' and period $T = 2\pi/\omega$.

Finally, we show that the roots of $P(\cdot, \lambda)$ are inside the unit disk for any λ on the unit circle, S^1. We will use Rouché's theorem [1]. Let

$$\begin{aligned} f_1(z) &= z^N + A\lambda^{\nu'}z^{N-N'} + B\lambda^{\nu-\nu'}z^{N'} + C\lambda^\nu, \\ f_2(z) &= z^N + A\lambda^{\nu'}z^{N-N'}, \\ f_3(z) &= z^N \qquad\qquad\qquad + B\lambda^{\nu-\nu'}z^{N'}. \end{aligned}$$

From (62), we see that $\{A, B, C\} \subseteq [0, 1)$, $A \ge BC$, $B \ge CA$ and $C \ge AB$. Thus,

$$\begin{aligned} d_2(z) := |f_2(z)|^2 - |f_1(z) - f_2(z)|^2 = |\lambda^{-\nu'}z^{N'} + A|^2 - |B\lambda^{-\nu'}z^{N'} + C|^2 \\ = 1 + A^2 - B^2 - C^2 + 2(A - BC)\cos\theta \ge (1-A)^2 - (B-C)^2, \end{aligned} \quad \text{(B.12)}$$

where $\lambda^{-\nu'}z^{N'} = e^{i\theta}$. Similarly,

$$d_3 := |f_3(z)|^2 - |f_1(z) - f_3(z)|^2 \ge (1-B)^2 - (A-C)^2. \quad \text{(B.13)}$$

Note that

$$\begin{aligned}
&B \le A, \quad C \le B &&\Rightarrow\quad B - C \le B - AB < 1 - A &&\Rightarrow\quad d_2(z) > 0 \text{ for } z \in S^1,\\
&B \le A, \quad C > B &&\Rightarrow\quad |C - A| < 1 - B &&\Rightarrow\quad d_3(z) > 0 \text{ for } z \in S^1,\\
&A \le B, \quad C \le A &&\Rightarrow\quad A - C \le A - AB < 1 - B &&\Rightarrow\quad d_3(z) > 0 \text{ for } z \in S^1,\\
&A \le B, \quad C > A &&\Rightarrow\quad |C - B| < 1 - A &&\Rightarrow\quad d_2(z) > 0 \text{ for } z \in S^1.
\end{aligned}$$

Thus, in all cases, $f_1(z) = P(z, \lambda)$ has the same number of zeros inside S^1 as $f_2(z)$ or $f_3(z)$, which each have N roots inside S^1. Since $f_1(z)$ is a polynomial of degree N, all the roots are inside S^1.

References

[1] L. V. Ahlfors, *Complex analysis: An introduction to the theory of analytic functions of one complex variable*, 3rd ed., McGraw-Hill, New York, 1978. MR 80c:30001 Zbl 0395.30001

[2] D. M. Ambrose and J. Wilkening, *Time-periodic solutions of the Benjamin-Ono equation*, preprint, 2008. arXiv 0804:3623

[3] ———, *Computation of time-periodic solutions of the vortex sheet with surface tension*, in preparation, 2009.

[4] D. M. Ambrose, *Well-posedness of vortex sheets with surface tension*, SIAM J. Math. Anal. **35** (2003), no. 1, 211–244. MR 2005g:76006 Zbl 1107.76010

[5] C. J. Amick and J. F. Toland, *Uniqueness and related analytic properties for the Benjamin–Ono equation: a nonlinear Neumann problem in the plane*, Acta Math. **167** (1991), no. 1-2, 107–126. MR 92i:35099 Zbl 0755.35108

[6] T. B. Benjamin, *Internal waves of permanent form in fluids of great depth*, J. Fluid Mech. **29** (1967), no. 3, 559–592.

[7] T. L. Bock and M. D. Kruskal, *A two-parameter Miura transformation of the Benjamin–Ono equation*, Phys. Lett. A **74** (1979), no. 3-4, 173–176. MR 82d:35083

[8] M. O. Bristeau, R. Glowinski, and J. Périaux, *Controllability methods for the computation of time-periodic solutions; application to scattering*, J. Comput. Phys. **147** (1998), no. 2, 265–292. MR 99k:65093 Zbl 0926.65054

[9] C. G. Broyden, *The convergence of a class of double-rank minimization algorithms. II. The new algorithm*, J. Inst. Math. Appl. **6** (1970), 222–231. MR 55 #6841 Zbl 0207.17401

[10] M. Cabral and R. Rosa, *Chaos for a damped and forced KdV equation*, Phys. D **192** (2004), no. 3-4, 265–278. MR 2005d:37161 Zbl 1061.35103

[11] R. Camassa and L. Lee, *Complete integrable particle methods and the recurrence of initial states for a nonlinear shallow-water wave equation*, J. Comput. Phys. **227** (2008), no. 15, 7206–7221. MR 2009e:76019 Zbl 05304665

[12] K. M. Case, *Meromorphic solutions of the Benjamin–Ono equation*, Phys. A **96** (1979), no. 1-2, 173–182. MR 80i:35026

[13] ———, *The Benjamin–Ono equation: a remarkable dynamical system*, Ann. Nuclear Energy **7** (1980), no. 4-5, 273–277. MR 82j:76012

[14] G. J. Cooper and A. Sayfy, *Additive Runge–Kutta methods for stiff ordinary differential equations*, Math. Comp. **40** (1983), no. 161, 207–218. MR 84b:65066 Zbl 0525.65053

[15] R. E. Davis and A. Acrivos, *Solitary internal waves in deep water*, J. Fluid Mech. **29** (1967), no. 3, 593–607.

[16] S. Y. Dobrokhotov and I. M. Krichever, *Multi-phase solutions of the Benjamin-Ono equation and their averaging*, Mat. Zametki **49** (1991), 42–58, In Russian; translated in *Math. Notes* **49** (1991), 583–594. MR 92g:35182

[17] E. Doedel, H. B. Keller, and J.-P. Kernévez, *Numerical analysis and control of bifurcation problems, II. Bifurcation in infinite dimensions*, Internat. J. Bifur. Chaos Appl. Sci. Engrg. **1** (1991), no. 4, 745–772. MR 93c:34001b Zbl 0876.65060

[18] A. S. Fokas and M. J. Ablowitz, *The inverse scattering transform for the Benjamin–Ono equation: a pivot to multidimensional problems*, Stud. Appl. Math. **68** (1983), no. 1, 1–10. MR 84f:35139 Zbl 0505.76031

[19] G. Iooss, P. I. Plotnikov, and J. F. Toland, *Standing waves on an infinitely deep perfect fluid under gravity*, Arch. Ration. Mech. Anal. **177** (2005), no. 3, 367–478. MR 2007a:76017 Zbl 02222506

[20] D. J. Kaup and Y. Matsuno, *The inverse scattering transform for the Benjamin–Ono equation*, Stud. Appl. Math. **101** (1998), no. 1, 73–98. MR 2000e:34146 Zbl 1136.34349

[21] C. A. Kennedy and M. H. Carpenter, *Additive Runge–Kutta schemes for convection-diffusion-reaction equations*, Appl. Numer. Math. **44** (2003), no. 1-2, 139–181. MR 2003m:65111 Zbl 1013.65103

[22] R. J. LeVeque, *On the interaction of nearly equal solitons in the KdV equation*, SIAM J. Appl. Math. **47** (1987), no. 2, 254–262. MR 89c:35138 Zbl 0637.35078

[23] J. Marsden and A. Weinstein, *Reduction of symplectic manifolds with symmetry*, Rep. Mathematical Phys. **5** (1974), no. 1, 121–130. MR 53 #6633 Zbl 0327.58005

[24] Y. Matsuno, *Interaction of the Benjamin–Ono solitons*, J. Phys. A **13** (1980), no. 5, 1519–1536. MR 81d:35072 Zbl 0437.35062

[25] Y. Matsuno, *Note on the Bäcklund transformation of the Benjamin–Ono equation*, J. Phys. Soc. Japan **54** (1985), no. 1, 45–50. MR 86j:35140

[26] ———, *New representations of multiperiodic and multisoliton solutions for a class of nonlocal soliton equations*, J. Phys. Soc. Japan **73** (2004), no. 12, 3285–3293. MR 2005i:35237 Zbl 1066.35103

[27] K. R. Meyer, *Symmetries and integrals in mechanics*, Dynamical systems (Proc. Sympos., Univ. Bahia, Academic Press, New York, 1973, pp. 259–272. MR 48 #9760 Zbl 0293.58009

[28] A. Nakamura, *Bäcklund transform and conservation laws of the Benjamin–Ono equation*, J. Phys. Soc. Japan **47** (1979), no. 4, 1335–1340. MR 80m:35068

[29] J. Nocedal and S. J. Wright, *Numerical optimization*, Springer Series in Operations Research, Springer, New York, 1999. MR 2001b:90002 Zbl 0930.65067

[30] H. Ono, *Algebraic solitary waves in stratified fluids*, J. Phys. Soc. Japan **39** (1975), no. 4, 1082–1091. MR 53 #2129

[31] P. I. Plotnikov and J. F. Toland, *Nash–Moser theory for standing water waves*, Arch. Ration. Mech. Anal. **159** (2001), no. 1, 1–83. MR 2002k:76019 Zbl 1033.76005

[32] J. Satsuma and Y. Ishimori, *Periodic wave and rational soliton solutions of the Benjamin–Ono equation*, J. Phys. Soc. Japan **46** (1979), no. 2, 681–687.

