

*Communications in
Applied
Mathematics and
Computational
Science*

vol. 9 no. 1 2014

Communications in Applied Mathematics and Computational Science

msp.org/camcos

EDITORS

MANAGING EDITOR

John B. Bell
Lawrence Berkeley National Laboratory, USA
jbbell@lbl.gov

BOARD OF EDITORS

Marsha Berger	New York University berger@cs.nyu.edu	Ahmed Ghoniem	Massachusetts Inst. of Technology, USA ghoniem@mit.edu
Alexandre Chorin	University of California, Berkeley, USA chorin@math.berkeley.edu	Raz Kupferman	The Hebrew University, Israel raz@math.huji.ac.il
Phil Colella	Lawrence Berkeley Nat. Lab., USA pcolella@lbl.gov	Randall J. LeVeque	University of Washington, USA rjl@amath.washington.edu
Peter Constantin	University of Chicago, USA const@cs.uchicago.edu	Mitchell Luskin	University of Minnesota, USA luskin@umn.edu
Maksymilian Dryja	Warsaw University, Poland maksymilian.dryja@acn.waw.pl	Yvon Maday	Université Pierre et Marie Curie, France maday@ann.jussieu.fr
M. Gregory Forest	University of North Carolina, USA forest@amath.unc.edu	James Sethian	University of California, Berkeley, USA sethian@math.berkeley.edu
Leslie Greengard	New York University, USA greengard@cims.nyu.edu	Juan Luis Vázquez	Universidad Autónoma de Madrid, Spain juanluis.vazquez@uam.es
Rupert Klein	Freie Universität Berlin, Germany rupert.klein@pik-potsdam.de	Alfio Quarteroni	Ecole Polytech. Féd. Lausanne, Switzerland alfio.quarteroni@epfl.ch
Nigel Goldenfeld	University of Illinois, USA nigel@uiuc.edu	Eitan Tadmor	University of Maryland, USA etadmor@cscamm.umd.edu
		Denis Talay	INRIA, France denis.talay@inria.fr

PRODUCTION

production@msp.org

Silvio Levy, Scientific Editor

See inside back cover or msp.org/camcos for submission instructions.

The subscription price for 2014 is US \$75/year for the electronic version, and \$105/year (+\$15, if shipping outside the US) for print and electronic. Subscriptions, requests for back issues from the last three years and changes of subscribers address should be sent to MSP.

Communications in Applied Mathematics and Computational Science (ISSN 2157-5452 electronic, 1559-3940 printed) at Mathematical Sciences Publishers, 798 Evans Hall #3840, c/o University of California, Berkeley, CA 94720-3840, is published continuously online. Periodical rate postage paid at Berkeley, CA 94704, and additional mailing offices.

CAMCoS peer review and production are managed by EditFLOW® from Mathematical Sciences Publishers.

PUBLISHED BY

 **mathematical sciences publishers**
nonprofit scientific publishing

<http://msp.org/>

© 2014 Mathematical Sciences Publishers

DISCRETE NONHOMOGENEOUS AND NONSTATIONARY LOGISTIC AND MARKOV REGRESSION MODELS FOR SPATIOTEMPORAL DATA WITH UNRESOLVED EXTERNAL INFLUENCES

JANA DE WILJES, LARS PUTZIG AND ILLIA HORENKO

Dynamical systems with different characteristic behavior at multiple scales can be modeled with hybrid methods combining a discrete model (e.g., corresponding to the microscale) triggered by a continuous mechanism and vice versa. A data-driven black-box-type framework is proposed, where the discrete model is parametrized with adaptive regression techniques and the output of the continuous counterpart (e.g., output of partial differential equations) is coupled to the discrete system of interest in the form of a fixed exogenous time series of external factors. Data availability represents a significant issue for this type of coupled discrete-continuous model, and it is shown that missing information/observations can be incorporated in the model via a nonstationary and nonhomogeneous formulation. An unbiased estimator for the discrete model dynamics in presence of unobserved external impacts is derived and used to construct a data-based nonstationary and nonhomogeneous parameter estimator based on an appropriately regularized spatiotemporal clustering algorithm. One-step and long-term predictions are considered, and a new Bayesian approach to discrete data assimilation of hidden information is proposed. To illustrate our method, we apply it to synthetic data sets and compare it with standard techniques of the machine-learning community (such as maximum-likelihood estimation, artificial neural networks and support vector machines).

1. Introduction

Discrete/categorical dynamical processes with a finite state space represent a challenge for standard data-based analysis tools. Heterogeneity of model properties over time and space as well as the discreteness of the data complicate the employment of standard time-series analysis techniques. Moreover, parametrization of the underlying process is often hampered by incompleteness of observational data.

Illia Horenko is the corresponding author.

MSC2010: primary 62-07, 62H30, 62M05, 62M10, 65C60; secondary 62M02, 62M20, 62M30, 62M45, 62H11.

Keywords: nonstationary, nonhomogeneous, discrete spatiotemporal time-series analysis, Markov regression, logistic, data assimilation.

In this paper, we want to address these problems by introducing a nonstationary, nonhomogeneous regression framework that allows taking a lack of observed information into account.

Adequate modeling and proper statistical handling of discrete processes (e.g., jump processes) is especially important for the proper description of multiscale dynamical systems. A typical modeling approach to multiscale dynamical systems is based on the employment of hybrid models, consisting of continuous and discrete model components [19; 20; 21]. While the continuous dynamics can be described with suitable PDEs, the discrete model can be estimated with appropriate data-based analysis methods. Communication between the two models can be achieved via incorporating the continuous data components (e.g., the output of PDEs or ODEs) as external statistical impact factors (or *covariates*) in the discrete part of the model.

Regression analysis [11] or pattern-recognition techniques such as artificial neural networks (ANN) [2; 24] or support vector machines (SVM) [8; 35] are popular instruments to approach the parametrization of dynamical processes. A common ansatz to model discrete-, categorical- and jump-processes is to deploy discrete choice models (e.g., logit or probit regression), which belong to the family of generalized linear models (GLM) [12; 10]. However, these classical techniques are usually restricted to time-independent model parameters, i.e., stationary models.

In this manuscript, we propose a nonstationary logistic regression model and also provide a direct approach to the discrete structure in the form of a nonstationary Markov regression. The key advantage of the proposed framework is that it allows us to parametrize the considered dynamical system corresponding to the data while taking all external influences into account, even those not explicitly available in the form of observation data. This is achieved by introducing an explicit dependency of the model parameters on time and location, i.e., by including an explicit temporal nonstationarity and spatial nonhomogeneity into the resulting model. Necessary assumptions and details will be given in [Proposition 2.1](#). A new numerical algorithm for the solution of the obtained inverse problem is formulated, and its numerical complexity is compared with the complexities of the standard algorithms of discrete data analysis. An adapted version of Akaike's information criterion is used to determine the best model fit corresponding to the data [30]. The resulting optimal parameters can then be employed to make predictions about future states of the process. In this context, a Bayesian approach to assimilate new hidden information (describing the impact of unresolved external factors) is proposed. Training and testing of the techniques are done on several sets of synthetic data, and the quality of one-step and long-term predictions is investigated.

The remainder of the paper is structured as follows. In [Section 2](#), spatiotemporal ensemble data is considered and the possibility to incorporate implicit external factors via a nonstationary Markov model formulation is demonstrated. A short

introduction to the nonstationary spatiotemporal Markov and logistic regression is given in [Section 3](#), where new aspects are emphasized and existing theory is reviewed. In [Section 4](#), a self-containing strategy to make predictions by means of the determined model parameters and a new approach to assimilate additional hidden data after obtaining new observations are introduced. Proposed methods of discrete data modeling, prediction and assimilation are investigated numerically in [Section 5](#) for different synthetic model scenarios and systematically compared to the standard methods of the machine learning community, i.e., ANN [2; 24; 18; 3] and SVM [8; 35]. A comparison of the different numerical methods is given in terms of the information content (i.e., Akaike information criterion) and the quality of long- and short-term data-based online model predictions.

2. Ensemble data and exterior quantities

In the following, the discrete state $s_i \in \{s_1, \dots, s_{N_s}\}$ of a microscopic cell $\omega(j, l)$, with $l \in \{1, \dots, N_{\text{ens}}\}$ being the index of cells of a lattice on a microscopic level and $j \in \{1, \dots, N_j\}$ the corresponding macroscopic cell, is considered. Put differently, a macroscopic lattice, with each cell being further subdivided into smaller grid cells of a microscopic scale, is regarded. It is assumed that it is possible to assign each microscopic cell $\omega(j, l)$ its discrete state s_i via a stochastic process $\sigma(t, j, l)$ dependent on the time $t \in \{1, \dots, N_T\}$. Discrete dynamical systems of such form are common natural phenomena, e.g., representing the spatiotemporal dynamics of changes in the aggregate states of water in climate/atmosphere/ocean sciences. However, such systems represent a challenge for existing data-based analysis tools as it is usually not possible to have access to the corresponding data on a microscopic scale. Since observations of a single discrete realization $\sigma(t, j, l)$ in many realistic applications are not directly accessible, one resorts to the often available information on relative frequencies (with respect to the states) of a finite ensemble of microscopic locations on a macroscopic level. In detail, this means considering all the cells $\omega(j, l)$ with $l \in \{1, \dots, N_{\text{ens}}\}$ for fixed j (corresponding to the macroscopic scale) and measuring/observing the empirical probability

$$\tilde{\pi}_i(t, j) = \frac{N_{s_i}(t, j)}{N_{\text{ens}}}, \quad (1)$$

which is the ratio of $N_{s_i}(t, j)$, the number of cells $\omega(j, l)$ currently (i.e., for fixed time t) in state s_i , to N_{ens} , the total number of microscopic lattice cells contained in each macroscopic grid location (i.e., for fixed j). Formally, the total number of microscopic cells $\omega(j, l)$ currently in state s_i is defined as

$$N_{s_i}(t, j) = \sum_{l=1}^{N_{\text{ens}}} \delta_{s_i}(\sigma(t, j, l)), \quad (2)$$

whereas $\delta_{s_i}(\cdot)$ is the Kronecker delta for the value s_i , i.e., $\delta_{s_i}(\sigma(t, j, l)) = 1$ if $\sigma(t, j, l) = s_i$, otherwise it is zero. Further, a vector of empirical probabilities

$$\tilde{\pi}(t, j) = \begin{bmatrix} \tilde{\pi}_1(t, j) \\ \vdots \\ \tilde{\pi}_{N_S}(t, j) \end{bmatrix} \in [0, 1]^{N_S \times 1} \quad (3)$$

is a good estimate of the actual probability distribution as the number of microscopic cells N_{ens} in each macroscopic cell j is usually exceptionally large, i.e.,

$$\pi_i(t, j) := \mathbb{P}[\sigma(t, j, l) = s_i] \approx \tilde{\pi}_i(t, j) \quad (4)$$

with

$$\pi(t, j) = \begin{bmatrix} \pi_1(t, j) \\ \vdots \\ \pi_{N_S}(t, j) \end{bmatrix} \in [0, 1]^{N_S \times 1}. \quad (5)$$

Thus, for the remainder of this manuscript, we assume that the observed relative frequencies are equal to the probabilities, i.e., that $\pi(t, j)$ can be observed for $t \in \{1, \dots, N_T\}$ and $j \in \{1, \dots, N_J\}$. Further, it is assumed that the process σ is driven by time- and space-dependent external forces $\bar{u}(t, j) \in \mathbb{R}^{N_F \times 1}$, influencing the underlying system. A graphical interpretation of the discrete dynamical process σ by means of an example realization $\sigma(t, j, l)$ for fixed time t with only two possible states s_1 and s_2 (displayed in gray and white) is shown in [Figure 1](#). The image also displays the relation of the different lattice scales; i.e., each macroscopic cell contains a microscopic lattice with N_{ens} cells.

In the following, the aim is to approximate the dynamical system of interest underlying the stochastic process σ with data-based analysis tools by means of observations $\pi(t, j)$ and available measurements of exterior influencing quantities $\bar{u}(t, j)$.

Implicit external factors. In the following section, we will continue under the assumption that the stochastic process $\sigma(t, j, l)$ is a Markov process, i.e., the probability of the process to be in state s_i depends on the time-wise previous state.¹ A Markov process can be described via a transition matrix $P(\bar{u}(t, j)) \in [0, 1]^{N_S \times N_S}$. The transition probabilities $\pi(t+1, j)$ for the next time step can then be expressed through the so-called master equation:

$$\pi(t+1, j)^\top = \pi(t, j)^\top P(\bar{u}(t, j)). \quad (6)$$

Simultaneous measurement/modeling of all of the external factor components may impose a serious problem for realistic applications as it is impossible to have

¹Existing spatial correlations are going to be considered by including information on neighboring cells in the external factors.