[33] J. Stoer and R. Bulirsch, *Introduction to numerical analysis*, 3rd ed., Texts in Applied Mathematics, no. 12, Springer, New York, 2002. MR 2003d:65001 Zbl 1004.65001

[34] D. Viswanath, *Recurrent motions within plane Couette turbulence*, J. Fluid Mech. **580** (2007), 339–358. MR 2008f:76100 Zbl 05167861

[35] D. Viswanath, *The fractal property of the Lorenz attractor*, Phys. D **190** (2004), no. 1-2, 115–128. MR 2005a:37057 Zbl 1041.37013

[36] J. Wilkening, *An infinite branching hierarchy of time-periodic solutions of the Benjamin–Ono equation*, preprint, 2008. arXiv 0811.4209

[37] J. Wilkening, *An algorithm for computing Jordan chains and inverting analytic matrix functions*, Linear Algebra Appl. **427** (2007), no. 1, 6–25. MR 2008m:15034 Zbl 1132.47010

[38] J. Wilkening, *Math 228a: Numerical solution of differential equations*, lecture notes, 2007.

[39] C. Wulff, J. S. W. Lamb, and I. Melbourne, *Bifurcation from relative periodic solutions*, Ergodic Theory Dynam. Systems **21** (2001), no. 2, 605–635. MR 2002f:37088 Zbl 0986.37044

Received November 25, 2008. Revised July 12, 2009.

DAVID M. AMBROSE: ambrose@math.drexel.edu
Department of Mathematics, Drexel University, Philadelphia, PA 19104, United States

JON WILKENING: wilken@math.berkeley.edu
Department of Mathematics and Lawrence Berkeley National Laboratory, University of California, Berkeley, CA 94720, United States
http://math.berkeley.edu/∼wilken

COMM. APP. MATH. AND COMP. SCI.
Vol. 4, No. 1, 2009

A PHASE TRANSITION APPROACH TO DETECTING SINGULARITIES OF PARTIAL DIFFERENTIAL EQUATIONS

PANAGIOTIS STINIS

We present a mesh refinement algorithm for detecting singularities of time-dependent partial differential equations. The algorithm is inspired by renormalization constructions used in statistical mechanics to evaluate the properties of a system near a critical point, that is, a phase transition. The main idea behind the algorithm is to treat the occurrence of singularities of time-dependent partial differential equations as phase transitions.

The algorithm assumes the knowledge of an accurate reduced model. In particular, we need only assume that we know the functional form of the reduced model, that is, the terms appearing in the reduced model, but not necessarily their coefficients. We provide a way of computing the necessary coefficients on the fly as needed.

We show how the mesh refinement algorithm can be used to calculate the blow-up rate as we approach the singularity. This calculation can be done in three different ways: (i) the direct approach where one monitors the blowing-up quantity as it approaches the singularity and uses the data to calculate the blow-up rate; (ii) the "phase transition" approach (à la Wilson) where one treats the singularity as a fixed point of the renormalization flow equation and proceeds to compute the blow-up rate via an analysis in the vicinity of the fixed point, and (iii) the "scaling" approach (à la Widom–Kadanoff) where one postulates the existence of scaling laws for different quantities close to the singularity, computes the associated exponents and then uses them to estimate the blow-up rate. Our algorithm allows a unified presentation of these three approaches.

The inviscid Burgers and the supercritical focusing Schrödinger equations are used as instructive examples to illustrate the constructions.

Introduction

The problem of how to construct mesh refinement methods and how to approach more efficiently possible singularities of partial differential equations has attracted considerable attention [1; 3; 2; 6; 7; 14]. At the same time, the problem of

MSC2000: 35L67, 65M50, 65M70, 76B99.

Keywords: singularities, partial differential equations, mesh refinement, phase transition, renormalization, blow-up, dimension reduction.

constructing dimensionally reduced models for large systems of ordinary differential equations (this covers the case of partial differential equations after discretization or series expansion of the solution) has also received considerable attention; see, for example, the review papers [11; 10]. The construction of an accurate reduced model has advantages beyond the obvious one of predicting the correct behavior for a reduced set of variables.

We present here an algorithm that is based on dimensional reduction and which can be used to perform mesh refinement and investigate possibly singular solutions of partial differential equations (see also [16]). The algorithm is inspired by constructions used in statistical mechanics to evaluate the properties of a system near a critical point [12; 5] (a critical point is a value for the controlling parameter of a system at which the behavior of the system changes abruptly). The idea underlying the computation of the properties at criticality is that while the form of the reduced system equations is important, one can extract even more information by looking at how the form of the reduced system is related to the form of the original (full dimensional) system [17; 18]. We extend this idea to the study of (possibly) singular solutions of partial differential equations by treating time as the controlling parameter and the instant of occurrence of a singularity as a critical value for the parameter, that is, a critical point.

Our approach has two advantages: (i) it provides a way of accurately monitoring the progress of a simulation towards underresolution, thus offering as a byproduct the time of occurrence of the possible singularity; (ii) it allows the formulation of a mesh refinement scheme that is able to reach the equation's window of interesting dynamics much more efficiently than an algorithm that simply starts with the maximum available resolution.

The mesh refinement algorithm can be used to calculate the blow-up rate as we approach the singularity. This calculation can be done in three different ways: (i) the direct approach where one monitors the blowing-up quantity as it approaches the singularity and uses the data to calculate the blow-up rate; (ii) the "phase transition" approach (à la Wilson) [12] where one treats the singularity as a fixed point of the renormalization flow equation and proceeds to compute the blow-up rate via an analysis in the vicinity of the fixed point, and (iii) the "scaling" approach (à la Widom–Kadanoff) [5] where one postulates the existence of scaling laws for different quantities close to the singularity, computes the associated exponents and then uses them to estimate the blow-up rate. Our algorithm allows a unified presentation of these three approaches.

The task of investigating numerically the appearance of a singularity is subtle. Clearly, since all calculations are performed with finite resolution and a singularity involves an infinity of active scales we can only come as close to the singularity as our resolution will allow. On a related note, a partial differential equation may exhibit near-singular solutions, that is, solutions which involve a large but not infinite number of active scales. From the point of view of computation a near-singular solution may appear as a singular one if we cannot afford enough computational power to fully resolve the near-singular solution. This possibility should be kept in mind before deciding that a singular solution is indeed present. In other words, given adequate resolution we can eliminate the possibility of a singularity. But it may be very hard to prove through a finite calculation that a singularity exists (we come back to these points in Sections 2 and 3).

The paper is organized as follows. In Section 1 we present the ideas behind the construction of the algorithm. In Section 2 we present the mesh refinement algorithm. In Section 3 we provide numerical results for the inviscid Burgers equation. In Section 4 we provide numerical results for the supercritical focusing Schrödinger equation. Section 5 shows how one can use the mesh refinement algorithm to compute the blow-up rate as a critical exponent, i.e. using solely properties of a renormalization (coarse-graining) process in the vicinity of the singularity. Section 6 contains a discussion of the results and some directions for future work.

1. The main construction

Suppose that we are interested in the possible development of singularities in the solution $v(x,t)$ of a partial differential equation (PDE)

$$v_t + H(t, x, v, v_x, \dots) = 0,$$

where H is a (generally) nonlinear operator and $x \in D \subseteq \mathbb{R}^d$ (the constructions extend readily to the case of systems of partial differential equations). After spatial discretization or expansion of the solution in series, the PDE transforms into a system of ordinary differential equations (ODEs). For simplicity we restrict ourselves to the case of periodic boundary conditions, so that a Fourier expansion of the solution leads to a system of ODEs for the Fourier coefficients. To simulate the system for the Fourier coefficients, we need to truncate at some point the Fourier expansion. Let $F \cup G$ denote the set of Fourier modes retained in the series, where we have split the Fourier modes in two sets, F and G. We call the modes in F resolved and the modes in G unresolved. One can construct, in principle, an exact reduced model for the modes in F, for example, through the Mori–Zwanzig formalism [8] (we do not deal here with the complications of constructing a good reduced model).

The main idea behind the algorithm is that the evolution of moments of the reduced set of modes, for example l_p norms of the modes in F, should be the same whether computed from the full or the reduced system. This is a generalization to time-dependent systems of the principle used in the theory of equilibrium phase transitions to compute the critical exponents [12; 15]. The idea underlying the computation of the critical exponents is that while the form of the reduced system of equations is important, one can extract even more information by looking at how the form of the reduced system is related to the form of the original (full dimensional) system. We extend this idea to the study of (possibly) singular solutions of partial differential equations by treating time as the controlling parameter and the instant of occurrence of a singularity as a critical value for the parameter, that is, a critical point. We note that even though our motivation for the present construction came from the theory of equilibrium phase transitions, we do not advocate that a singularity is a phase transition in the conventional sense. It can be thought of as a transition from a strong solution to an appropriately defined weak solution but one does not have to push the analogy further. We want to point out here that the problem we are addressing is different from the subject known as dynamic critical phenomena [12, Chapter 8]. There, one is interested in the computation of time-dependent quantities as a controlling parameter, other than time, reaches its critical value. In our case, time *is* the controlling parameter and we are interested in the behavior of the solution as time reaches a critical value.

The above arguments can be made more precise. The original system of equations for the modes $F \cup G$ is given by

$$\frac{du(t)}{dt} = R(t, u(t)),$$

where $u = (\{u_k\})$, $k \in F \cup G$ is the vector of Fourier coefficients of u and R is the Fourier transform of the operator H. The system should be supplemented with an initial condition $u(0) = u_0$. The vector of Fourier coefficients can be written as $u = (\hat{u}, \tilde{u})$, where $\hat{u}$ are the resolved modes (those in F) and $\tilde{u}$ the unresolved ones (those in G). Similarly, for the right hand sides (RHS) we have $R(t, u) = (\hat{R}(t, u), \tilde{R}(t, u))$. Note that the RHS of the resolved modes involves both resolved and unresolved modes. In anticipation of the construction of a reduced model we can rewrite the RHS as $R(t, u) = R^{(0)}(t, u) = (\hat{R}^{(0)}(t, u), \tilde{R}^{(0)}(t, u))$. Recall that the main idea behind the current mesh refinement approach is to construct a reduced model for the modes in F and compare the evolution of these modes by the reduced model to the evolution of the same modes by the full system. In general, when one constructs a reduced model, additional terms appear on the RHS of the equations of the reduced model. The role of these additional terms is to account for the interactions between the resolved and unresolved modes, since the unresolved modes no longer appear explicitly in the reduced model. As is standard in renormalization theory [5], one can augment the RHS of the equations in the full system by including such additional terms. That is accomplished by multiplying each of these additional terms by a zero coefficient. In this way, the reduced and full systems' RHSs have the same functional form. In particular, for each mode u_k, $k \in F \cup G$, we can rewrite $R_k^{(0)}(t,u)$ as

$$R_k^{(0)}(t,u(t)) = \sum_{i=1}^{m} a_i^{(0)} R_{ik}^{(0)}(t,u(t)),$$

where $R_{1k}^{(0)}(t,u(t)) = R_k^{(0)}(t,u(t))$ and $R_{ik}^{(0)}(t,u(t))$, for $i = 2,\dots,m$ are of the same functional form as the additional terms which appear in the reduced model. This is easy to do by taking $a_1^{(0)} = 1$ and $a_i^{(0)} = 0$, for $i = 2,\dots,m$. Thus, the equation for the mode u_k, $k \in F \cup G$ is written as

$$\frac{du_k(t)}{dt} = R_k(t,u) = R_k^{(0)}(t,u(t)) = \sum_{i=1}^{m} a_i^{(0)} R_{ik}^{(0)}(t,u(t)). \tag{1}$$

Correspondingly, the reduced model for the mode u'_k, $k \in F$ is given by

$$\frac{du'_k(t)}{dt} = R_k^{(1)}(t,\hat{u}'(t)) = \sum_{i=1}^{m} a_i^{(1)} R_{ik}^{(1)}(t,\hat{u}'(t)), \tag{2}$$

with initial condition $u'_k(0) = u_{0k}$. We repeat that the functions $R_{ik}^{(1)}$, $i = 1,\dots,m$, $k \in F$, have the same form as the functions $R_{ik}^{(0)}$, $i = 1,\dots,m$, $k \in F$, but they depend *only* on the reduced set of modes F. Dimensional reduction transforms the vector $a^{(0)} = (a_1^{(0)},\dots,a_m^{(0)})$ to $a^{(1)} = (a_1^{(1)},\dots,a_m^{(1)})$. This allows one to determine the relation of the full to the reduced system by focusing on the change of the vector $a^{(0)}$ to $a^{(1)}$. Also, the vectors $a^{(0)}$ and $a^{(1)}$ do not have to be constant in time. This does not change the analysis that follows.

Define m quantities $\hat{E}_i$, $i = 1,\dots,m$, involving only modes in F. For example, these could be L_p norms of the reduced set of modes. To proceed we require that these quantities' rates of change be the same when computed from (1) and (2), i.e.,

$$\frac{d\hat{E}_i(\hat{u})}{dt} = \frac{d\hat{E}_i(\hat{u}')}{dt}, \qquad i = 1,\dots,m. \tag{3}$$

Note that similar conditions, albeit time-independent, lie at the heart of the renormalization group theory for equilibrium systems [5, page 154]. In fact, it is these conditions that allow the definition and calculation of the (renormalization) matrix whose eigenvalues are used to calculate the critical exponents. In the current (time-dependent) setting, the renormalization matrix is defined by differentiating $d\hat{E}_i(\hat{u})/dt$ with respect to $a^{(0)}$ and using (3) to obtain

$$\frac{\partial}{\partial a_j^{(0)}}\Big(\frac{d\hat{E}_i(\hat{u})}{dt}\Big) = \sum_{k=1}^{m} \frac{\partial}{\partial a_k^{(1)}}\Big(\frac{d\hat{E}_i(\hat{u}')}{dt}\Big)\frac{\partial a_k^{(1)}}{\partial a_j^{(0)}}, \quad i,j = 1,\dots,m. \tag{4}$$

We define the renormalization matrix $M_{kj} = \dfrac{\partial a_k^{(1)}}{\partial a_j^{(0)}}$, $k, j = 1, \dots, m$, and the matrices

$$A_{kj} = \frac{\partial}{\partial a_j^{(0)}} \Big(\frac{d\hat{E}_k(\hat{u})}{dt} \Big) \quad \text{and} \quad B_{kj} = \frac{\partial}{\partial a_j^{(1)}} \Big(\frac{d\hat{E}_k(\hat{u}')}{dt} \Big), \qquad k, j = 1, \dots, m.$$

Equation (4) can be written in matrix form as

$$A = MB. \tag{5}$$

The entries of A describe the contributions of the different terms appearing on the RHS of the full system to the rate of change of E_i. The same can be said for the entries of matrix B and the reduced model.

The eigenvalues of the matrix M contain information about the behavior of the reduced system relative to the full system. In fact, they measure whether the full and reduced systems deviate or approach. In the renormalization theory of critical phenomena, the eigenvalues of M at the critical point are used to analyze the system properties close to criticality. The analysis is based on the assumption that the eigenvalues of M change slowly near the critical point so that even if one cannot compute exactly *on* the critical point, it is possible to get an accurate estimate of them by computations near the critical point. Then, one performs a linear stability analysis near the fixed point and computes the system properties. The situation in the case of singularities of PDEs is different. In this case, the eigenvalues of M vary *most rapidly* near the singularity, due to the full system's rapid deterioration. Thus, we are not able to use linear stability analysis near the singularity. However, we are still able to extract the relevant blow-up rates (see Section 5).

1.1. ***An instructive example.*** We use the one-dimensional inviscid Burgers equation as an instructive example for the constructions presented in this section. The equation is given by

$$u_t + uu_x = 0. \tag{6}$$

This equation should be supplemented with an initial condition $u(x, 0) = u_0(x)$ and boundary conditions. We solve (6) in the interval $[0, 2\pi]$ with periodic boundary conditions. This allows us to expand the solution in Fourier series

$$u^M(x, t) = \sum_{k \in F \cup G} u_k(t) e^{ikx},$$

where $F \cup G = [-M/2, M/2 - 1]$. We have written the set of Fourier modes as the union of two sets in anticipation of the construction of the reduced model comprising only of the modes in $F = [-N/2, N/2 - 1]$, where $N < M$. The equation of motion for the Fourier mode u_k becomes

$$\frac{du_k}{dt} = -\frac{ik}{2} \sum_{\substack{p+q=k \\ p,q\in F\cup G}} u_p u_q. \tag{7}$$

1.1.1. *The t-model.* We need to choose a reduced model for the modes in F. We use a reduced model, known as the *t-model*, which follows correctly the behavior of the solution to the inviscid Burgers equation even after the formation of shocks [4; 13]. The t-model was first derived in the context of statistical irreversible mechanics [9] and was later analyzed in [4; 13]. It is based on the assumption of the absence of time scale separation between the resolved and unresolved modes. We will use the same model for the case with nonzero viscosity and comment on its validity when appropriate. For a mode u'_k in F the model is given by

$$\frac{d}{dt}u'_k = -\frac{ik}{2} \sum_{\substack{p+q=k \\ p\in F,\, q\in F}} u'_p u'_q - \frac{ik}{2} \sum_{\substack{p+q=k \\ p\in F,\, q\in G}} u'_p \Bigg[-t\frac{iq}{2} \sum_{\substack{r+s=q \\ r\in F,\, s\in F}} u'_r u'_s \Bigg] \\ - \frac{ik}{2} \sum_{\substack{p+q=k \\ p\in G,\, q\in F}} \Bigg[-t\frac{ip}{2} \sum_{\substack{r+s=p \\ r\in F,\, s\in F}} u'_r u'_s \Bigg] u'_q. \tag{8}$$