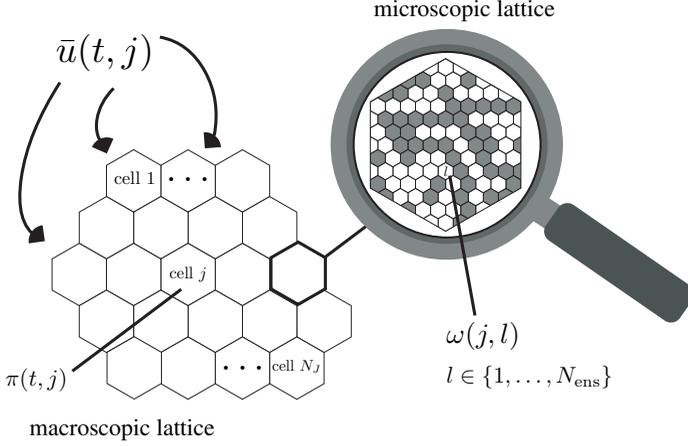


Figure 1. The above figure displays a graphical interpretation of the relation between the microscopic locations $\omega(j, l)$ and the macroscopic observation $\pi(t, j)$. The time t is fixed, and the considered system has two states, i.e., $N_S = 2$, which are displayed in white and gray. Thus, the process $\sigma(t, j, l)$ takes values in the set $\{s_1, s_2\} = \{\text{white, gray}\}$. The honeycomb lattice on the left-hand side corresponds to the macroscopic cells $j \in \{1, \dots, N_J\}$ associated with the observations $\pi(t, j)$. The microscopic lattice indexed $l \in \{1, \dots, N_{\text{ens}}\}$ is illustrated using a fine grid only clearly visible with a magnifying glass (see the hexagonal lattice on the right) and is contained in each cell of the coarse-grid. Additionally, the dependence of the dynamics of σ on external factors $\bar{u}(t, j)$ is visualized.

access to all the quantities influencing a system of interest in general. Therefore, in the following, we will distinguish between explicit and implicit external factors

$$\bar{u}(t, j) = \begin{bmatrix} u(t, j) \\ u^{\text{unres}}(t, j) \end{bmatrix} \in \mathbb{R}^{(N_E + N_I) \times 1} \quad (7)$$

and consider the known

$$u(t, j) = \begin{bmatrix} u_1(t, j) \\ \vdots \\ u_{N_E}(t, j) \end{bmatrix} \in \mathcal{U} \subset \mathbb{R}^{N_E \times 1} \quad (8)$$

as well as the unresolved factors

$$u^{\text{unres}}(t, j) = \begin{bmatrix} u_1^{\text{unres}}(t, j) \\ \vdots \\ u_{N_I}^{\text{unres}}(t, j) \end{bmatrix} \in \mathbb{R}^{N_I \times 1}, \quad (9)$$

according to their availability in the measurement/observation process.² It is important to stress that a vector of external factors $\bar{u}(t, j)$ consists of any quantities potentially playing a role in the dynamics of the regarded system including random,

² $N_F = N_E + N_I$.

deterministic or artificially added elements. For instance, the vector can contain influences other than the currently regarded scales $t \in \{1, \dots, N_T\}$ and $j \in \{1, \dots, N_J\}$ (time-wise as well as location-wise). Specifically, this means that important external forces coming from the microscopic scale as well as exterior factors having an impact on the state of the microscopic grid cells are included in $u^{\text{unres}}(t, j)$. Note, in particular, that the vector of implicit external factors is, as already mentioned above, not limited to deterministic factors but can have stochastic random processes as entries. Further, in order to consider existing spatial correlations, the mean of previous neighboring cell states that are calculated from the observational data $\pi(t-1, j)$ are added to the vector of explicit external factors representing another example of the wide range of possible and allowed quantities contained in $\bar{u}(t, j)$. Along the lines of [16], the abstract dependency of the transition matrix $P(\bar{u}(t, j))$ on unresolved external factors $u^{\text{unres}}(t, j)$ is approached by approximating the matrix with an appropriate linear combination of explicitly time- and space-dependent matrices. Specifically, such a nonstationary and nonhomogeneous formulation is possible under the following conditions:

Proposition 2.1. (1) *If the function $P(\bar{u}(t, j))$ is continuously differentiable and has bounded second derivatives, it can be decomposed in the form*

$$P(\bar{u}(t, j)) = P_0(t, j) + \sum_{e=1}^{N_E} P_e(t, j)u_e(t, j) + \varepsilon(t, j) \quad (10)$$

with $\mathbf{E}[\varepsilon(t, j)] = 0$ and $P_e(t, j) \in \mathbb{R}^{N_S \times N_S}$.

- (2) *If in addition to (1) the deviations of the entries of vector $\bar{u}(t, j)$ from their respective means are statistically independent in j and t , also the different realizations of $\varepsilon(t, j)$ are independent of each other in j and t .³*
- (3) *If the function $P(\bar{u}(t, j))$ is three times continuously differentiable and has bounded third derivatives, it can be decomposed in the form*

$$P(\bar{u}(t, j)) = P_0(t, j) + \sum_{e=1}^{N_E} (P_e(t, j) + \rho_e(t, j))u_e(t, j) + \varepsilon(t, j) \quad (11)$$

with $\mathbf{E}[\varepsilon(t, j)] = 0$ and $\mathbf{E}[\rho_e(t, j)] = 0$. Realizations of the noise process $\rho_e(t, j)$ for different t, j and e are not necessarily independent of each other or of $\varepsilon(t, j)$ realizations.

Proof. (1) For this proof, without loss of generality, we will assume that the external factors are ordered such that the explicit factors are the first N_E entries of $\bar{u}(t, j)$.

³ This does not necessarily imply that $\varepsilon(t, j)$ should also be identically distributed, i.e., i.i.d.

By performing a Taylor expansion on the transition matrix $P(\bar{u}(t, j))$ around the means $\mu(t, j) = [\mathbf{E}(\bar{u}_1(t, j)), \dots, \mathbf{E}(\bar{u}_{N_E+N_I}(t, j))] \in \mathbb{R}^{(N_E+N_I) \times 1}$, we obtain

$$P(\bar{u}(t, j)) = P(\mu(t, j)) + \sum_{e=1}^{N_E} \frac{\partial P(\mu(t, j))}{\partial \bar{u}_e(t, j)} (\bar{u}_e(t, j) - \mu_e(t, j)) + \sum_{|\alpha|=2} R_\alpha(\bar{u}(t, j)) (\bar{u}(t, j) - \mu(t, j))^\alpha, \quad (12)$$

where α is a multi-index and

$$R_\alpha(\bar{u}(t, j)) = \frac{2}{\alpha!} \int_0^1 (1-x) D^\alpha P(\mu(t, j) + x(\bar{u}(t, j) - \mu(t, j))) dx. \quad (13)$$

Note that $R_\alpha(\bar{u}(t, j))$ is bounded as the second derivatives of P are assumed to be bounded. Resorting the terms and defining

$$P_e(t, j) = \frac{\partial P(\mu(t, j))}{\partial \bar{u}_e(t, j)}, \quad e = 1, \dots, N_E, \quad (14)$$

$$\begin{aligned} \varepsilon(t, j) = & \sum_{e=N_E+1}^{N_E+N_I} \frac{\partial P(\mu(t, j))}{\partial \bar{u}_e(t, j)} (\bar{u}_e(t, j) - \mu_e(t, j)) \\ & + \sum_{|\alpha|=2} R_\alpha(\bar{u}(t, j)) (\bar{u}(t, j) - \mu(t, j))^\alpha \\ & - \mathbf{E} \left[\sum_{e=N_E+1}^{N_E+N_I} \frac{\partial \bar{P}(\mu(t, j))}{\partial \bar{u}_e(t, j)} (\bar{u}_e(t, j) - \mu_e(t, j)) \right. \\ & \left. + \sum_{|\alpha|=2} R_\alpha(\bar{u}(t, j)) (\bar{u}(t, j) - \mu(t, j))^\alpha \right], \quad (15) \end{aligned}$$

$$\begin{aligned} P_0(t, j) = & P(\mu(t, j)) - \sum_{e=1}^{N_E} \frac{\partial P(\mu(t, j))}{\partial \bar{u}_e(t, j)} \mu_e(t, j) \\ & + \mathbf{E} \left[\sum_{e=N_E+1}^{N_E+N_I} \frac{\partial P(\mu(t, j))}{\partial \bar{u}_e(t, j)} (\bar{u}_e(t, j) - \mu_e(t, j)) \right. \\ & \left. + \sum_{|\alpha|=2} R_\alpha(\bar{u}(t, j)) (\bar{u}(t, j) - \mu(t, j))^\alpha \right] \quad (16) \end{aligned}$$

yields (10) and especially $\mathbf{E}[\varepsilon(t, j)] = 0$.

(2) If the entries of the vector $\bar{u}(t, j) - \mu(t, j)$ for fixed $e \in \{1, \dots, N_F\}$ are independent for all j and t , the $\varepsilon(t, j)$ (as defined above) are just functions of the independent variables; thus, they are independent of each other again.

(3) The proof of this statement is given in [Appendix A](#). \square

Remarks 2.2. • The noise processes $\rho_e(t, j)$ and $\varepsilon(t, j)$ are not pairwise independent for fixed j and t . Further, there are no a priori assumptions concerning the distribution of $\rho_e(t, j)$ and $\varepsilon(t, j)$.

- Although the error $\varepsilon(t, j)$ is expected to be close to zero, it is important to mention that the variance of $\varepsilon(t, j)$ can take any value and therefore can lead to an arbitrary error term. This problem occurs most likely when the main influencing quantities are not available in the form of observational data.
- As the result of the proposition, the two expansions (10) and (11) deploy two conceptually different models of the noise for the master equation (6). Whereas (10) deploys a purely additive noise term, next-order expansion (11) contains a mixture of additive and multiplicative noise processes. Because of its simplicity, expansion (10) will be used for the construction of the nonhomogeneous and nonstationary data-driven Markov estimators in [Section 3](#).

Summarizing, an approach to address the predicament of missing data, specifically in the context of external influences, is proposed for dynamical system with an underlying Markovian process. It is assumed that the transition matrix has a linear structure so that the implicit dependency on unresolved external factors can be reflected in the explicit dependency on time and location.

3. Method

In this section we introduce methods for the analysis of discrete spatiotemporal data. As the details of nonstationary analysis of temporal data have already been addressed in earlier papers [[16](#); [17](#); [9](#); [30](#)], we will restrict this introduction to a short overview and will only emphasize new aspects concerning, e.g., the spatial component of the data or the details concerning the logistic regression.

3A. Inverse problem formulation. For a general consideration of the observed processes $\sigma(t, j, l)$, we assume that the correlation between the dynamical system and the measurements $\pi(t, j) \in [0, 1]^{N_S \times 1}$ can be expressed with a *direct mathematical model*

$$\pi(t+1, j) = f(\pi(t, j), \dots, \pi(t - N_M, j), \theta(\bar{u}(t, j))), \quad (17)$$

defined by a model function $f(\cdot)$ dependent on current and previous observations up to a memory depth N_M and model parameters $\theta(\bar{u}(t, j))$ from some parameter space Ω dependent on external factors $\bar{u}(t, j) \in \mathbb{R}^{N_F \times 1}$. Note that $\bar{u}(t, j)$ is a vector of all influences driving the system of interest. In particular, it can include information from the microscopic scale (e.g., from locations $\omega(j, l)$ with $l \in \{1, \dots, N_{\text{ens}}\}$) and other spatial components (e.g., neighboring cells), thus allowing to model

any existing spatial correlations. Further, the analytic expression of the model function f can also include random processes, e.g.,

$$f(\theta(t, j)) := \theta(t, j) + \lambda(t, j). \quad (18)$$

In this basic example, the random process $\lambda(t, j)$ has an expected value zero for all t and j , is i.i.d. (independent identically distributed) and can be interpreted as measurement errors or implicit quantities influencing the considered system. The reader is referred to [30] for more model function examples. For a given model function f and parameter function $\theta(\bar{u}(t, j))$, the problem of finding an appropriate time series $\pi(t, j)$ is called the *direct mathematical problem*. In this manuscript, we consider the opposite *inverse problem*: given the observations $\pi(t, j)$, which parameters $\theta(\bar{u}(t, j))$ with respect to the model function f describe the data “best”? In order to find model parameters $\theta(\bar{u}(t, j))$ that minimize the “distance” between the data and the model-based time series, we need to introduce a measuring functional

$$g(\pi(t+1, j), \dots, \pi(t-N_M, j), \theta(\bar{u}(t, j))) : \\ [0, 1]^{N_S} \times \dots \times [0, 1]^{N_S} \times \Omega \rightarrow \mathbb{R}_{\geq 0}, \quad (19)$$

which we will refer to as a *model distance function*. The corresponding inverse problem is defined as

$$\mathbf{L}(\theta(\bar{u}(t, j))) \\ = \sum_{t=1}^{N_T} \sum_{j=1}^{N_J} g(\pi(t+1, j), \dots, \pi(t-N_M, j), \theta(\bar{u}(t, j))) \rightarrow \min_{\theta(\bar{u}(t, j))} \quad (20)$$

and is referred to as an *averaged clustering functional*. A suitable function g can be derived from any metric $d(\cdot, \cdot)$:

$$g(\pi(t+1, j), \dots, \pi(t-N_M, j), \theta(\bar{u}(t, j))) \\ = (d(\pi(t+1, j), \mathbf{E}[f(\pi(t, j), \dots, \pi(t-N_M, j), \theta(\bar{u}(t, j)))]))^2. \quad (21)$$

We will consider the Euclidean metric $d_2(x, y) = \|x - y\|_2$ for the remainder of the manuscript. We will introduce two different model functions, f^{logit} and f^{Markov} , on which we will focus for the remainder of the paper. In particular, these two models will be numerically investigated in [Section 5](#).

3A1. Logistic regression. The model f^{logit} , introduced in the following, is a non-stationary and nonhomogeneous spatiotemporal extension of discrete choice models, which are standard techniques in the context of discrete data regression. This model class is a member in the generalized linear model (GLM) family [12; 10]. Discrete choice models can be derived from *utility theory* where the state of the regarded

process $\sigma(t, j, l)$ is assumed to be associated with a benefit or utility measure. In detail, this means that the process can be expressed as the function

$$\sigma(t, j, l) = \begin{cases} s_1 & \text{if } \mathcal{C}_1[u(t, j), B^1(t, j)] > \mathcal{C}_i[u(t, j), B^i(t, j)] \forall i \neq 1, \\ \vdots & \\ s_{N_S} & \text{if } \mathcal{C}_{N_S}[u(t, j), B^{N_S}(t, j)] > \mathcal{C}_i[u(t, j), B^i(t, j)] \forall i \neq N_S, \end{cases} \quad (22)$$

whereas

$$\mathcal{C}_i[u(t, j), B^i(t, j)] := \beta_0^i(t, j) + \sum_{e=1}^{N_E} \beta_e^i(t, j) u_e(t, j) + \xi^i(t, j) \quad (23)$$

is the utility measure dependent on unknown coefficients

$$B^i(t, j) = \begin{bmatrix} \beta_0^i(t, j) \\ \vdots \\ \beta_{N_E}^i(t, j) \end{bmatrix} \in \mathbb{R}^{(N_E+1) \times 1} \quad (24)$$

on observable (explicit) factors $u(t, j) \in \mathcal{U} \subset \mathbb{R}^{N_E \times 1}$ and on errors $\xi^i(t, j)$ characterizing the influences that could not be obtained through measurement (e.g., implicit external factors) [28; 29]. This implies that the probability for the dynamical process $\sigma(t, j, l)$ to be in state s_i can be expressed as follows:⁴

$$\begin{aligned} \mathbb{P}[\sigma(t, j, l) = s_i] &= \mathbb{P}[\mathcal{C}_i[u(t, j), B^i(t, j)] > \mathcal{C}_h[u(t, j), B^h(t, j)] \forall h \neq i] \quad (25) \\ &= \mathbb{P} \left[\beta_0^i(t, j) + \sum_{e=1}^{N_E} \beta_e^i(t, j) u_e(t, j) + \xi^i(t, j) \right. \\ &\quad \left. > \beta_0^h(t, j) + \sum_{e=1}^{N_E} \beta_e^h(t, j) u_e(t, j) + \xi^h(t, j) \forall h \neq i \right] \\ &= \mathbb{P} \left[\beta_0^i(t, j) - \beta_0^h(t, j) + \sum_{e=1}^{N_E} [\beta_e^i(t, j) - \beta_e^h(t, j)] u_e(t, j) + \xi^i(t, j) \right. \\ &\quad \left. > \xi^h(t, j) \forall h \neq i \right]. \end{aligned}$$

Various discrete choice models arise assuming different parametric forms of distributions for the random error terms $\xi^1(t, j), \dots, \xi^{N_S}(t, j)$. The logistic regression and the probit model are the most prominent examples of that model class; e.g., for logit models, the random part of the utility is assumed to be i.i.d. extreme value distributed (also known as Gumbel distribution), and for probit models, it is assumed to be multivariate normal. Results gained with either approach are similar, and

⁴Note that the probability of $\mathcal{C}_i[u(t, j), B^i(t, j)] = \mathcal{C}_h[u(t, j), B^h(t, j)]$ is assumed to be zero (see [29]).

significant differences are rare [26]. A multinomial logistic model, i.e., $N_S \geq 2$, is considered in the following. Consequently, the errors $\xi^1(t, j), \dots, \xi^{N_S}(t, j)$ are assumed to be i.i.d. with the Gumbel distribution resulting in the state probabilities

$$\mathbb{P}[\sigma(t, j, l) = s_i] = \frac{\exp\left(\beta_0^i(t, j) + \sum_{e=1}^{N_E} \beta_e^i(t, j)u_e(t, j)\right)}{\sum_{h=1}^{N_S} \exp\left(\beta_0^h(t, j) + \sum_{e=1}^{N_E} \beta_e^h(t, j)u_e(t, j)\right)} \quad \forall i. \quad (26)$$

The reader is referred to [29; 36] for a detailed probabilistic derivation. The corresponding model function f^{logit} with logistic regression parameter $B(t, j) = [B^1(t, j), \dots, B^{N_S}(t, j)] \in \mathbb{R}^{(N_E+1) \times N_S}$ is expressed as

$$\pi(t, j) := \theta^{\text{logit}}(B(t, j), u(t, j)) + \zeta(t, j), \quad (27)$$

where

$$\theta^{\text{logit}}(B(t, j), u(t, j)) = \begin{bmatrix} \mathbb{P}[\sigma(t, j, l) = s_1] \\ \vdots \\ \mathbb{P}[\sigma(t, j, l) = s_{N_S}] \end{bmatrix} \in \mathbb{R}^{N_S \times 1} \quad (28)$$

and $\zeta(t, j)$ is assumed to be an error process (e.g., please see the error of example model function given in (18)) related to the unknown implicit external influences and possible measurement errors. Note that there is no additional assumption concerning the probability distribution of $\zeta(t, j)$. The inverse problem corresponding to (27) with a model distance function g induced by the Euclidean metric has the form

$$\mathbf{L}(B(t, j)) = \sum_{t=1}^{N_T} \sum_{j=1}^{N_J} \|\pi(t, j) - \theta^{\text{logit}}(B(t, j), u(t, j))\|_2^2 \rightarrow \min_{B(t, j)}. \quad (29)$$

The standard logit model is one of the most used discrete choice models; nevertheless, it is important to check whether the problem setting of a certain considered application fits the model properties and whether it would be more reasonable to deploy a different discrete choice model. In this context, it is important to note that the logit model exhibits the *independence of irrelevant alternatives* (IIA) property [27], which states that for any two alternatives states s_i and s_h the ratio of the corresponding probabilities is

$$\exp\left(\beta_0^i(t, j) - \beta_0^h(t, j) + \sum_{e=1}^{N_E} (\beta_e^i(t, j) - \beta_e^h(t, j))u_e(t, j)\right). \quad (30)$$

In other words, the ratio does not depend on any state other than s_i and s_h and the relative odds remain the same [36]. Although this property might be realistic in some choice situations, it might be inappropriate in others [7]. Specifically, for

sets with similar states, i.e., states that are good substitutes of one another in the regarded system/application, the IIA property becomes implausible. This issue is often motivated with an example originating from a discussion McFadden offered in [29] on the subject: an individual takes one of the choices in the alternative set of states {auto, blue bus} with probability distribution $[2/3 \ 1/3]$, and then a red bus is added to the set of states, which causes the “intuitive” probability distribution, i.e., $[2/3 \ 1/6 \ 1/6]$, to vary from the one implied by the IIA axiom $[1/2 \ 1/4 \ 1/4]$.

The direct model function given in (27) can be extended in order to describe processes with memory, e.g., by including the previous (in time) and/or neighboring (in location space) values of the probability density $\pi(t, j)$ as the additional components of the external factors vector $u(t, j)$, e.g.,

$$u_{N_E+1}(t, j) := \pi_1(t-1, j). \quad (31)$$

Such logit models with Markov effects incorporated in the above form of external factors are known as *dynamical logit models* [32; 15]. One of the main drawbacks of the logistic regression ansatz is the internally embedded mapping (from the closed interval $[0, 1]$ to the continuum of real numbers $(-\infty, \infty)$) used to approach the discrete/categorical data with continuous regression techniques. This transformation causes computational instability on the boundaries of the logistic cumulative density function. Further, it is not possible to directly access the impact of the explicit external factors, which complicates the interpretations of the exterior influences.

Nevertheless, logistic regression is a good option for systems with nonlinear behavior. As a matter of fact, a nonlinear process can also be interpolated via a sequence of piecewise linear but nonstationary and nonhomogeneous local models. But in a case when the dynamics of the observed process are nonlinear as well as nonstationary and nonhomogeneous, it is more sensible to describe the system with an intrinsically nonlinear model (e.g., the nonstationary nonhomogeneous logistic regression).

3A2. Markov regression. As a locally linear alternative to the logistic regression model described above, we consider a nonstationary nonhomogeneous Markov regression. In order to incorporate all external factors in the model, we assume that the transition matrix $P(\bar{u}(t, j))$ corresponding to an observed Markovian dynamical process $\sigma(t, j, l)$ is continuously differentiable and has bounded second derivatives. Employing the results of Proposition 2.1, the following decomposition of the transition matrix is considered:

$$P(t, j, u(t, j)) = P_0(t, j) + \sum_{e=1}^{N_E} P_e(t, j) u_e(t, j). \quad (32)$$

The model function f^{Markov} is defined on the basis of an adapted stochastic master

equation (6):

$$\pi(t+1, j)^\top := \pi(t, j)^\top (P(t, j, u(t, j)) + \varepsilon(t, j)). \quad (33)$$

Then it is possible to formulate the following inverse problem:

$$\begin{aligned} & \mathbf{L}(P(t, j, u(t, j))) \\ &= \sum_{t=1}^{N_T} \sum_{j=1}^{N_J} \|\pi(t+1, j)^\top - \pi(t, j)^\top P(t, j, u(t, j))\|_2^2 \rightarrow \min_{P(t, j, u(t, j))}. \end{aligned} \quad (34)$$

3B. Interpolation. The optimization problem (20) exhibits several computational drawbacks such as ill-posedness (in the sense of Hadamard [13]) and therefore needs to undergo a series of changes in the form of regularizations. In the following, we make use of the fact that many real-life systems from various areas of application exhibit a certain level of persistence. Subsequently, it is possible to interpolate the model parameter function $\theta(\bar{u}(t, j))$ with a fixed number of N_K stationary and homogeneous model parameters $\theta_k(u(t, j))$ and corresponding affiliations $\gamma_k(t, j)$ with $k \in \{1, \dots, N_K\}$. This approach leads to a less ill-posed description of the considered dynamical system. Thus, assuming the existence of such local models $\Theta(u(t, j)) = [\theta_1(u(t, j)), \dots, \theta_{N_K}(u(t, j))]$ and weights $\Gamma(t, j) = [\gamma_1(t, j), \dots, \gamma_{N_K}(t, j)] \in [0, 1]^{1 \times N_K}$, the model distance functional first introduced in (19) can be phrased in the following interpolated formulation:

$$\begin{aligned} & g(\pi(t+1, j), \dots, \pi(t-N_M, j), \theta(\bar{u}(t, j))) \\ &= \sum_{k=1}^{N_K} \gamma_k(t, j) g(\pi(t+1, j), \dots, \pi(t-N_M, j), \theta_k(u(t, j))). \end{aligned} \quad (35)$$

The affiliation process $\Gamma(t, j)$ characterizes the regime behavior and the nonstationary and nonhomogeneous nature of the dynamical system. The weights $\gamma_k(t, j)$ have the specification to take positive values and sum up to one over all N_K local models, i.e.,

$$\sum_{k=1}^{N_K} \gamma_k(t, j) = 1 \quad \forall j \in \{1, \dots, N_J\}, t \in \{1, \dots, N_T\}, \quad (36)$$

$$\gamma_k(t, j) \geq 0 \quad \forall j \in \{1, \dots, N_J\}, t \in \{1, \dots, N_T\}, k \in \{1, \dots, N_K\}. \quad (37)$$

Then the corresponding inverse problem can formally be expressed by

$$\mathbf{L}(\Gamma(t, j), \Theta(u(t, j))) = \sum_{j=1}^{N_J} \mathbf{L}_j(\Gamma(:, j), \Theta(u(t, j))) \rightarrow \min_{\Gamma(t, j), \Theta(u(t, j))} \quad (38)$$

with

$$\begin{aligned} & \mathbf{L}_j(\Gamma(\cdot, j), \Theta(u(t, j))) \\ &= \sum_{t=1}^{N_T} \sum_{k=1}^{N_K} \gamma_k(t, j) g(\pi(t+1, j), \dots, \pi(t-N_M, j), \theta_k(u(t, j))). \end{aligned} \quad (39)$$

Note that the constraints are independent for each location j . This independence in space and the structure of the functional \mathbf{L} will be exploited in the numerical optimization of (38) with respect to $\Gamma(t, j)$. The main idea is that every location j can be regarded separately due to the fact that the overall functional \mathbf{L} is a sum of local (uncoupled in $\Gamma(\cdot, j)$) functionals \mathbf{L}_j with (uncoupled in $\Gamma(\cdot, j)$) constraints (36) and (37). A corresponding numerical algorithm exploiting this structure of the problem will be discussed in detail in Section 3D. The influence of the implicit external factors $u^{\text{unres}}(t, j)$ is reflected in the explicit time- and space-dependence of the affiliation process $\Gamma(t, j)$.

In case of the logistic regression, this regularization means that we need to find a set of locally stationary and homogeneous (i.e., not dependent on time t and location j) model parameters $\{B_1, \dots, B_{N_K}\}$ with $B_k = [B_k^1, \dots, B_k^{N_S}] \in \mathbb{R}^{(N_E+1) \times N_S} \forall k \in \{1, \dots, N_K\}$. For the Markov regression, the interpolated version of (34) is

$$\begin{aligned} & \mathbf{L}(\Gamma(t, j), P(u(t, j))) \\ &= \sum_{j=1}^{N_J} \sum_{t=1}^{N_T} \sum_{k=1}^{N_K} \gamma_k(t, j) \|\pi(t+1, j)^\top - \pi(t, j)^\top P^k(u(t, j))\|_2^2 \rightarrow \min_{\Gamma(t, j), P(u(t, j))}, \end{aligned} \quad (40)$$

where the local Markovian transition operators $P(u(t, j)) = [P^1(u(t, j)), \dots, P^{N_K}(u(t, j))] \in \mathbb{R}^{N_S \times N_S N_K}$ for fixed t and j are defined in a linear approximation:

$$P^k(u(t, j)) = P_0^k + \sum_{e=1}^{N_E} P_e^k u_e(t, j) \quad \forall k \in \{1, \dots, N_K\}. \quad (41)$$

To ensure that the stochasticity of the Markov transition operator remains preserved, the optimization problem is subject to a number of constraints. Since the transition matrices $P^k(u(t, j))$ are stochastic matrices, the matrices P_e^k are required to satisfy the equalities

$$P_0^k \mathbf{1} = \mathbf{1} \quad \forall k \in \{1, \dots, N_K\}, \quad (42)$$

$$P_e^k \mathbf{1} = \mathbf{0} \quad \forall e \in \{1, \dots, N_E\}, k \in \{1, \dots, N_K\}, \quad (43)$$

whereas $\mathbf{1} \in \mathbb{R}^{N_S \times 1}$ is a column vector with all entries equal to one and analogously $\mathbf{0} \in \mathbb{R}^{N_S \times 1}$ refers to the corresponding vector with all entries equal to zero. Furthermore, the entries of $P^k(u(t, j))$ need to be greater than or equal to zero. In the case of a rectangular domain \mathcal{Q} , the feasible number of 2^{N_E} inequality constraints

(consisting of all possible combinations of suprema and infima of the N_E explicit external factors $u_e(t, j)$)

$$\{P_0^k\}_{n,m} + \sum_{e=1}^{N_E} \{P_e^k\}_{n,m} \left[\begin{array}{c} \sup_{t,j} u_e(t, j) \\ \inf_{t,j} u_e(t, j) \end{array} \right] \geq 0 \quad \forall k, n, m \quad (44)$$

is sufficient to satisfy this condition. See [30] for more details and a proof of (44) for the purely temporal case; extension to the spatiotemporal case given in equations (41)–(44) above is straightforward.