The first term on the RHS of (8) is of the same form as the first term in (7), except that the term in (8) is defined only for the modes in F. The viscous term is the same. The third and fourth terms in (8) are not present in (7). They are cubic in the Fourier modes and they are effecting the drain of energy out of the modes in F. We should note here that the cubic terms in the t-model do not depend on the viscosity. To conform with the notation introduced earlier we rewrite (8) as

$$\frac{d}{dt}u'_k = a_1^{(1)} \Bigg[-\frac{ik}{2} \sum_{\substack{p+q=k \\ p\in F,\, q\in F}} u'_p u'_q \Bigg] \\ + a_2^{(1)} \Bigg[-\frac{ik}{2} \sum_{\substack{p+q=k \\ p\in F,\, q\in G}} u'_p \Bigg[-t\frac{iq}{2} \sum_{\substack{r+s=q \\ r\in F,\, s\in F}} u'_r u'_s \Bigg] - \frac{ik}{2} \sum_{\substack{p+q=k \\ p\in G,\, q\in F}} \Bigg[-t\frac{ip}{2} \sum_{\substack{r+s=p \\ r\in F,\, s\in F}} u'_r u'_s \Bigg] u'_q \Bigg],$$

where $a_1^{(1)} = 1$ and $a_2^{(1)} = 1$. We rewrite Equation (7) as

$$\frac{du_k}{dt} = a_1^{(0)} \Bigg[-\frac{ik}{2} \sum_{\substack{p+q=k \\ p,q\in F\cup G}} u_p u_q \Bigg] + a_2^{(0)} \Bigg[-\frac{ik}{2} \sum_{\substack{p+q=k \\ p\in F\cup G,\, q\in I}} u_p \Bigg[-t\frac{iq}{2} \sum_{\substack{r+s=q \\ r\in F\cup G,\, s\in F\cup G}} u_r u_s \Bigg] \\ -\frac{ik}{2} \sum_{\substack{p+q=k \\ p\in I,\, q\in F\cup G}} \Bigg[-t\frac{ip}{2} \sum_{\substack{r+s=p \\ r\in F\cup G,\, s\in F\cup G}} u_r u_s \Bigg] u_q \Bigg],$$

where $a_1^{(0)} = 1$ and $a_2^{(0)} = 0$. The reader should note that we have introduced a new set of modes I. This is the set of unresolved modes for the *full* system. The reason for introducing the set I is that, as is the case in renormalization formulations, the terms appearing in the RHS of the equations at the different levels of resolution should be of the same functional form. The difference between the different levels of resolution should be only in the range of modes used. Since the t-model involves a quadratic convolution sum with one index in the resolved range and the other in the unresolved range, we should use the same functional form when constructing the corresponding term for the full system. Thus, this term should involve a convolution sum with one index in the range $F \cup G$ and the other in I.

Further, define

$$\hat{R}_{1k}^{(0)}(t, \hat{u}(t)) = -\frac{ik}{2} \sum_{\substack{p+q=k \\ p,q \in F \cup G}} u_p u_q,$$

$$\begin{aligned} &\hat{R}_{2k}^{(0)}(t, \hat{u}(t)) \\ &\quad = -\frac{ik}{2} \sum_{\substack{p+q=k \\ p \in F \cup G,\, q \in I}} u_p \Bigg[-t\frac{iq}{2} \sum_{\substack{r+s=q \\ r \in F \cup G,\, s \in F \cup G}} u_r u_s \Bigg] - \frac{ik}{2} \sum_{\substack{p+q=k \\ p \in I,\, q \in F \cup G}} \Bigg[-t\frac{ip}{2} \sum_{\substack{r+s=p \\ r \in F \cup G,\, s \in F \cup G}} u_r u_s \Bigg] u_q. \end{aligned}$$

Also, define

$$\hat{R}_{1k}^{(1)}(t, \hat{u}'(t)) = -\frac{ik}{2} \sum_{\substack{p+q=k \\ p,q \in F}} u'_p u'_q,$$

$$\begin{aligned} &\hat{R}_{2k}^{(1)}(t, \hat{u}'(t)) \\ &\qquad = -\frac{ik}{2} \sum_{\substack{p+q=k \\ p \in F,\, q \in G}} u'_p \Bigg[-t\frac{iq}{2} \sum_{\substack{r+s=q \\ r \in F,\, s \in F}} u'_r u'_s \Bigg] - \frac{ik}{2} \sum_{\substack{p+q=k \\ p \in G,\, q \in F}} \Bigg[-t\frac{ip}{2} \sum_{\substack{r+s=p \\ r \in F,\, s \in F}} u'_r u'_s \Bigg] u'_q. \end{aligned}$$

Thus, the equations of motion for the resolved modes in the full system and the reduced model can be written as

$$\frac{du_k}{dt} = \sum_{i=1}^{2} a_i^{(0)} \hat{R}_{ik}^{(0)}(t, u(t)) \tag{9}$$

and

$$\frac{du'_k}{dt} = \sum_{i=1}^{2} a_i^{(1)} \hat{R}_{ik}^{(1)}(t, \hat{u}'(t)). \tag{10}$$

To proceed, we need to define the quantities $\hat{E}_i$, $i = 1, \dots, m$. In our case, $m = 2$ and we need to define $\hat{E}_1$ and $\hat{E}_2$. The choice of the $\hat{E}_i$ is not unique. We chose for our experiments $\hat{E}_1 = \sum_{k\in F} |u_k|^2$ and $\hat{E}_2 = \sum_{k\in F} |u_k|^4$. The rates of change of the $\hat{E}_i$ are given for the full system by

$$\frac{d\hat{E}_1}{dt} = \sum_{k\in F} a_1^{(0)} 2\,\mathrm{Re}(\hat{R}_{1k}^{(0)}(t,\hat{u}(t))u_k^*) + a_2^{(0)} 2\,\mathrm{Re}(\hat{R}_{2k}^{(0)}(t,\hat{u}(t))u_k^*)$$

and

$$\frac{d\hat{E}_2}{dt} = \sum_{k\in F} a_1^{(0)} 2\,\mathrm{Re}(2\hat{R}_{1k}^{(0)}(t,\hat{u}(t))|u_k|^2 u_k^*) + a_2^{(0)} 2\,\mathrm{Re}(2\hat{R}_{2k}^{(0)}(t,\hat{u}(t))|u_k|^2 u_k^*),$$

where u_k^* is the complex conjugate of u_k. Similarly, for the reduced system we have

$$\frac{d\hat{E}_1}{dt} = \sum_{k\in F} a_1^{(1)} 2\,\mathrm{Re}(\hat{R}_{1k}^{(1)}(t,\hat{u}'(t))u_k'^*) + a_2^{(1)} 2\,\mathrm{Re}(\hat{R}_{2k}^{(1)}(t,\hat{u}'(t))u_k'^*)$$

and

$$\frac{d\hat{E}_2}{dt} = \sum_{k\in F} a_1^{(1)} 2\,\mathrm{Re}(2\hat{R}_{1k}^{(1)}(t,\hat{u}'(t))|u_k'|^2 u_k'^*) + a_2^{(1)} 2\,\mathrm{Re}(2\hat{R}_{2k}^{(1)}(t,\hat{u}'(t))|u_k'|^2 u_k'^*).$$

The equations for the rates of change of the $\hat{E}_i$ can be used for the computation of the 2×2 matrices A and B through the relations (4).

2. The mesh refinement algorithm

We continue our presentation with the mesh refinement algorithm. The construction in the previous section requires the exact knowledge of an accurate reduced model. This means the knowledge of *both* the functional form of the reduced model and the associated coefficient vector $a^{(1)}$. In fact, it is possible to relax this constraint by requiring the knowledge only of the functional form of the reduced model, that is, knowledge of the vector $\hat{R}^{(1)}$ but *not* of $a^{(1)}$. This can be considered as a time-dependent generalization of the Swendsen renormalization algorithm (for example, see the nice presentation in [5, Chapter 5]), even though here we do not have a statistical framework. The Swendsen algorithm is based on the observation that knowledge of *only* the functional form of the reduced model but not necessarily of the associated coefficient vector $a^{(1)}$ is enough for computing quantities of the reduced system. In particular, the matrix B can be calculated by using the resolved modes' values as computed from the full system.

As we have mentioned before, the entries of B describe the contributions of the different terms appearing on the RHS of the reduced system to the rate of change of E_i (the same for the entries of matrix A and the full model). The determinant of the matrix B measures whether there is need for the *reduced* system to transfer energy to smaller scales. The time instant when $\det B$ becomes nonzero, T_B, signals the onset of energy transfer from the modes in F to the modes in G. The determinant of the matrix A measures whether there is need for the *full* system to transfer energy to smaller scales. The time instant when $\det A$ becomes nonzero, T_A, signals the onset of underresolution of the full system. The time interval $[T_B, T_A)$ is our window of opportunity to refine the mesh, without losing accuracy and without wasting computational resources. We will use the value of $\det B$ as a criterion to decide when it is time to refine the mesh.

Note that if there exists a singularity, the interval $\Delta T = T_A - T_B$ will shrink to zero as we increase the resolution. The converse is not necessarily true. If ΔT appears to converge to zero as we increase the resolution does not mean that there certainly exists a singularity. Since all the calculations are finite, there is only a maximum resolution that we can afford. It may well be that an even larger, and presently unattainable, resolution could reveal that there is no singularity.

Mesh refinement algorithm.

(1) Choose a value for TOL. For this value of TOL run a mesh refinement calculation, starting, say, from N_{start} modes to N_{final} modes. For example, let $N_{\text{start}} = 32$ and double at each refinement until, say, $N_{\text{final}} = 256$ modes. Record the values of the quantities $\hat{E}_i$, $i = 1, \dots, m$ when $N = N_{\text{final}}$ and $|\det B| = TOL$. Let's call this simulation $S1$.

(2) For the same value of TOL run a calculation with $N_{\text{start}} = N_{\text{final}}$ modes (for the example $N_{\text{start}} = N_{\text{final}} = 256$). Record the values of the quantities $\hat{E}_i$, $i = 1, \dots, m$ when $|\det B| = TOL$. Let's call this simulation $S2$.

(3) Compare to within how many digits of accuracy the quantities $\hat{E}_i$, $i = 1, \dots, m$ computed from $S1$ and $S2$ agree. If the agreement is to within a specified accuracy, say five digits, then choose this value of TOL. If the agreement is in fewer digits, then decrease TOL (more stringent criterion) and repeat until agreement is met.

(4) Use the above decided value of TOL to perform a mesh refinement calculation with a larger magnification ratio, i.e. a larger value for the ratio $N_{\text{final}}/N_{\text{start}}$.

The agreement in digits of accuracy between $S1$ and $S2$ depends on the form of the terms chosen for the reduced model. Even though we do not know the coefficients of the reduced model, knowledge of the correct functional form of the terms can affect significantly the accuracy of the results. This situation is well known in the numerical study of critical exponents in equilibrium phase transitions; see [5, Chapter 5].

2.1. ***How to compute the coefficients of the reduced model.*** When we only know the functional form of the terms appearing in the reduced model but not their coefficients it is not possible to evolve a reduced system. We present a way of actually computing the coefficients of the reduced model as needed. If the quantities $\hat{E}_i$, $i = 1, \ldots, m$ are, for example, L_p norms of the Fourier modes, then we can multiply (2) with appropriate quantities and combine with (3) to get

$$\begin{aligned}
\frac{d\hat{E}_1(\hat{u})}{dt} &= \sum_{i=1}^{m} a_i^{(1)} \hat{U}_{i1}^{(1)}(t, \hat{u}(t)),\\
\frac{d\hat{E}_2(\hat{u})}{dt} &= \sum_{i=1}^{m} a_i^{(1)} \hat{U}_{i2}^{(1)}(t, \hat{u}(t)),\\
\cdots &= \cdots\\
\frac{d\hat{E}_m(\hat{u})}{dt} &= \sum_{i=1}^{m} a_i^{(1)} \hat{U}_{im}^{(1)}(t, \hat{u}(t)),
\end{aligned}$$

where $\hat{U}_{ij}^{(1)}$, $i, j = 1, \ldots, m$ are the new RHS functions that appear. Note that the RHS of the equations above does not involve primed quantities. The reason is that here the reduced quantities are computed by using the values of the resolved modes from the full system. The above system of equations is a linear system of equations for the vector of coefficients $a^{(1)}$. In fact, the matrix of the system is the transpose B^T of the matrix B. The linear system can be written as

$$B^T a^{(1)} = \boldsymbol{e}, \tag{11}$$

where $\boldsymbol{e} = (d\hat{E}_1(\hat{u})/dt, \ldots, d\hat{E}_m(\hat{u})/dt)$. This system of equations can provide us with the time evolution of the vector $a^{(1)}$.

The determination of coefficients for the reduced model through the system (11) is a time-dependent version of the method of moments. We specify the coefficients of the reduced model so that the reduced model reproduces the rates of change of a finite number of moments of the solution. This construction can actually be used as an adaptive way of determining a reduced model if one has access to experimental values of the rates of change of a finite number of moments. Suppose that we are conducting a real world experiment where we can compute the values of a finite number of moments on a coarse grid only. Then we can use the system (11) at predetermined instants to update a model defined on the coarse grid. Results of this construction will be presented elsewhere.

3. Numerical results for the inviscid Burgers equation

We present numerical results of the mesh refinement algorithm for the inviscid Burgers equation with the initial condition $u(x, 0) = \sin(x)$. This initial condition leads to a singularity forming at time $T_c = 1$. Figure 1 contains results about the time spent between refinement steps and the time reached with the maximum allowed

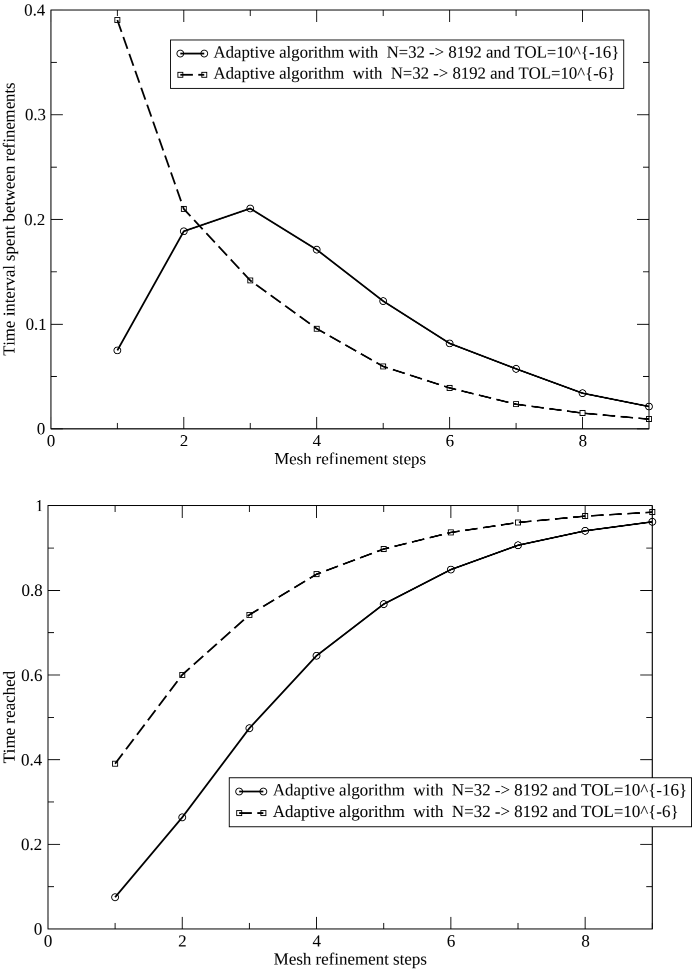


Figure 1. Top: Time spent between refinement steps for different tolerance values. Bottom: Time reached with the maximum allowed resolution.

resolution. We start from a resolution $N_{\text{start}} = 32$ and allow a maximum resolution of $N_{\text{final}} = 8192$. We present results for two values of the tolerance $TOL1 = 10^{-16}$ and $TOL2 = 10^{-6}$. When the tolerance criterion is less strict the algorithm can reach later times before running out of resolution.

In Figure 2 we compare the velocity field produced by the algorithm with $N_{\text{start}} = 32$, $N_{\text{final}} = 8192$ and $TOL1 = 10^{-16}$ with the velocity field produced by the algorithm with $N_{\text{start}} = N_{\text{final}} = 8192$ and the same tolerance. It is obvious that the results are in very good agreement. However, the mesh refinement calculation was about 240 times faster. The final time reached by the algorithm is $T = 0.962$.

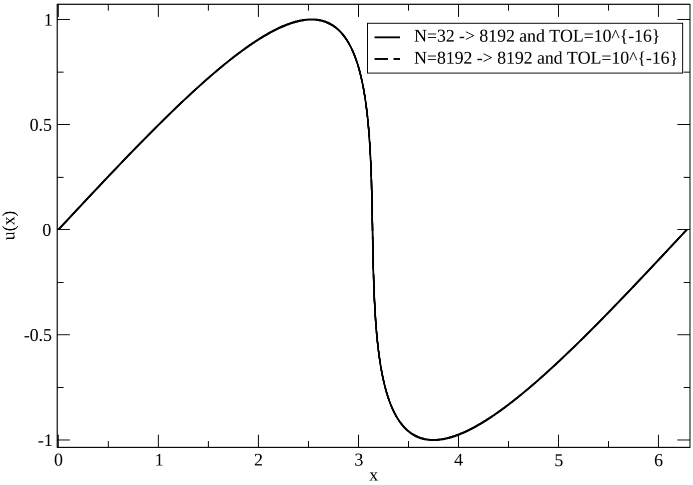


Figure 2. Comparison of the velocity field produced at the time of termination of the mesh refinement algorithm for two different magnification ratios. The first simulation has $N_{\text{start}} = 32$ and $N_{\text{final}g} = 8192$ while the second has $N_{\text{start}} = N_{\text{final}g} = 8192$.