3C. Spatial and temporal persistence. The problem formulation is still ill-posed since its solution may not be unique due to many possibilities to choose the switching process Γ . Therefore, we need to make further assumptions/restrictions on the function space that contains the switching process and add another constraint to the optimization problem. More precisely, to approach this issue, we limit the number of transitions of $\gamma_k(\cdot, j)$, introducing a persistency constraint on the time interval

$$|\gamma_k(\cdot, j)|_{\text{BV}(1, N_T)} = \sum_{t=1}^{N_T-1} |\gamma_k(t+1, j) - \gamma_k(t, j)| \leq N_C \quad (45)$$

that holds for every location $j \in \{1, \dots, N_J\}$. Without an additional spatial regularization, the constraints for parameter $\Gamma(t, j)$ are still independent for every location. This structural advantage allows us to compute each $\Gamma(\cdot, j)$ separately if the value of the parameter $\Theta(u(t, j))$ is kept fixed. In some situations, it might be reasonable to limit the variation along the locations as well (e.g., a limitation concerning only the neighboring cells of a location), but constraints on the switching process Γ would result in a global coupling (in j) for different optimization problems L_j from (39), leading to immense numerical costs. Furthermore, an identification of the best model in terms of parameter choice, discussed in the next paragraph, would have to be pursued for all possible combinations of choices for $N_C(j)$, $j \in \{1, \dots, N_J\}$, as well, leading to a computationally expensive analysis. This additional regularization over spatial locations is an aspect of further research.

3D. Numerical approach and computational complexity. The inverse problem posed in (38) has no general analytic solution and is not convex (i.e., it is not possible to obtain an unique global minimum with standard approaches, e.g., gradient descent or Newton methods). But the global optimizers $\Gamma^*(t, j)$ and $\Theta^*(u(t, j))$ can be approximated combining a *subspace algorithm* and simulated annealing [22]. The main idea of the subspace algorithm is to exploit the above-mentioned structural property of the optimization problem (38), i.e., that the simple convex optimization problems can be stated for Γ and Θ separately, i.e., for (i) an optimization with respect to $\Gamma(\cdot, j)$ for a fixed Θ and (ii) an optimization with respect to Θ for

a fixed Γ . Dividing the optimization problem with two sets of unknowns into two minimizations over just one set of parameters reduces the originally high-dimensional and nonconvex problem to two manageable problems that can be approached with standard optimization techniques, e.g., simplex method for the above subspace step (i) and quadratic minimization with linear equality and inequality constraints for the above subspace step (ii). It is straightforward to demonstrate that the subsequent repetition of steps (i) and (ii) leads to a strict minimization of the original functional \mathbf{L} , and since the average model distance functional is bounded with zero from below, this procedure will converge to a local minimum of \mathbf{L} . Iterations over the subproblems only converge to local minima, and simulated annealing approaches [22; 25] can be deployed in combination with the subspace-iteration algorithm to avoid getting trapped in the local minimum. The details of the algorithm are now given in the pseudocode in [Algorithm 1](#).

In contrast to the time-dependent algorithm introduced in [30], the additional spatial dimension j is involved in the above scheme. Since a spatial regularization is not included, the affiliations $\Gamma(\cdot, j)$ are determined for each location j separately (see the for-loop on line 6), i.e., the problem of optimizing \mathbf{L} with respect to Γ is equivalent to separate optimization of N_J suboptimization problems given by functionals \mathbf{L}_j defined in (39). The local stationary and homogeneous model parameters θ_i , on the other hand, are computed for all t and j simultaneously (line 9). A separate computation for every spatial component is not possible here since different \mathbf{L}_j are coupled through Θ .

In order to obtain a global minimizer of (38), the subspace-iteration algorithm is repeated $N_{\text{anneal}}^{\text{FEM}}$ times with different randomly sampled initial parameters $\Gamma^{[0]}$ (see lines 2 and 3). This form of simulated annealing helps to avoid local minima by trying to consider the entire parameter space. Since the annealing steps can be run independently, it is possible to reduce the corresponding computational complexity via an “embarrassingly parallel” implementation. The necessary memory capacity as well as the computing time can be further decreased by using a time-discretized (with finite elements) version of the full process Γ [30]. This form of dimension reduction is especially beneficial when modeling time-persistent dynamical systems with few transitions between the local models (i.e., systems where a comparatively small number of finite element functions ($N_{\text{basis}}^{\text{FEM}} \ll N_T$) is sufficient for qualitative results).

Computational cost of the proposed technique is dependent on the number of locations N_J and the number $N_{\text{basis}}^{\text{FEM}}$ of finite elements for the time discretization. The run time for the Γ calculation is proportional to $\mathcal{O}(N_J N_K (2N_{\text{basis}}^{\text{FEM}} - 1)^\kappa)$, where $\kappa \geq 1$ is the parameter dependent on the choice of the numerical scheme for the $\Gamma(\cdot, j)$ -optimization (Step 1 of [Algorithm 1](#)). As already indicated above, the computational complexity of the determination of Θ varies for different model classes and the spatial component can be regarded as an additional dimension in the

input : Set number of different regimes N_K , value for time-wise transition boundary N_C , number of simulated annealing steps $N_{\text{anneal}}^{\text{FEM}}$ and optimization tolerance value τol (optional: number of finite-element functions $N_{\text{basis}}^{\text{FEM}}$).

output: Global optimizers $\Gamma^*(t, j)$ and $\Theta^*(u(t, j))$

- 1 $\mathbf{L}_{\min} = 1000000$
- 2 **for** $r = 1 : N_{\text{anneal}}$ **do**
- 3 Generate random initial $\Gamma_r^{[0]}$ and compute $\Theta_r^{[0]}$.
- 4 **while** $|\mathbf{L}(\Gamma_r^{[s]}, \Theta_r^{[s]}) - \mathbf{L}(\Gamma_r^{[s-1]}, \Theta_r^{[s-1]})| \geq \tau ol$ **do**
- 5 Step 1:
- 6 **for** $j = 1 : N_J$ **do**
- 7 Determine $\Gamma_r^{[s+1]}(:, j) = \arg \min \mathbf{L}_j(\Gamma(:, j), \Theta_r^{[s]})$ subject to constraints (36), (37) and (45), whereas $\Theta_r^{[s-1]}$ denotes the current fixed approximation of the optimal Θ^* . Apply standard methods of linear minimization with linear equality and inequality constraints (e.g., simplex method).
- 8 Step 2:
- 9 Compute $\Theta_r^{[s+1]} = \arg \min \mathbf{L}(\Gamma_r^{[s+1]}, \Theta)$ (additional constraints depend on the model, e.g., constraints (42)–(44) in case of the Markovian process and no constraints in the logistic regression case). Apply standard methods of quadratic optimization with linear equality and inequality constraints.
- 10 $s := s + 1$.
- 11 **if** $\mathbf{L}_{\min} \geq \mathbf{L}(\Gamma_r^*(t, j), \Theta_r^*(u(t, j)))$ **then**
- 12 $\mathbf{L}_{\min} = \mathbf{L}(\Gamma_r^*(t, j), \Theta_r^*(u(t, j)))$
- 13 $\Gamma^* = \Gamma_r^*$
- 14 $\Theta^* = \Theta_r^*$
- 15 **Return** Γ^* and Θ^* .

Algorithm 1: Subspace algorithm with annealing steps.

problem. In Step 2 of the algorithm, one needs to solve a quadratic minimization problem subject to linear constraints (equalities and inequalities) to compute the matrices P_e^k considering the nonstationary nonhomogeneous Markov regression (see (40)). Such problems are known to be NP-complete [37]. For the logistic model (see (27)), the computational complexity of the Step 2 can be expressed as $\mathcal{O}(N_K N_T N_J)$ [31]. The overall resulting numerical cost of the proposed method is in the range of the average complexity of standard approaches such as artificial neural networks ($\mathcal{O}((N_{\text{weights}}^{\text{ANN}})^3)$ where $N_{\text{weights}}^{\text{ANN}}$ is the number of ANN parameters, i.e., neural biases and weights⁵) and support vector machines ($\mathcal{O}(N_T^2 N_E)$ with N_E

⁵This number is directly proportional to the number of neurons and depends on the type of the transfer functions and network architecture.

referring to the number of explicit external factors). Details of the techniques and their computational time complexity will be discussed in [Section 5](#).

3E. Information criterion. A further issue originates from the selection of the parameters N_K and N_C , which can lead to a variety of models differing in terms of quality and complexity. This problem is addressed by applying a *modified* formulation of *Akaike's information criterion* (mAIC). The main idea of the method is based on approximating the time series of the obtained model errors $g(\pi(t+1, j), \dots, \pi(t-N_M, j), \theta_k(u(t, j)))$ through an optimal nonparametric scalar-valued stochastic process, followed by the comparison of the mAIC values for the obtained processes from different models. A detailed description of the method can be found in [\[30\]](#). The main advantage of this approach is that no a priori parametric probabilistic assumptions about the analyzed data are necessary.⁶

The main idea of an information criterion is that the quality of the determined model is weighted against the total number of parameters involved in the calculation of the model [\[1\]](#). In other words, the aim is to identify the model that fits best with the fewest number of necessary model parameters, e.g.,

$$\text{mAIC}(N_K, N_C) = -2 \log(\mathcal{L}(N_K, N_C)) + 2|M(N_K, N_C)|. \quad (46)$$

Here the likelihood $\mathcal{L}(N_K, N_C)$ corresponds to the underlying model characterized by N_K different regimes with a maximum of N_C transitions between them and is defined as

$$\begin{aligned} &\mathcal{L}(N_K, N_C) \\ &= \prod_{j=1}^{N_J} \prod_{t=1}^{N_T} \prod_{k=1}^{N_K} \sum \gamma_k(t, j) \phi_k(g(\pi(t+1, j), \dots, \pi(t-N_M, j), \theta_k(u(t, j))) | N_{\phi_k}). \end{aligned} \quad (47)$$

A detailed derivation of the likelihood function for the nonstationary case can also be found in [\[30\]](#). The expression above is its straightforward extension to the nonstationary and nonhomogeneous case. The functional $M(N_K, N_C)$ describes the total number of involved model parameters, which in the case of the logistic regression consists of

$$|M^{\text{logit}}(N_K, N_C)| = |\Gamma| + N_K(N_E + 1) \quad (48)$$

and for the Markov regression is

$$|M^{\text{Markov}}(N_K, N_C)| = |\Gamma| + N_K N_S (N_S - 1) (N_E + 1). \quad (49)$$

This modified version of Akaike's information criterion coupled with the nonstationary and nonhomogeneous logistic and Markov regression (introduced above)

⁶Such parametric a priori assumptions are needed to compute the log-likelihood of the data in context of standard information criteria like AIC.

allows us to simultaneously identify the optimal model and the optimal values of the parameters N_K and N_C .

In practice, mAIC values for different cluster values $N_K \in \mathcal{S}_1$ and persistency parameter values $N_C \in \mathcal{S}_2$ might not vary substantially. By appointing only one model, other suitable ones are discarded, resulting in an unnecessary information loss [5]. In this case, the mAIC values of the possible models are ranked via the deviation from the lowest mAIC value, i.e.,

$$\Delta(N_K, N_C) = \exp \left[\frac{\min_{(N'_K, N'_C) \in \mathcal{S}_1 \times \mathcal{S}_2} (\text{mAIC}(N'_K, N'_C)) - \text{mAIC}(N_K, N_C)}{2} \right]. \quad (50)$$

If there is more than one probable model, then the overall model can be considered as a multimodel, i.e., a weighted linear combination of individual models with the model weights [5] given by

$$w(N_K, N_C) := \frac{\Delta(N_K, N_C)}{\sum_{(N'_K, N'_C) \in \mathcal{S}_1 \times \mathcal{S}_2} \Delta(N'_K, N'_C)}. \quad (51)$$

Besides determining the optimal model with respect to the parameters N_K and N_C , the criterion can also be used to determine the better model in terms of the prior assumptions. Since different models are compared with respect to the same observation data and the same form of the nonparametric likelihood-estimation procedure described in [30] (based on fitting the optimal stochastic process to the time series of the model residuals), resulting mAIC values can be used to identify the statistically optimal model from a given class of models (e.g., Markov, logit, ANN and SVM). Practical examples of this data-based model-discrimination procedure will be given in the last sections of this manuscript.

4. Prediction and assimilation of additional information

Suppose the global optimal model parameters $\Gamma^*(t, j)$ and $\Theta^*(u(t, j))$ with respect to the average model distance functional $\mathbf{L}(\Gamma(t, j), \Theta(u(t, j)))$ introduced in (38) can be determined with the proposed numerical scheme (see Algorithm 1); then it is possible to approximate the observed time series

$$\pi(t+1, j) \approx f \left(\pi(t, j), \dots, \pi(t - N_M, j), \sum_{k=1}^{N_K} \gamma_k^*(t, j) \theta_k^*(u(t, j)) \right) \quad (52)$$

on the basis of the formal definition of the direct model function. This ansatz, used to approximate the vector of state probabilities, is discussed in detail in [30] and allows

us to directly concatenate the two model parameters $\Gamma^*(t, j)$ and $\Theta^*(u(t, j))$.⁷ In most of the practical applications, a further aspect of interest is a prediction $\hat{\pi}(N_T + N_{\text{pred}}, j)$ of the probability distribution $\pi(N_T + N_{\text{pred}}, j)$ outside of the observed time sequence $\{1, \dots, N_T\}$. The quantity N_{pred} denotes the prediction depth, i.e., the total number of prediction steps in time. The difficulty lies in the nonstationarity and nonhomogeneity of the model formulation; i.e., any prediction crucially depends on $\Gamma^*(t, j)$, which is only defined for the observed time sequence $\{1, \dots, N_T\}$. In order to predict future affiliations $\hat{\Gamma}(N_T + N_{\text{pred}}, j)$, the process $\Gamma^*(t, j)$ can be regarded as an observed time series of probabilities to be in N_K different discrete states. Subsequently, the proposed Markov regression framework (given in (40)) can be applied to determine the model parameters describing $\Gamma^*(t, j)$. To avoid an infinite sequence of prediction problems caused by nonstationarity and nonhomogeneity, the model of the affiliation process $\Gamma^*(t, j)$ is assumed to have only one regime (i.e., $N_K = 1$). Although this is a strong restriction, it is important to note that stationarity as well as homogeneity are common assumptions in time-series analysis. This self-contained strategy to determine

$$\hat{\Gamma}(N_T + N_{\text{pred}}, j) = \Gamma^*(N_T, j) \prod_{\tau=0}^{N_{\text{pred}}-1} \left(\left[P_0^\Gamma + \sum_{e=1}^{N_E} P_e^\Gamma u_e(N_T + \tau, j) \right] \right) \quad (53)$$

has been introduced in [16] (in the context of purely time-dependent data) and further discussed and deployed in [30]. The model transition matrix, characterizing the dynamics of the affiliation $\Gamma^*(t, j)$, is denoted $P^\Gamma(u(t, j))$ and is a linear combination of explicit external factors $u(t, j)$ and matrices $P_0^\Gamma, \dots, P_{N_E}^\Gamma$ (see (41) for $N_K = 1$). In a case when the data $\pi(N_T + 1, j)$ for the next time step can be obtained, the new information can be used to update the $\hat{\Gamma}(t, j)$ -predictor. A strategy for updating the prediction $\hat{\Gamma}(N_T + 1, j)$ conditioned on the additional information $\pi(N_T + 1, j)$ has recently been introduced in [16; 30] and is based on the maximum-likelihood principle, i.e.,

$$\begin{aligned} \gamma_k^*(N_T + 1, j) &= \begin{cases} 1 & \text{if } k = \arg \min_h g(\pi(t + 1, j), \dots, \pi(t - N_M, j), \theta_h(u(t, j))), \\ 0 & \text{otherwise.} \end{cases} \end{aligned} \quad (54)$$

The update $\gamma_k^*(N_T + 1, j)$ is assumed to be optimal (hence the superscript $*$). In detail, this means that it is possible to identify all local regimes θ_k describing the dynamical process $\sigma(t, j, l)$ on the basis of the available data measured in the time sequence $\{1, \dots, N_T\}$. Further, it is necessary to assume the affiliation process Γ^*

⁷Note that the model function f needs to be linear in its parameters and the model distance functional g has to be strictly convex to pursue the equation given in (52) (for a detailed derivation see [30]). This is the case for the proposed Markov as well as for the logistic model.

is deterministic (i.e., takes only values in the set $\{0, 1\}$). In the following, a new update method is proposed on the basis of Bayes' theorem that allows for a fuzzy affiliation. We denote $\Gamma(N_T + 1, j)$ to be the true but unknown cluster affiliation of the dynamical system under observation and $\widehat{\Gamma}(N_T + 1, j)$ the (prior) prediction, calculated only with the information from the previous time steps $t \in \{1, \dots, N_T\}$, and as $\dot{\Gamma}(N_T + 1, j)$, we denote the posterior estimate based on the new observation $\pi(N_T + 1, j)$. The following proposition gives an analytical form of the posterior estimate of the hidden model affiliation function and shows how the implicit impact of the unresolved external factors can be assimilated into the model:

Proposition 4.1. *Let the entries of $\gamma_k(t, j)$ for all j, t and k only assume values zero or one and the predictor $\widehat{\Gamma}(N_T + 1, j)$ be a prior probability distribution for $\Gamma(N_T + 1, j)$ in the sense that*

$$\mathbb{P}[\gamma_k(N_T + 1, j) = 1] = \hat{\gamma}_k(N_T + 1, j). \quad (55)$$

Moreover, let the distribution of the observation $\pi(N_T + 1, j)$ given the information about the affiliation γ_k at time t be independent of the prediction $\widehat{\Gamma}(t, j)$. Then the posterior distribution of the regime assigning process $\Gamma(N_T + 1, j)$ is of the form

$$\dot{\gamma}_k(N_T + 1, j) = \frac{\mathbb{P}[\pi(N_T + 1, j) | \gamma_k(N_T + 1, j) = 1] \hat{\gamma}_k(N_T + 1, j)}{\sum_{h=1}^{N_K} \mathbb{P}[\pi(N_T + 1, j) | \gamma_h(N_T + 1, j) = 1] \hat{\gamma}_h(N_T + 1, j)}. \quad (56)$$

Proof. Using the above assumptions and Bayes' theorem, the following holds:

$$\begin{aligned} & \mathbb{P}[\gamma_k(N_T + 1, j) = 1 | \pi(N_T + 1, j)] \\ &= \frac{\mathbb{P}[\gamma_k(N_T + 1, j) = 1; \pi(N_T + 1, j)]}{\mathbb{P}[\pi(N_T + 1, j)]} \\ &= \frac{\mathbb{P}[\pi(N_T + 1, j) | \gamma_k(N_T + 1, j) = 1] \mathbb{P}[\gamma_k(N_T + 1, j) = 1]}{\mathbb{P}[\pi(N_T + 1, j)]} \\ &= \frac{\mathbb{P}[\pi(N_T + 1, j) | \gamma_k(N_T + 1, j) = 1] \mathbb{P}[\gamma_k(N_T + 1, j) = 1]}{\sum_{h=1}^{N_K} \mathbb{P}[\pi(N_T + 1, j) | \gamma_h(N_T + 1, j) = 1] \mathbb{P}[\gamma_h(N_T + 1, j) = 1]}. \quad \square \end{aligned}$$

As will be demonstrated by numerical examples in the next section, formula (56) improves an estimation of the new affiliations in comparison to the maximum-likelihood approach (54) deployed in [16; 30]. Although the affiliation is “fuzzy” (i.e., resulting affiliations may take values between zero and one), it is (as demonstrated by the numerical tests) less prone with respect to introducing unjustified switches between the local models.

5. Numerical investigation

To explore the characteristic properties of the introduced nonstationary and nonhomogeneous regression framework, we apply it to three different synthetic data sets. Note that we actively chose to work with artificial rather than real-life examples due to the specific settings necessary to analyze the proposed framework. In a real-life observation, for example, there is no reliable information about the influencing factors $u^{\text{unres}}(t, j)$ that are not available in form of measurements.

Different model functions (e.g., Markov and logit) for the framework proposed in [Section 3](#) as well as other standard techniques of time-series analysis (e.g., SVM and ANN) are considered in the following. It is necessary to distinguish between the different resulting model parameters via additional superscript tags (e.g., $\Gamma^{\text{Markov}}(t, j)$ or $\Gamma^{\text{logit}}(t, j)$). The same labeling system is employed for approximations of the actual observations $\pi(t, j)$ determined with model parameters computed with various methodologies (e.g., $\pi^{\text{Markov}}(t, j)$ or $\pi^{\text{logit}}(t, j)$ or $\pi^{\text{ANN}}(t, j)$). Due to the fact that the considered observations are artificial, all parameters and variables used to generate the synthetic data are tagged with the superscript *syn* (e.g., $\Gamma^{\text{logit}}(t, j)$). Some tags are specifying the settings used for a specific algorithm such as the number of annealing steps (e.g., $N_{\text{anneal}}^{\text{ANN}}$ or $N_{\text{anneal}}^{\text{FEM}}$) or the regularization factor (e.g., N_C^{FEM} or N_C^{SVM}). Note that the regularization factor N_C can have superscripts FEM as well as Markov or logit although all of those labels correspond to the technique proposed in [Section 3](#). This further distinction is necessary as the abbreviation FEM is a general reference to the framework introduced in the current manuscript. Resulting parameters derived with any technique, which are considered to be optimal in the sense that the corresponding model has the lowest AIC, have a superscript $*$.

A few variables remain free of labels as they are independent of the parameter-identification process and are assumed to be the same for all the employed techniques, e.g., the number of explicit external factors N_E , the number of discrete states N_S or the number of considered locations N_J .

One aspect of the numerical investigation includes testing of the various considered parameter-identification techniques with respect to predictions, i.e., approximate data that was not given for the computation of corresponding model parameters. Thus, it is necessary to divide the time sequence $\{1, \dots, N_T\}$ describing the time-wise length of the observations $\pi(t, j)$ into two components $\{1, \dots, N_{T_{\text{train}}}\}$ and $\{N_{T_{\text{train}}} + 1, \dots, N_T\}$. The first sequence will be referred to as *training sequence* and the second one will be known as *test sequence*.

The first data set is discussed in [Section 5A](#) and is employed to demonstrate the general feasibility of the proposed nonstationary and nonhomogeneous Markov regression as well as the logistic regression frameworks under “good conditions” (all relevant data is given for the computations, i.e., no unresolved external factors

$u^{\text{unres}}(t, j)$). In [Section 5B](#), the focus is on the key attribute of the Markov regression technique presented in this paper, which allows us to take missing/unavailable external factors into account. To numerically investigate this theoretical incorporation of unobserved information, a synthetic data set is generated with $N_F = 101$ external factors $\bar{u}(t, j)$ and only one of these 101 factors is made available for the calculation of the model parameters (i.e., $N_E = 1$ and $N_I = 100$).

The last example data set is chosen to numerically investigate (again considering the nonstationary and nonhomogeneous Markov regression) the newly proposed update of the prediction $\widehat{\Gamma}(N_T + 1, j)$ (see [Proposition 4.1](#)). The quality of the determined model is analyzed and compared to the results of two standard frameworks from machine learning (namely artificial neural networks [[2](#); [24](#); [18](#); [3](#)] and support vector machines [[8](#); [35](#)]).

5A. Nonstationary example. Under ideal conditions, the regarded dynamical process $\sigma(t, j, l)$ has the Markov property and all external influences are available in the form of observation data. The toy example considered in this section allows us to check the basic feasibility of the proposed technique and also serves as a reference for an example under “bad conditions”, investigated in [Section 5B](#). The data is generated using the proposed Markov model structure (see [\(32\)](#)) and pseudorandom numbers generated by the computer. In the following, two algorithms are outlined in order to explain the synthetic data. At first, the affiliation process $\Gamma^{\text{syn}}(t, j)$ subject to constraints [\(45\)](#), [\(36\)](#) and [\(37\)](#) is generated.

The synthetic parameter Γ^{syn} is generated with pseudorandom numbers that, for simplicity, are restricted to the set $\{0, 1\}$. Furthermore, a certain level of persistency is forced on $\Gamma^{\text{syn}}(t, j)$, meaning that the total number of transitions is limited to N_C^{syn} (see lines 3–12 of [Algorithm 2](#)). As the weights $\gamma_k^{\text{syn}}(t, j)$, generated with [Algorithm 2](#), only take values in $\{0, 1\}$, it is possible to directly assume⁸

$$P^{\text{syn}}(t, j, u(t, j)) \approx \sum_{k=1}^{N_C^{\text{syn}}} \gamma_k^{\text{syn}}(t, j) P^{k \text{ syn}}(u(t, j)), \quad (57)$$

whereas the definition of $P^{k \text{ syn}}(u(t, j))$ is given in [\(41\)](#). Then a synthetic time series $\pi^{\text{syn}}(t, j)$ can be computed on the basis of the definition of the ensemble data by generating an ensemble of N_{ens} Markov chain realizations $\sigma^{\text{syn}}(t, j, l) \in \{s_1, \dots, s_{N_S}\}$ given the transition matrix $P^{\text{syn}}(t, j, u(t, j))$ (see [Algorithm 3](#)). The transition matrix $P^{\text{syn}}(t, j, u(t, j))$ is calculated using the assumed model structure given in [\(41\)](#) and [\(57\)](#) (see line 5). Further, it is assumed that $P^{\text{syn}}(t, j, u(t, j))$ also depends linearly on the implicit external factors $u^{\text{unres}}(t, j)$, given for the