3.1. ***The direct approach to calculating the blow-up rate.*** A mesh refinement algorithm can be used not only to approach a potential singularity but also estimate the rate at which the solution or some function of it blows-up. We restrict ourselves to the case of an algebraic (in time) singularity, meaning that some function of the solution diverges as $\sim |T_c - T|^{-\gamma}$, where $\gamma > 0$. Let us assume for a moment that T_c is known. One obvious way of estimating γ, is to run the mesh refinement algorithm and store the values of the blow-up quantity, say, ξ_n, $n = 1, \dots, N$, and the instant T_n at which each refinement took place. Then, one can plot (in log-log) the values of the blow-up quantity at the different refinement instants T_n as a function of the distance from the singularity $T_c - T_n$ and estimate the slope of the curve. That would provide us with the blow-up rate. Before we proceed, we have to address the issue of the value of T_c which is, in general, unknown. Thus, the value of T_c has to be calculated from the algorithm. It is simple to see that small errors in the estimation of T_c can lead to huge errors in the estimation of the blow-up rate. One way of estimating T_c is the following: for different choices of T_c, plot, in log-log coordinates, the values of the blow-up quantity at the refinement instants T_n as a function of the distance from the singularity $T_c - T_n$ and pick the value of T_c for which this plot is a straight line. This can be decided by monitoring the value of the correlation coefficient for a linear regression.

We present results of the above construction for the inviscid Burgers equation with the initial condition $u(x, 0) = \sin(x)$. This initial condition leads to a singularity forming at time $T_c = 1$. The maximum absolute value of the velocity gradient blows

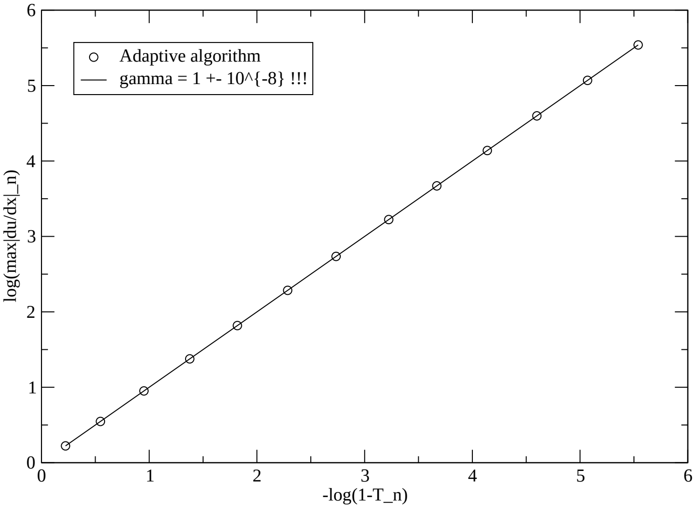


Figure 3. Log-log plot of the maximum absolute value of the velocity gradient $\max|\partial u/\partial x|_n$ and $(1-T_n)^{-1}$ for the different refinement steps (indexed by n).

up as $(1-T)^{-1}$. Figure 3 shows the log-log plot of the maximum absolute value of the velocity gradient $\max|\partial u/\partial x|_n$ and of the inverse distance from the singularity time $(1-T_n)^{-1}$ as recorded at the different refinement steps T_n. The slope of the curve is $\gamma = 1 \pm 10^{-8}$. Note that the minute error in the estimate shows that the refinement algorithm keeps the calculation well-resolved even very close to the singularity. The calculations were performed using the mesh refinement algorithm of Section 2 with the refinement tolerance criterion $TOL = \det B$ set to 10^{-10}. We should note that for this value of TOL, the value of $\det A$ for the full system is still much smaller than the double precision roundoff threshold of 10^{-16}. For this calculation we set $N_{\text{start}} = 32$ and $N_{\text{final}} = 131072$ and the algorithm terminated at time $T = 0.996$. The mesh refinement is about 3000 times faster than a calculation with $N_{\text{start}} = N_{\text{final}} = 131072$.

4. Numerical results for the supercritical focusing Schrödinger equation

We continue with numerical results about the supercritical focusing Schrödinger equation. The focusing Schrödinger equation is given by

$$i\,\partial u/\partial t + \Delta u + |u|^{2\sigma} u = 0, \quad \text{where } \sigma > 0. \tag{12}$$

The equation needs to be supplemented by an initial condition $u(x,0) = u_0(x)$ and boundary conditions. It has been conjectured by Zakharov [19] that in d dimensions, when $\sigma > 2/d$, and for a sufficiently large initial condition, the solution of (12) will blow-up at a finite time T, and the behavior of the solution close to the blow-up time is given by

$$u(x,t) = ((2\kappa(T-t))^{-1/2(1/\sigma+i\omega/\kappa)} Q((2\kappa(T-t))^{-1/2}|x|),$$

where $Q(\xi)$ is a complex-valued function with appropriate decay properties and κ and ω are parameters to be determined. For the maximum of the solution we have

$$\max |u(x,t)| \sim (T-t)^{-1/(2\sigma)} \quad \text{as } t \to T.$$

Although the mathematical theory is not yet complete, overwhelming evidence from numerical and formal analytical calculations suggests that the conjecture is true. Here, we restrict attention to the one-dimensional case and to periodic boundary conditions in the domain $[0, 2\pi]$. In the one-dimensional case, according to the conjecture, the solution exhibits an algebraic finite time blow-up when $\sigma > 2$. Here we present results for the case $\sigma = 3$. In the numerical experiments we used the initial condition

$$u_0(x,0) = iA\exp(-(x-\pi)^2),$$

for different values of A. For this initial condition we have $\max |u_0(x)| = A$ at $x = \pi$. Figure 4 shows the initial condition for $A = 1.35$ and the solution as computed by the mesh refinement algorithm with $N_{\text{start}} = 48$ and $N_{\text{final}} = 10368$. The tolerance criterion $TOL = \det B$ was set to 10^{-16}. The algorithm was implemented with the t-model for the reduced system as in the case of inviscid Burgers.

Table 1 contains the estimated blow-up exponents for the maximum of the solution for different values of A. For $A = 1.242$ the mesh refinement algorithm does not run out of resolution which signals the absence of a singularity. For all the other cases and for $N_{\text{start}} = 48$ and $N_{\text{final}} = 10368$, the mesh refinement algorithm was about 200 times faster than a calculation performed with $N_{\text{start}} = N_{\text{final}} = 10368$. Unlike the case of the inviscid Burgers equation, here we cannot estimate beforehand the exact time T of the blow-up. We do that in the way proposed in the previous section. In particular, for different choices of T, we calculated the correlation coefficient of the linear fit (in log-log coordinates), of the values of the blow-up quantity as a function of the distance from the singularity $T-t$ and picked the value of T for which the correlation coefficient is largest. For all the cases shown in Table 1 the correlation coefficient is about 0.999999999. The algorithm is able to approach the estimated singularity instant T to within 5×10^{-5} units of time. The conjectured blow-up exponent for the maximum of the solution when $\sigma = 3$ is $1/2\sigma = 1/6 \sim 0.1667$. The relative deviation of the estimated values of the exponent relative to the conjectured value of the exponent is within 1 percent for all the values of A examined except for $A = 1.8$ and $A = 2$.

$\max \lvert u_0(x) \rvert$	Est. exp. α	$\lvert$Rel. dev.$\rvert$ (%)	Sing. form
1.242	—	—	—
1.243	0.1652	0.90	$(T-t)^{-\alpha}$
1.250	0.1648	1.14	
1.255	0.1654	0.78	
1.260	0.1649	1.08	
1.300	0.1655	0.72	
1.350	0.1662	0.30	
1.500	0.1678	0.66	
1.600	0.1691	1.44	
1.800	0.1727 (0.1684)	3.60 (1.01)	
2.000	0.1766 (0.1696)	5.93 (1.74)	

Table 1. Estimated blow-up exponents for the supercritical ($\sigma = 3$) Schrödinger equation. The relative deviation is from the conjectured value of $1/2\sigma = 1/6 \sim 0.1667$.

We would like to make a comment about the discrepancy for $A = 1.8$ and $A = 2$. It is to be expected that if one keeps the same maximum resolution while increasing the magnitude of the initial condition, i.e. the value of A, after some value of A the algorithm runs out of resolution before it can come close enough to the singularity for the asymptotic behavior to settle in. To elucidate this point we also ran the mesh refinement algorithm with $N_{\text{final}} = 34992$ for $A = 1.8$ and $A = 2$. The estimated values of the blow-up exponent are included in Table 1 in parentheses. As we see, if one uses large enough resolution, the relative deviation of the estimated values of the exponent relative to the conjectured value of the exponent decreases again to within 1 percent. Note that for $N_{\text{start}} = 48$ and $N_{\text{final}} = 34992$ the mesh refinement algorithm is about 400 times faster than a calculation with $N_{\text{start}} = N_{\text{final}} = 34992$. As expected, the acceleration factor increases when $N_{\text{final}}/N_{\text{start}}$ increases.

5. Calculation of the blow-up rate as a critical exponent

As we have said, we are also interested in showing how the blow-up rate estimate can be obtained using properties of a renormalization flow, that is, a coarse-graining process. There are two ways to do that: (i) Wilson's or "phase transition" approach, where one treats the singularity as a fixed point of a renormalization transformation and computes the blow-up rate by analysis in the vicinity of the fixed point, and (ii) the Widom–Kadanoff or "scaling approach", where one assumes the existence of certain scaling laws in the vicinity of the singularity and then combines them to obtain the blow-up rate.