⁸For more information on this approximation of the transition matrix $P^{\text{syn}}(t, j, u(t, j))$, see elucidations in [Section 4](#), or for a more detailed discussion on the matter (for purely time-dependent model parameters), the reader is referred to [[30](#)].

```

input : Choose values for  $N_K^{\text{syn}}, N_C^{\text{syn}}, N_T$  and  $N_J$ .
output :  $\Gamma^{\text{syn}}(t, j)$ 
1 for  $j = 1 : N_J$  do
2    $\gamma_k^{\text{syn}}(:, j) = [] \forall k \in \{1, \dots, N_K\}$ 
3   for  $c = 1 : N_C^{\text{syn}}$  do
4      $N_{\text{dummy}} = \text{round}(2N_T / (N_C^{\text{syn}} * \text{rand}([0, 1])))$ 
5      $\text{dummy0} = (0, \dots, 0) \in \mathbb{R}^{1 \times N_{\text{dummy}}}$ 
6      $\text{dummy1} = (1, \dots, 1) \in \mathbb{R}^{1 \times N_{\text{dummy}}}$ 
7      $r = \text{rand}(\{1, \dots, N_K^{\text{syn}}\})$ 
8     for  $k = 1 : N_K^{\text{syn}}$  do
9       if  $r == k$  then
10         $\gamma_k^{\text{syn}}(:, j) = [\gamma_k^{\text{syn}}(:, j) \text{ dummy1}]$ 
11       else
12         $\gamma_k^{\text{syn}}(:, j) = [\gamma_k^{\text{syn}}(:, j) \text{ dummy0}]$ 
13     if  $\text{length}(\gamma_1^{\text{syn}}(:, j)) \geq N_T$  then
14        $\gamma_k^{\text{syn}}(:, j) = \gamma_k^{\text{syn}}(1 : N_T, j) \forall k \in \{1, \dots, N_K^{\text{syn}}\}$ 
15     else
16        $N_{\text{dummy}} = N_T - \text{length}(\gamma_1^{\text{syn}}(:, j))$ 
17        $\text{dummy0} = (0, \dots, 0) \in \mathbb{R}^{1 \times N_{\text{dummy}}}$ 
18        $\text{dummy1} = (1, \dots, 1) \in \mathbb{R}^{1 \times N_{\text{dummy}}}$ 
19        $\gamma_1^{\text{syn}}(:, j) = [\gamma_1^{\text{syn}}(:, j) \text{ dummy1}]$ 
20        $\gamma_k^{\text{syn}}(:, j) = [\gamma_k^{\text{syn}}(:, j) \text{ dummy0}] \forall k \in \{2, \dots, N_K^{\text{syn}}\}$ 
21    $\Gamma^{\text{syn}}(:, j) = [\gamma_1^{\text{syn}}(:, j), \dots, \gamma_{N_K^{\text{syn}}}^{\text{syn}}(:, j)]$ 

```

Algorithm 2: Generate synthetic affiliation $\Gamma^{\text{syn}}(t, j)$.

generation of artificial data. Hence, analogously to the synthetic model matrices $P_1^{k \text{ syn}}, \dots, P_{N_E}^{k \text{ syn}}$, corresponding to the explicit external factors $u(t, j)$, a set of matrices $P_{N_E+1}^{k \text{ syn}}, \dots, P_{N_E+N_I}^{k \text{ syn}}$, related to the unresolved factors $u^{\text{unres}}(t, j)$, is chosen for $k \in \{1, \dots, N_K^{\text{syn}}\}$.

In order to generate samples from a distribution, as necessary in lines 7–9 of [Algorithm 3](#), one can employ standard techniques such as rejection sampling (also known as the acceptance-rejection method) [6; 33; 38]. Finally, the artificial data $\pi^{\text{syn}}(t, j)$ can be computed considering the quotients $N_{s_i}(t, j)/N_{\text{ens}}$ first introduced in (1), which are assumed to be a good approximation of the probability $\pi^{\text{syn}}(t, j)$ for large N_{ens} . The affiliation $\gamma_k^{\text{syn}}(t, j)$ is generated with the following setting: $N_C^{\text{syn}} = 10$, $N_K^{\text{syn}} = 2$, $N_T = 400$, $N_J = 24$, $N_S = 2$, $N_E = 2$ and $N_I = 0$. The first explicit external influence $u_1(t, j)$ is set to be a time- and location-dependent sinus function. As the second factor, we use the average of the neighboring cell

input : Choose values for N_K^{syn} , $\Gamma^{\text{syn}}(t, j)$ for all t and j (already generated), N_T , N_J , N_E , N_I , N_S and N_{ens} . Define synthetic model matrices $P_0^{k \text{ syn}}, \dots, P_{N_E}^{k \text{ syn}}, P_{N_E+1}^{k \text{ syn}}, \dots, P_{N_E+N_I}^{k \text{ syn}}$ with $k \in \{1, \dots, N_K^{\text{syn}}\}$, a finite set of discrete states $\{s_1, \dots, s_{N_S}\}$ and explicit as well as implicit external factors, i.e., $u(t, j)$ and $u^{\text{unres}}(t, j)$.

output: $\pi^{\text{syn}}(t, j)$

```

1 Initialize  $\sigma^{\text{syn}}(0, j, l) = \mathbf{rand}\{s_1, \dots, s_{N_S}\} \forall j \in \{1, \dots, N_J\}, l \in \{1, \dots, N_{\text{ens}}\}$ 
2 for  $t = 1 : N_T$  do
3   for  $j = 1 : N_J$  do
4      $P^{\text{syn}}(t, j, u(t, j)) =$ 
        $\sum_{k=1}^{N_K} \gamma_k(t, j) (P_0^{k \text{ syn}} + \sum_{e=1}^{N_E} P_e^{k \text{ syn}} u_e(t, j) + \sum_{e=1}^{N_I} P_{N_E+e}^{k \text{ syn}} u_e^{\text{unres}}(t, j))$ 
5     for  $l = 1 : N_{\text{ens}}$  do
6        $h = \mathbf{index}(\sigma^{\text{syn}}(t-1, j, l))$ 
7        $\sigma^{\text{syn}}(t, j, l) = \begin{cases} s_1 & \text{with probability } \{P^{\text{syn}}(t, j, u(t, j))\}_{h1}, \\ \vdots \\ s_{N_S} & \text{with probability } \{P^{\text{syn}}(t, j, u(t, j))\}_{hN_S} \end{cases}$ 
8       (see rejection sampling [6; 33; 38])
9     for  $i = 1 : N_S$  do
10       $\pi_i^{\text{syn}}(t, j) = \mathbf{counter}(\sigma^{\text{syn}}(t, j, l) = s_i) / N_{\text{ens}}$ 

```

Algorithm 3: Generate synthetic data $\pi^{\text{syn}}(t, j)$.

states at the previous time step, i.e.,

$$u_2(t, j) := \mathbf{average}_{r \in \text{neigh}(j)}(\pi(t, r)). \quad (58)$$

It allows us to model the spatial relations and to evaluate the statistical impact of adjacent location states. To be able to speak of neighbors in the spatial sense, a honeycomb lattice is assumed and each hexagon is assigned to one location. The choice of this lattice allows us to work with six neighbors for every location, each sharing an edge with the considered cell. To generate the data, we define matrices

$$P_0^{1 \text{ syn}} = \begin{bmatrix} 0.7 & 0.3 \\ 0.7 & 0.3 \end{bmatrix}, \quad P_1^{1 \text{ syn}} = \begin{bmatrix} 0.28 & -0.28 \\ 0.28 & -0.28 \end{bmatrix}, \quad P_2^{1 \text{ syn}} = \begin{bmatrix} -0.01 & 0.01 \\ -0.01 & 0.01 \end{bmatrix} \quad (59)$$

and

$$P_0^{2 \text{ syn}} = \begin{bmatrix} 0.3 & 0.7 \\ 0.3 & 0.7 \end{bmatrix}, \quad P_1^{2 \text{ syn}} = \begin{bmatrix} 0.24 & -0.24 \\ 0.24 & -0.24 \end{bmatrix}, \quad P_2^{2 \text{ syn}} = \begin{bmatrix} 0.05 & -0.05 \\ 0.05 & -0.05 \end{bmatrix}. \quad (60)$$

The primary focus of this example lies on checking the techniques' attributes. This includes the ability to infer good (i.e., unbiased) approximations of the model

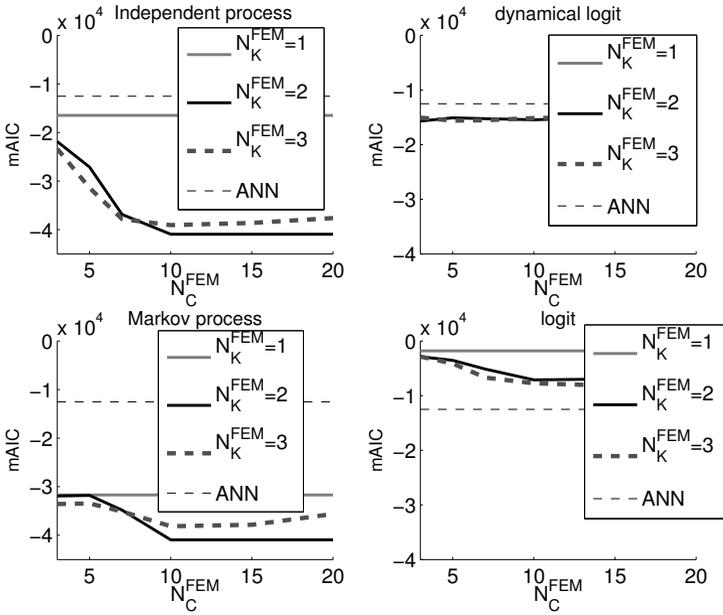


Figure 2. The four panels display the mAIC values for different parameters $N_K^{\text{FEM}} \in \{1, 2, 3\}$ and $N_C^{\text{FEM}} \in \{3, 5, 7, 10, 15, 20\}$ whereas each panel corresponds to a different model ansatz: Markov model, independent process, logistic model and dynamical logistic model. Additionally, the mAIC value calculated for the ANN results is shown.

parameters (i.e., $\Gamma^{\text{syn}}(t, j)$, $P^{k \text{ syn}}(u(t, j))$, N_K^{syn} and N_C^{syn}) as well as to generate a qualitative estimate of the distribution $\pi^{\text{syn}}(t, j)$. The proposed framework (four different direct model functions are considered, i.e., Markov and logit both with and without memory) is applied to the training sequence $\{1, \dots, 360\}$ (i.e., $N_{T_{\text{train}}} = 360$) of the synthetic data $\pi^{\text{syn}}(t, j)$ and the subspace algorithm is iterated $N_{\text{anneal}}^{\text{FEM}} = 10$ (for all four model assumptions) times in order to find a global minimum.⁹ The calculation is done for different parameters values $N_K^{\text{FEM}} \in \{1, 2, 3\}$ and $N_C^{\text{FEM}} \in \{3, 5, 7, 10, 15, 20\}$. Further, the corresponding mAIC values are computed with the proposed adapted information criterion. The resulting values are displayed in Figure 2. As can be seen in the panels on the left side of Figure 2, the mAIC values for the originally chosen maximal number of transitions N_C^{syn} and number of regimes N_K^{syn} are the lowest for the Markov framework with and without memory (i.e., the variables $N_C^{\text{Markov}} = N_C^{\text{syn}}$ and $N_K^{\text{Markov}} = N_K^{\text{syn}}$ are correctly identified). The results for the runs with logistic model assumptions (again with and without memory) have much bigger mAIC values (displayed in the panels on the right-hand side of Figure 2). Moreover, the mAIC value corresponding to the results of a neural network run

⁹For the remainder of the paper, we denote the AIC-optimal parameters computed by the framework with a superscripted *.

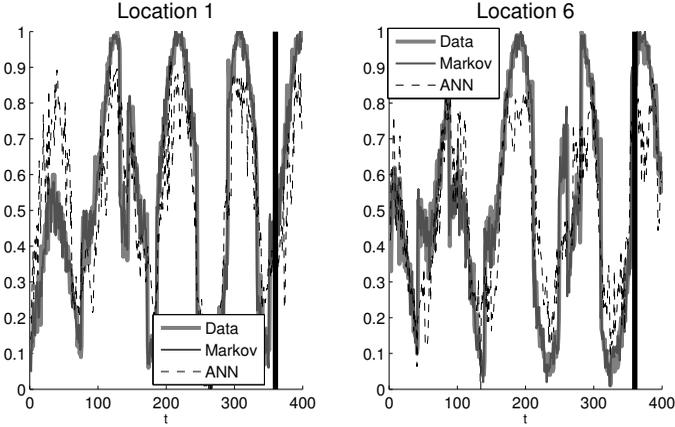


Figure 3. Approximations of the synthetic data $\pi_1^{\text{syn}}(t, j)$ retrieved with two different techniques: ANN (settings: $N_{\text{neurons}}^{\text{ANN}} = 20$, Levenberg–Marquardt backpropagation and $N_{\text{anneal}}^{\text{ANN}} = 10$) and nonstationary Markov regression (settings: no memory, $N_{\text{anneal}}^{\text{Markov}} = 10$, $N_C^{\text{Markov}} = 10$ and $N_K^{\text{Markov}} = 2$) are presented. Each of the panels corresponds to a location. The vertical black line at time $N_{\text{train}} = 360$ marks the last data point of the training data and the beginning of prediction sequence. The ANN approximation $\pi_1^{\text{ANN}}(t, j)$ is shown as a thin dashed line, and the approximation $\pi_1^{\text{Markov}}(t, j)$ obtained with the Markov model is displayed as a thin solid line.