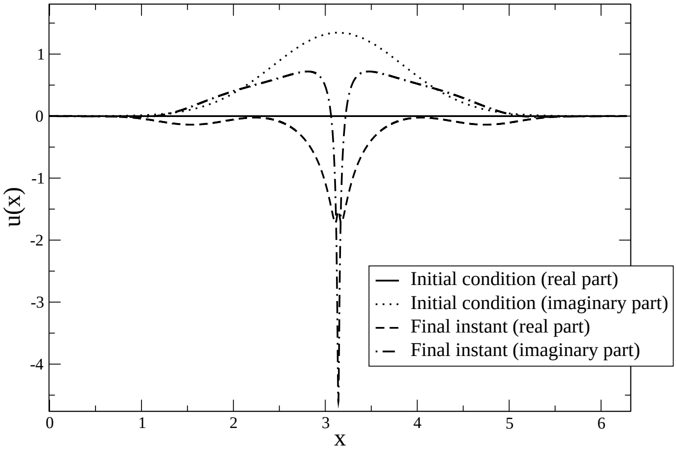


Figure 4. Supercritical ($\sigma = 3$) Schrödinger equation with $\max |u_0(x)| = 1.35$.

5.1. ***The "phase transition" approach.*** The key idea is that a series of successive refinement steps (going to smaller and smaller scales) can be seen (approximately) as a coarse-graining process in reverse. Thus, one can run the mesh refinement algorithm, compute and store the coefficients of the reduced model at each refinement step and then use them to reconstruct the renormalization flow from smaller to larger scales. In this case, the smallest scale that the refinement algorithm reached is the starting scale of the renormalization flow. For the case of a time-dependent PDE the mesh refinement algorithm allows us to get closer and closer to the singularity instant T_c. Thus, the renormalization procedure will take us further and further away from T_c.

There are two ways to show how the renormalization flow can be used to compute the blow-up rate. The first, the "phase transition" approach, assumes that the phase transition is a fixed point of the renormalization flow and proceeds with an analysis near the fixed point [5, pages 124–27]. However, as we mentioned in the discussion following (5), we do not use a linear stability analysis because the eigenvalues of M vary most rapidly near the fixed point. Instead, we deal with the full (nonlinear) renormalization flow.

The second way, the "scaling" approach, is just a manipulation of different scaling laws assumed to hold asymptotically near the singularity. Of course, both lead to the same expression for the blow-up rate. We choose to present both since it elucidates further the connection between the techniques presented in this paper and those used in the theory of equilibrium phase transitions.

We start our presentation of the blow-up rate calculation with the phase transition approach [5]. Let us suppose that near the singularity instant T_c a quantity ξ

behaves as $|T_c - T|^{-\gamma}$. For the case of Burgers this would be the maximum of the velocity gradient, that is, max $|\partial u/\partial x|$. We want to find the value of γ. As we have said we assume that we have computed and stored a sequence of coefficients for the reduced model, the associated length scale, the value of the blow-up quantity and the time of occurrence of the refinement step. Then, by simply reversing the sequence indexing, we have the necessary quantities for the description of a renormalization flow which starts close to T_c and moves further away with every coarse-graining step. Since every renormalization step brings us further away from the critical point T_c, the values of the blow-up quantity become smaller with every renormalization step. Thus, if we coarse-grain the length scale at which we probe the problem by a factor of b at each step (where $b > 1$), then $\xi_{n+1} = \xi_n/b^{\beta_2}$, with $\beta_2 > 0$. This implies $\xi_n \sim l_n^{-\beta_2}$ and thus β_2 can be computed from the refinement algorithm data collected. The coefficient of the reduced model which monitors the deviation of the full and reduced model will increase with each renormalization step, that is, $\alpha_{n+1} = \alpha_n b^{\beta_1}$, with $\beta_1 > 0$. This implies $\alpha_n \sim l_n^{\beta_1}$ and β_1 can also be computed from the collected data. Moreover, repeated application of the recursive relation for the coefficient α_n gives $\alpha_n = \alpha_0 (b^{\beta_1})^n$. This relation is the analog of the recursive relation derived in the theory of phase transitions by linearization of the renormalization flow around the critical (fixed) point. Here we did *not* resort to a linearization procedure. To proceed, we need to estimate the behavior of α_0, the starting point of the renormalization flow. In the theory of phase transitions, the behavior of the coefficient α_0 is assumed to be linear in $|T_c - T|$. However, there is no a priori reason for such a behavior. We assume that $\alpha_0 = C_2|T_c - T|^{\delta}$, where δ can also be computed from the collected data.

Let us summarize what we have obtained so far. As we renormalize, the blow-up quantity decreases and the reduced model coefficient that monitors the deviation of the full and reduced model increases. Following the phase transition approach we thus assume that if we take enough renormalization steps then we have

$$\frac{\xi}{C_1 (b^{\beta_2})^n} = u \quad \text{and} \quad C_2|T_c - T|^{\delta}(b^{\beta_1})^n = v,$$

where u, v are quantities of the same order and C_1, C_2 are constants that depend on the initial conditions. We can eliminate n in the above two relations and get

$$\xi \sim |T_c - T|^{-\gamma}, \quad \text{with } \gamma = \frac{\delta \beta_2}{\beta_1}.$$

Thus, we have expressed the blow-up rate exponent γ as a function of scaling exponents that are associated with properties of the renormalization flow.

Before we conclude with this approach, we need to make one more comment. We have said before that the phase transition approach treats the singularity as a fixed point of the renormalization flow. To do that one has to construct a differential equation for the evolution of the coefficient α with respect to l. Note that by the way we have defined it, α is dimensionless. The equation for its evolution with changes in l is given by $l\,\partial\alpha/\partial l = \beta(\alpha)$ [5]. The RHS of the equation is called the beta function and its zeros determine the fixed points of the renormalization flow. Since $\alpha = Cl^{\beta_1}$, for some constant C, we have $l\,\partial\alpha/\partial l = C\beta_1 l^{\beta_1}$. So, the beta function is $\beta(\alpha) = C\beta_1 l^{\beta_1} = \beta_1\alpha$. So, the only fixed point of the beta function is $\alpha = 0$. If $\beta_1 > 0$ then $\alpha = 0$ corresponds to $l = 0$, that is, the zero scale. But this is exactly the active scale reached at the instant that the singularity occurs. So, the singularity is indeed a fixed point of the renormalization flow as long as $\beta_1 > 0$. Moreover, if $\beta_1 > 0$, this fixed point is unstable, so that if we start close to it, the renormalization flow will take us further away. This is indeed the case for the Burgers equation, as we show numerically in the next section.

This concludes the phase transition approach.

5.2. ***The "scaling" approach.*** We conclude with the "scaling" approach which is based on direct combination of the different scaling laws associated with the renormalization flow. Indeed, let $\xi \sim |T_c - T|^{-\gamma'}$, where γ' is the blow-up rate exponent to be estimated. If we assume that near T_c we have $\xi \sim l^{-\beta_2}$, $\alpha \sim l^{\beta_1}$ and $\alpha \sim |T_c - T|^{\delta}$, we can use the renormalization flow to estimate β_1, β_2 and δ. Then a straightforward combination of the three scaling laws leads to $\gamma' = \delta\beta_2/\beta_1$. So, $\gamma' = \gamma$ and as expected this approach leads to the same expression for the blow-up rate exponent as the phase transition approach.

Figures 5-7 show how one can use the above construction to estimate the blow-up rate γ from renormalization flow quantities. Recall that the coefficient of the reduced model that monitors the deviation of the reduced and full systems is $a_2^{(n)}$. Also, that the index n appearing in the figures is used now to count the renormalization steps which are the opposite of the refinement steps. The length scale l_n at which we probe the system for the different renormalization steps is the length scale of the reduced model. This means that if we have a full system calculation with N_n modes, then $l_n = 22\pi/N_n$, since the reduced model has half the resolution of the full system.

From the data we estimate the exponents

$$\beta_2 = 0.670 \pm 0.001, \quad \beta_1 = 0.739 \pm 0.007, \quad \delta = 1.1026 \pm 10^{-8}.$$

From these estimates we get $\gamma' = 1 \pm 0.01$. Thus, when we compute the blow-up rate using solely renormalization flow quantities, the estimation error is larger than when computing this rate directly. This is to be expected since we had to combine three empirically determined scaling laws, each one of which comes with its own error and also relies entirely on the adequacy of the reduced model. Nevertheless,

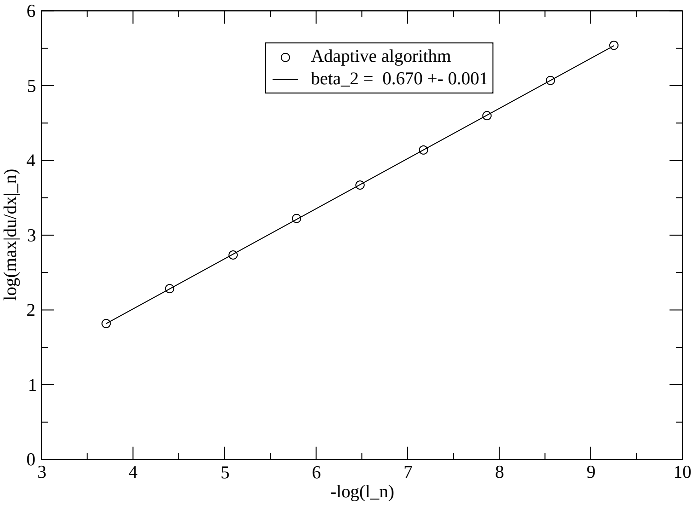


Figure 5. Log-log plot of the maximum absolute velocity gradient max $|\partial u/\partial x|_n$ and the inverse length scale of the reduced system l_n^{-1} for the different renormalization steps (indexed by n).