(details below) is also presented in each of the four panels. The calculated model parameters of the Markov process without memory applied to the synthetic data for $N_K^{\text{Markov}} = 2$ and $N_C^{\text{Markov}} = 10$ are used to simulate $\pi^{\text{Markov}}(t, j)$ employing [Algorithm 3](#) with parameters $\Gamma^*(t, j)$ and $P^*(u(t, j))$ (see [Figure 3](#)). It is compared to results obtained with artificial neural networks (ANN) [[2](#); [24](#); [18](#); [3](#)] and support vector machines (SVM) [[8](#); [35](#)]. These techniques are popular pattern-recognition algorithms and can both be used to model spatiotemporal data. As a representative ANN, we consider a feedforward network, more specifically a multilayer perceptron (MLP) [[3](#)]. According to the theory, a network of this particular architecture with two hidden layers can be used to approximate an arbitrary nonlinear function [[23](#)]. For many cases, a single-layer network (with an arbitrary depth, i.e., number of neurons) is enough and can already describe most of the practically relevant functions [[18](#)]. Typically used transfer function classes are linear-, step- or sigmoid-functions. Multilayer feedforward networks with logistic sigmoid transfer functions are universal approximators [[18](#)], and therefore, we will deploy this type of ANN in the numerical tests below. We train networks with different numbers of hidden neurons and continue with the network that has the smallest residuals ($N_{\text{neurons}}^{\text{ANN}} = 20$). This means that a particular ANN with $N_{\text{neurons}}^{\text{ANN}} = w$ neurons is considered to be the best fit when $\sum_{t,j} \|\pi^{\text{syn}}(t, j) - \pi^{\text{ANN}(w)}(t, j)\|_2^2 \leq \sum_{t,j} \|\pi^{\text{syn}}(t, j) - \pi^{\text{ANN}(v)}(t, j)\|_2^2$ for all of the regarded neuron numbers $v, w \in [5, 10, 15, 20, 25, 30, 40, 50]$. The

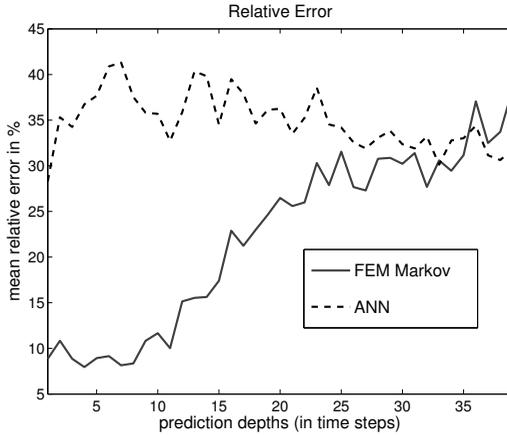


Figure 4. The mean relative error in % (in Euclidean metric) is shown dependent on the prediction depth $\tau \in \{1, \dots, N_{\text{pred}}\}$ (note that $N_{\text{pred}} = 39$). More specifically, the shown prediction error is computed as follows: $\text{mean}_j(\varpi(j, \tau) / \|\pi^{\text{syn}}(N_{T_{\text{train}}} + \tau, j)\|_2^2) * 100$ with $j \in \{1, \dots, N_j\}$ (details can be found in [Algorithm 4](#)).

Levenberg–Marquardt backpropagation is employed to optimize the network, and since it only converges to a local minimum, we also use annealing steps ($N_{\text{anneal}}^{\text{ANN}} = 10$) to approach a global solution.

An attempt to reconstruct the synthetic data $\pi_1^{\text{syn}}(t, j)$ for the entire time sequence, i.e., $t \in \{1, \dots, 400\}$, with the two different techniques, namely ANN and the Markov regression proposed in [Section 3](#), is shown in [Figure 3](#). Regarding the test sequence $\{1, \dots, N_{T_{\text{train}}}\}$, the approximation $\pi_1^{\text{ANN}}(t, j)$ (see the thin dashed line in the panels of [Figure 3](#)), corresponding to ANN, mostly follows the original path $\pi_1^{\text{syn}}(t, j)$. The performance of the ANN framework is also satisfactory when confronted with the test data (i.e., external factors $u(t, j)$ with $t \in \{361, \dots, 400\}$, starting from the thick black vertical line in both panels of [Figure 3](#)). The Markov regression technique (see the thin solid line in panels of [Figure 3](#)) restores the original series in the training sequence, i.e., in the first 360 time steps, more accurately than the ANN. In pursuance of approximating $\Gamma^{\text{syn}}(t, j)$ for $t \in \{N_{T_{\text{train}}}, \dots, N_T\}$, the self-contained strategy outlined in [Section 4](#) is employed. Details of the procedure to obtain $\hat{\pi}^{\text{Markov}}(t, j)$, i.e., approximating the synthetic data for the test sequence, can be found in the pseudocode of [Algorithm 4](#).

As can be seen in [Figure 3](#) (right from the vertical black line), the nonstationary nonhomogeneous Markov regression provides a high quality approximation $\hat{\pi}^{\text{Markov}}(t, j)$ of the artificial time series $\pi^{\text{syn}}(t, j)$ in the test sequence. The quality of the calculated model can also be accessed comparing the estimated local Markov parameters matrices with the synthetic ones $P_0^k, \dots, P_{N_E}^k$ with

input : $\Gamma^*(t, j)$ for $t \in \{1, \dots, N_{T_{\text{train}}}\}$, set maximal prediction depth N_{pred} and $u(t, j)$ for $t \in \{1, \dots, N_T\}$
output: $\varpi(j, \tau)$ with $\tau \in \{1, \dots, N_{\text{pred}}\}$ and $\hat{\pi}^{\text{Markov}}(t, j)$ with $t \in \{N_{T_{\text{train}}} + 1, \dots, N_T\}$

- 1 **for** $j = 1 : N_J$ **do**
- 2 Determine model parameter $P^\Gamma(u(t, j))$ characterizing the underlying model of $\Gamma^*(t, j)$ via stationary Markov regression.
- 3 **for** $\tau = 1 : N_{\text{pred}}$ **do**
- 4 $\hat{\Gamma}(N_{T_{\text{train}}} + \tau, j) = \Gamma^*(N_{T_{\text{train}}}, j) \prod_{h=0}^{\tau-1} P^\Gamma(u(N_{T_{\text{train}}} + h, j))$ (see (53))
- 5 Generate $\hat{\pi}(N_{T_{\text{train}}} + \tau, j)$ employing [Algorithm 3](#) (lines 3 to 10) using $\hat{\Gamma}(N_{T_{\text{train}}} + \tau, j)$.
- 6 $\varpi(j, \tau) = \|\pi^{\text{syn}}(N_{T_{\text{train}}} + \tau, j) - \hat{\pi}^{\text{Markov}}(N_{T_{\text{train}}} + \tau, j)\|_2^2$

Algorithm 4: Prediction.

$k \in \{1, \dots, N_K^{\text{syn}}\}$ (given in (59) and (60)) that have been used to generate the data

$$P_0^{1\text{Markov}} = \begin{bmatrix} 0.6999 & 0.3001 \\ 0.3001 & 0.6999 \end{bmatrix}, \quad P_1^{1\text{Markov}} = \begin{bmatrix} 0.2801 & -0.2801 \\ 0.2801 & -0.2801 \end{bmatrix}, \quad (61)$$

$$P_2^{1\text{Markov}} = \begin{bmatrix} -0.0125 & -0.0515 \\ -0.0125 & -0.0515 \end{bmatrix}$$

and

$$P_0^{2\text{Markov}} = \begin{bmatrix} 0.3003 & 0.69971 \\ 0.3003 & 0.69971 \end{bmatrix}, \quad P_1^{2\text{Markov}} = \begin{bmatrix} 0.24 & -0.24 \\ 0.24 & -0.24 \end{bmatrix}, \quad (62)$$

$$P_2^{2\text{Markov}} = \begin{bmatrix} 0.0515 & -0.0515 \\ 0.0515 & -0.0515 \end{bmatrix}.$$

Furthermore, the error plot of [Figure 4](#) also indicates the superiority of the Markov model in terms of relative prediction error

$$\varpi_{\text{rel}}(\tau) = \underset{j}{\text{mean}}(\varpi(j, \tau) / \|\pi^{\text{syn}}(N_{T_{\text{train}}} + \tau, j)\|_2^2) * 100 \quad (63)$$

up to a prediction depth of approximately 23 time steps ahead. The computation of the error term $\varpi(j, \tau)$ is explained in [Algorithm 4](#). An alternative possibility to model the discrete/categorical processes is provided by the support vector machines. SVMs are used for the classification of a given data set $u(t, j)$ with $t \in \{1, \dots, N_T\}$ and $j \in \{1, \dots, N_J\}$ with respect to a set of different *classes* (or states) $\{s_1, \dots, s_{N_S}\}$. This is achieved via geometrical separation, i.e., appropriate placing of hyperplanes in \mathcal{U} , dividing the values $u(t, j)$ in N_S different segments, thus associating $u(t, j)$ for each t and j with one class/state s_i . In the training phase, the assignment of the data values $u(t, j)$ to the classes is computed according to the values of the discrete process $\sigma^{\text{syn}}(t, j, l)$. As the microscopic information about the discrete states of the

process is unavailable, a threshold of 0.5 is set and $\pi^{\text{syn}}(t, j)$ is rounded accordingly so that the data has two categories, i.e., two classes. The optimization problem corresponding to the SVMs can be formulated as a quadratic minimization procedure resulting in a unique robust solution. In contrast, the ANNs (that are fitted through a nonconvex gradient-based optimization procedure) do not provide a unique solution of the inverse problem and therefore are in general less robust than SVMs. Different kernel functions are considered (specifically linear, quadratic, polynomial and radial basis functions), and the best fit (again regarding the residuals) was obtained for the radial basis function. The SVM run takes less computing time than the MLP run but needs a lot of support vectors to characterize the process. This overfitting is reflected in the very big $\text{mAIC} = 3.5193 \times 10^4$ value. In general, the computational complexity of SVMs with Gaussian radial basis function kernel (in the worst case) is $\mathcal{O}(N_T^2 N_E)$ for the training of each location [4]. But in most of the cases, it is possible to considerably reduce the computation time, e.g., by working with small values of the regularization parameter N_C^{SVM} for a faster convergence or, alternatively, increasing the number of training samples [34]. Determination of an optimal feedforward network with a nonlinear transfer function for a set of considered training data also requires solving a sequence of quadratic optimization problems. For the ANN calculations in this paper, we employed the Levenberg–Marquardt backpropagation algorithm, which is known to be very efficient [14]. Note, however, that the technique scales badly with the number of involved weights $N_{\text{weights}}^{\text{ANN}} = N_{\text{neurons}}^{\text{ANN}} (N_E + N_E^2 + \text{biases})$ (it is necessary to compute the inverse of the $N_{\text{weights}}^{\text{ANN}} \times N_{\text{weights}}^{\text{ANN}}$ Hessian matrix in each iteration step, which has a complexity $\mathcal{O}((N_{\text{weights}}^{\text{ANN}})^3)$). It is advisable to switch to a different gradient-descent algorithm for high-dimensional systems (i.e., systems that require more than a couple hundred weights) [39]. Further, the ANN fitting requires a longer run time due to the necessary annealing steps.

The SVM results are visualized in Figure 5 along with the approximations determined with the nonstationary Markov regression and the neural network (settings like in Figure 3). The assignment calculated with the SVM in general corresponds to the original data. Wrong categorization in the form of single outliers is mostly caused by data values too close to the threshold 0.5. Longer periods of wrong association especially in the test time sequence suggest that support vector machines are not feasible for prediction of spatiotemporal data of this particular nature.

Summing up, the proposed regression framework provides feasible and qualitative results. Nevertheless, it is important to mention that the considered synthetic data in this section is inherently designed to suit the model technique. The aim here was not to prove the overall superiority of the proposed algorithm in comparison to standard methods like ANN and SVM but to give the reader an idea of its capabilities under “good” conditions and as a contrast to the ill-posed example with missing external factors outlined in the next section.

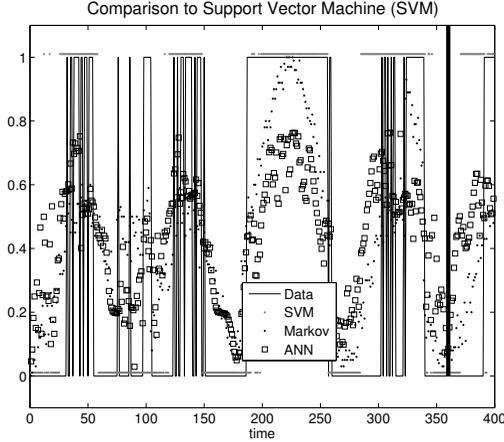


Figure 5. Dotted approximations of $\pi_1^{\text{syn}}(t, j)$ (for one fixed location) determined with a nonstationary Markov regression and a feedforward neural network and an output of support vector machines are shown. The beginning of the predicted time series is marked with a vertical black line.

5B. Example with missing (implicit) external factors. A key conceptual advantage of the proposed Markov regression framework is that implicit external factors, influencing the data, can be reflected in the nonstationary and nonhomogeneous formulation of the model. In order to numerically investigate this property, the framework is applied to a synthetic time series $\pi^{\text{syn}}(t, j)$ ($N_S = 2$) generated employing [Algorithm 3](#) with the number of implicit external factors set to $N_I = 100$ and the number of regimes fixed to be one ($N_K^{\text{syn}} = 1$), i.e., the artificial system is stationary and homogeneous and influenced by forces $u^{\text{unres}}(t, j)$ not available as observations.¹⁰ For the construction, we choose one explicit external factor (computed as a mean of neighboring states of the previous time step) and 100 implicit influences in the form of sinus functions (randomly chosen between: $u_e^{\text{unres}}(t, j) := \sin^2((2\pi te)/360 + j/20)$ and $u_e^{\text{unres}}(t, j) := \cos^2((2\pi te)/360 + j/20)$) depending on time t ($N_T := 400$), location j ($N_J := 24$) and the index of the particular external factor $e \in \{1, \dots, N_I\}$. Further, the model matrices for the one considered regime are defined:

$$P_0^{1 \text{ syn}} = \begin{bmatrix} 0.5 & 0.5 \\ 0.5 & 0.5 \end{bmatrix}, \quad P_1^{1 \text{ syn}} = \begin{bmatrix} 0.05 & -0.05 \\ 0.05 & -0.05 \end{bmatrix}, \quad P_2^{1 \text{ syn}} = \begin{bmatrix} 0.42 & -0.42 \\ 0.42 & -0.42 \end{bmatrix} \quad (64)$$

and

$$P_{e+2}^{1 \text{ syn}} = \begin{bmatrix} 0.0002 & -0.0002 \\ 0.0002 & -0.0002 \end{bmatrix} \quad \forall e \in \{1, \dots, N_I - 2\}. \quad (65)$$

¹⁰Note that it is not necessary to use [Algorithm 2](#) since $\Gamma^{\text{syn}}(t, j) := \mathbf{ones}(1, N_T, N_J)$.