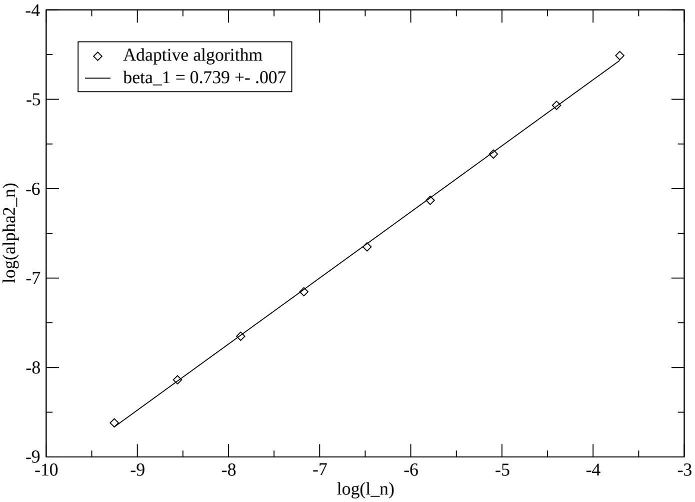


Figure 6. Log-log plot of the coefficient $a_2^{(n)}$ of the t-model and the length scale of the reduced system l_n for the different renormalization steps (indexed by n).

the accuracy obtained is acceptable and moreover, it highlights the accuracy of the t-model for this equation.

Finally, since $\beta_1 = 0.739 > 0$, we conclude that the singularity is an unstable fixed point of the renormalization flow (see discussion at the end of Section 5.1).

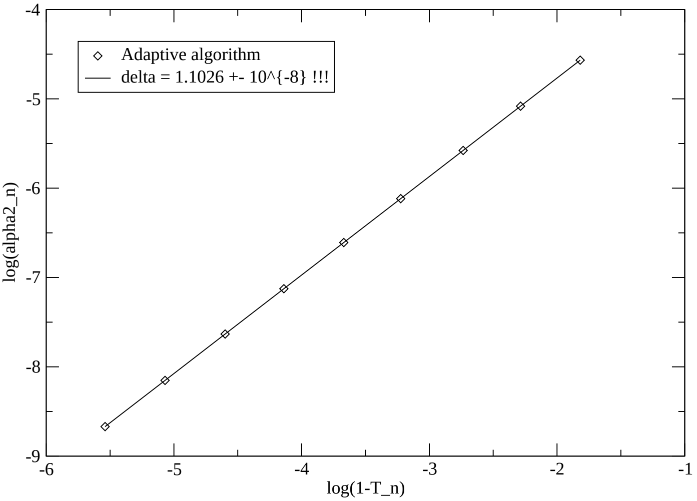


Figure 7. Log-log plot of the coefficient $a_2^{(n)}$ of the t-model and $1 - T_n$ for the different renormalization steps (indexed by n).

6. Conclusions and future work

We have presented a mesh refinement algorithm, inspired by renormalization constructions in critical phenomena, which allows the efficient location and approach of a possible singularity. The algorithm assumes knowledge of an accurate reduced model. In particular, it assumes knowledge of the functional form of the reduced model but not of the actual coefficients. We provide a way of computing the necessary coefficients on the fly as needed. On a theoretical level, the algorithm can be used to study the behavior of (near-) singular solutions. On the practical side, it can be used as a mesh refinement tool.

We have only examined the simple case of periodic boundary conditions under uniform mesh refinement. We plan to extend the constructions presented here to a real space formulation allowing the treatment of nonperiodic boundary conditions and more complicated geometries. In that case, one can divide the domain into subdomains and apply the mesh refinement algorithm individually in the different subdomains. In addition, the algorithm can be modified to perform mesh-coarsening after the computationally intensive time interval of the simulation has passed.

The original motivation behind the development of the algorithm was the open problem of the formation of singularities in finite time for the incompressible Euler and Navier–Stokes equations of fluid mechanics. In addition to helping with the issue of singularity formation, we hope that the algorithm can be of use in the simulation of real world flows by allowing a better assessment of the onset of underresolution.

Acknowledgments

I am grateful to Professors G. I. Barenblatt, A. J. Chorin and O. H. Hald for their ongoing guidance and support. I would like to thank Prof. V. Sverak for helpful discussions and Professors S. Weinberg and K. Wilson for inspiration.

References

[1] Ann S. Almgren, John B. Bell, Phillip Colella, Louis H. Howell, and Michael L. Welcome, *A conservative adaptive projection method for the variable density incompressible Navier–Stokes equations*, J. Comput. Phys. **142** (1998), no. 1, 1–46. MR 99k:76096 Zbl 0933.76055

[2] M. Berger and P. Colella, *Local adaptive mesh refinement for shock hydrodynamics*, J. Comp. Phys. **82** (1989), 62–84.

[3] M. Berger and R. V. Kohn, *A rescaling algorithm for the numerical calculation of blowing-up solutions*, Comm. Pure Appl. Math. **41** (1988), no. 6, 841–863. MR 89g:65154 Zbl 0652.65070

[4] David Bernstein, *Optimal prediction of Burgers's equation*, Multiscale Model. Simul. **6** (2007), no. 1, 27–52. MR 2008b:76034 Zbl 1135.65373

[5] J. Binney, N. Dowrick, A. Fisher, and M. Newman, *The theory of critical phenomena (an introduction to the renormalization group)*, The Clarendon Press, Oxford, 1992. Zbl 0771.00009

[6] Chris J. Budd, Weizhang Huang, and Robert D. Russell, *Moving mesh methods for problems with blow-up*, SIAM J. Sci. Comput. **17** (1996), no. 2, 305–327. MR 96j:65094 Zbl 0860.35050

[7] H. D. Ceniceros and Hou T. Y., *An efficient dynamically adaptive mesh for potentially singular solutions*, J. Comp. Phys. **172** (2001), 609–639. Zbl 0986.65087

[8] Alexandre J. Chorin, Ole H. Hald, and Raz Kupferman, *Optimal prediction and the Mori–Zwanzig representation of irreversible processes*, Proc. Natl. Acad. Sci. USA **97** (2000), no. 7, 2968–2973. MR 2000m:82045 Zbl 0968.60036

[9] ______, *Optimal prediction with memory*, Phys. D **166** (2002), no. 3-4, 239–257. MR 2003e:62150 Zbl 1017.60046

[10] Alexandre J. Chorin and P. Stinis, *Problem reduction, renormalization, and memory*, Commun. Appl. Math. Comput. Sci. **1** (2006), 1–27. MR 2007f:82092 Zbl 1108.82023

[11] Dror Givon, Raz Kupferman, and Andrew Stuart, *Extracting macroscopic dynamics: model problems and algorithms*, Nonlinearity **17** (2004), no. 6, R55–R127. MR 2006i:82081 Zbl 1073.82038

[12] N. Goldenfeld, *Lectures on phase transitions and the renormalization group*, Perseus Books, Reading, MA, 1992.

[13] O. H. Hald and P. Stinis, *Optimal prediction and the rate of decay for solutions of the Euler equations in two and three dimensions*, Proc. Natl. Acad. Sci. USA **104** (2007), no. 16, 6527–6532. MR 2008e:76100 Zbl 1155.76036

[14] M. J. Landman, G. C. Papanicolaou, C. Sulem, and P. Sulem, *Rate of blowup for solutions of the nonlinear Schrödinger equation at critical dimension*, Phys. Rev. A **38** (1988), 3837–3847.

[15] P. Stinis, *A maximum likelihood algorithm for the estimation and renormalization of exponential densities*, J. Comput. Phys. **208** (2005), no. 2, 691–703. MR 2144736 Zbl 1075.65017

[16] ______, *Dimensional reduction as a tool for mesh refinement and tracking singularities of pdes*, preprint, 2007. arXiv 0706.2895

[17] S. Weinberg, *Why the renormalization group is a good thing*, Asymptotic realms of physics: essays in honor of Francis E. Low (Alan H. Guth, Kerson Huang, and Robert L. Jaffe, eds.), MIT Press, Cambridge, MA, 1983, pp. 1–19.

[18] Kenneth G. Wilson, *The renormalization group and critical phenomena*, Rev. Modern Phys. **55** (1983), no. 3, 583–600. MR 84m:82008

[19] V. E. Zakharov, *Collapse of self-focusing langmuir waves: handbook of plasma physics, 2*, North Holland, Amsterdam, 1984.

Received June 13, 2009. Revised November 1, 2009.

PANAGIOTIS STINIS: stinis@math.umn.edu
Department of Mathematics, University of Minnesota, Minneapolis, MN 55455, United States
http://www.math.umn.edu/pacim/stinis.html

Communications in Applied Mathematics and Computational Science

vol. 4 no. 1 2009