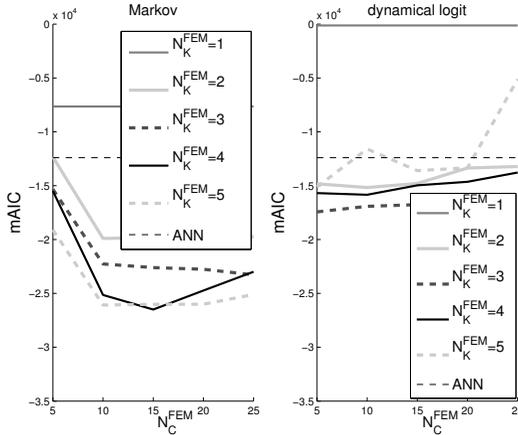


Figure 6. The mAIC values for runs of the Markov regression and the dynamical logistic regression applied to the second synthetic data series run for different values of N_C^{FEM} and N_K^{FEM} are displayed in this graph. Moreover, the value for the ANN result is shown.

The first implicit external factor $\bar{u}_2(t, j) = u_1^{\text{unres}}(t, j)$ thereby has the most significant influence, and all the other external factors have a much smaller impact. The proposed nonstationary nonhomogeneous Markov regression is applied to part of the generated data (i.e., $\pi^{\text{syn}}(t, j)$ with $t \in \{1, \dots, 360\}$ and $j \in \{1, \dots, 24\}$) for $N_K^{\text{FEM}} \in \{1, 2, 3, 4, 5\}$ and $N_C^{\text{FEM}} \in \{5, 10, 15, 20, 25\}$ with $N_{\text{anneal}}^{\text{FEM}} = 10$. Note that the implicit external factors $u^{\text{unres}}(t, j)$ are not made available for the regression procedure. The optimal model fit is determined via the modified information criterion (46). The resulting graphs can be seen in the left panel of Figure 6.

The lowest mAIC value has a model with up to 15 transitions between four regimes, i.e., $N_C^{\text{Markov}} = 15$ and $N_K^{\text{Markov}} = 4$. Thus, the synthetic stationary homogeneous model is described with a nonstationary and nonhomogeneous model capturing the original process and reflecting the implicit external factors $u^{\text{unres}}(t, j)$. In contrast, the dynamical logistic regression, applied to the data set, has bigger mAIC values and hence represents a worse description for the analyzed data. Two approximations of the ensemble distribution $\pi^{\text{syn}}(t, j)$ for different locations are shown in Figure 7. The plots illustrate that the nonstationary nonhomogeneous Markov regression is feasible even for observations where the biggest part of the relevant information is not provided in the form of measurements. The data $\pi^{\text{syn}}(t, j)$ in the test sequence, i.e., $t \in \{361, \dots, 400\} \forall j$, is approximated by computing a one-step prediction $\hat{\Gamma}(361, j)$ (see (53)) and using Algorithm 3 to determine $\hat{\pi}^{\text{Markov}}(361, j) \forall j$. Then the proposed Bayesian-update scheme is employed to update $\hat{\Gamma}^{\text{Markov}}(361, j)$ (see (56) in Proposition 4.1) using new data information $\pi^{\text{syn}}(361, j)$. These steps are iterated until $\hat{\pi}^{\text{Markov}}(N_T, j)$ can be calculated (note that the updated $\hat{\Gamma}(t, j)$ is used as the affiliation parameter $\Gamma^*(t, j)$

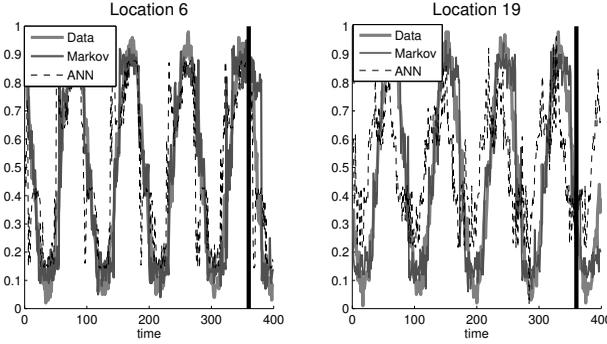


Figure 7. Each graph displays approximations of the data $\pi_1^{\text{syn}}(t, j)$ generated by means of different models, i.e., Markov regression (settings: no memory, $N_{\text{anneal}}^{\text{FEM}} = 10$, $N_{\text{C}}^* \text{Markov} = 15$, $N_{\text{K}}^* \text{Markov} = 4$) and an optimal ANN (settings: $N_{\text{neurons}}^{\text{ANN}} = 10$, Levenberg–Marquardt backpropagation, $N_{\text{anneal}}^{\text{ANN}} = 10$). The artificial time series $\pi_1^{\text{syn}}(t, j)$ is shown as a thick gray line. The start of the prediction is marked with a black vertical line at $N_{T_{\text{train}}} = 360$.

for all previous time steps t in the prediction sequence $\{361, \dots, 400\}$). The resulting prediction has a good quality as can be seen from the right-hand side of the vertical black line in the two panels of Figure 7.

In order to give an impression on the feasibility of standard techniques under “bad” conditions, such as artificially generated for this example, ANNs are applied to $\pi^{\text{syn}}(t, j)$ (settings: $N_{\text{neurons}}^{\text{ANN}} = 10$, Levenberg–Marquardt backpropagation and $N_{\text{anneal}}^{\text{ANN}} = 10$). The quality of the ANN results strongly depends on the location. This is caused by the dependence of the implicit external factors on the location; i.e., the implicit impact on the data is differing for each cell. In other words, the ANN framework does not allow restoring the devolution of the data without the additional information of the implicit external factors for location 19 and all other locations that are strongly influenced by the unresolved quantities. This is due to the fact that, in contrast to the nonstationary and nonhomogeneous Markov regression model presented above, the parameters (such as neuron weights and biases) of the standard ANN are time- and location-independent. In other words, ANN as well as SVM represent intrinsically stationary and homogeneous models. Because of this reason, both ANN and SVM as model classes have difficulties in capturing the effects of unobserved external factors. Concluding, it is possible to obtain qualitative results with ANN for the constructed dynamical system when enough information is provided in the form of data (see Section 5A) but is not a reliable option for realistic systems with data availability problems.

5C. Assimilation of additional information. The purpose of this example is to demonstrate the application of the Bayesian-update scheme (see Proposition 2.1

in Section 4) when compared to a simple maximum-likelihood allocation of new data (see (54)) or machine-learning algorithms like SVMs or ANNs. To this end, a transition path $\Gamma^{\text{syn}}(t, j)$ (employing Algorithm 2) of length $N_T = 10000$, switching between $N_K^{\text{syn}} = 2$ local models and $N_C^{\text{syn}} = 5$ transitions, was generated for $N_J = 10$ different locations. This path was then used to generate a time series, switching between $N_S = 2$ discrete values s_1 and s_2 without external influences (i.e., $N_F = 0$ and $N_{\text{ens}} = 1$) according to the following rules:

- (1) In the first model θ_1 , the process at time t is modeled by a Bernoulli-random variable with a probability 0.6 to be in the state s_1 .
- (2) For the second model θ_2 , a Markov chain is used to obtain the value of $\sigma(t, j, l)$; here the probability for the next value to be in different state than the previous value is 0.3.

For the training of the model, the natural choice for this example is the nonstationary nonhomogeneous Markov regression model (as introduced in (34)); the first 9000 time steps are chosen as a training set (i.e., $N_{T_{\text{train}}} = 9000$). To obtain a statistically significant result, the analysis is done not only for one but for 200 different time series (as already mentioned, $N_{\text{ens}} = 1$), all sharing the same transition path $\Gamma^{\text{syn}}(t, j)$. This allows us to draw first statistical conclusions and make the comparison of different methods independent of a single stochastic realization of the process. Since the focus is on the statistical significance rather than on the size of the ensemble, it is necessary to interpret the outcome of a single observation as the corresponding ensemble data, i.e.,

$$\pi_i^{\text{syn}}(t, j) = \delta_{s_i}(\sigma^{\text{syn}}(t, j, l)) \quad \text{for } i \in \{1, 2\}, \quad (66)$$

where δ_{s_i} is the Kronecker delta for the value s_i (i.e, being one if s_i is observed, else zero). To predict the incoming values of the time series ($t > N_{T_{\text{train}}} = 9000$), one needs to predict the affiliation vector $\Gamma^*(t, j)$ first. To this end, a self-contained strategy, proposed in Section 4, is employed. In other words, a transition matrix P^Γ is fitted to the 9000 elements of the transition path. This Markov chain is then used to propagate the current distribution of the affiliation to the next step and so forth. Of course, this makes the prediction very sensitive to finding the correct affiliation of data points [30] not included in the initial analysis of the time series. To demonstrate this sensitivity, the updating procedure as in Section 4 is compared to an SVM, an ANN and the maximum-likelihood affiliation (defined in (54)) of the data points. The SVM and ANN are additionally provided with the previous observation as this is used in the other two assimilation methods as well; thus, all four methods can make use of the same input information. To this end, the dimension of the data is doubled by creating the vectors $[\pi^{\text{syn}}(t, j) \pi^{\text{syn}}(t-1, j)]$. Additionally, different kernel functions were tried for SVM and different transfer

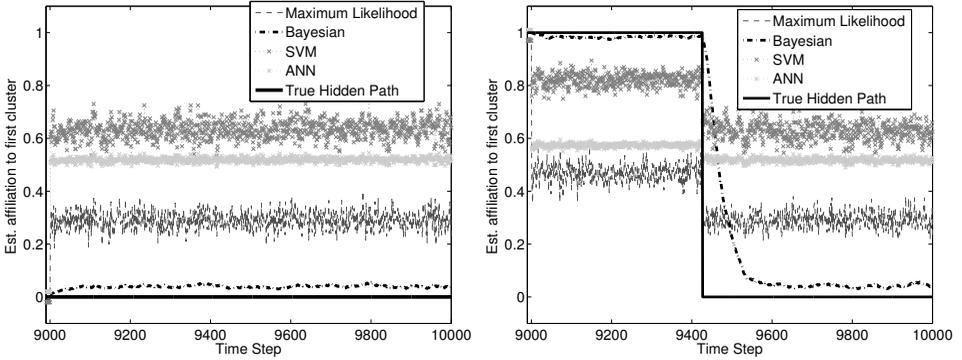


Figure 8. Average assimilation of 1000 untrained data points to the clusters for two different transition paths (in two different locations). The sample consisted of 200 different realizations of the time series with 9000 training points. To improve visibility, the allocations are shifted by up to 0.02. The beginning of the prediction is time step 9001.

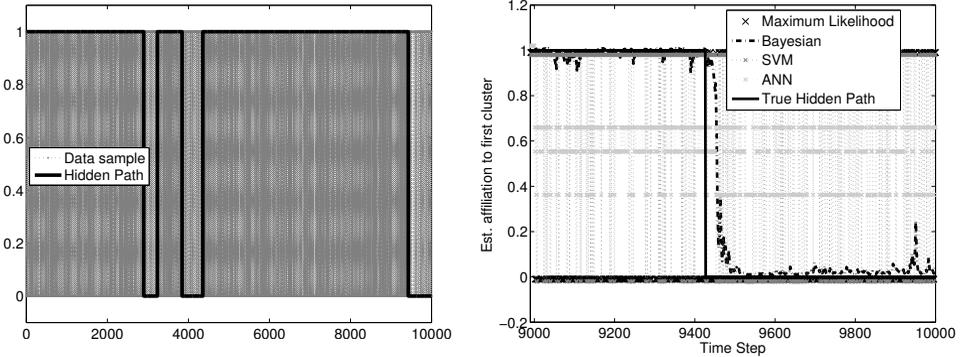


Figure 9. Left panel: Typical data set and the transition path used for the creation of the data. Right panel: Result of the assimilation schemes; only the relevant points are shown. The beginning of the assimilation is time step 9001. As can be seen, only the Bayesian assimilation scheme (black dashed line) based on Proposition 2.1 completely recovers the true persistent structure of the original hidden process (black solid line).

functions and numbers of neurons for the ANN; an optimal configuration in each model class was obtained applying the standard AIC procedure. Out of the 10 locations, two are shown here, one with constant original allocation in the prediction time frame (Figure 8, left panel) and one with a jump in the allocation (Figure 8, right panel). Additionally, a typical data set is shown in Figure 9 (left panel) and the affiliation functions resulting from the different assimilation methods are depicted in the right panel of Figure 9.

All four updating procedures generate affiliations that are not free of errors. To measure the quality of an allocation, the average distance

$$\frac{1}{(N_T - N_{T_{\text{train}}}) * N_J} \sum_{t=N_{\text{pred}}+1}^{N_T} \sum_{j=1}^{N_J} |\hat{\gamma}_1(t, j) - \gamma_1(t, j)|$$

of the estimated affiliation and the original path is averaged over all 200 realizations. Resulting error estimates are shown in [Table 1](#).

Algorithm	Error
maximum-likelihood affiliation	0.3142
Bayesian update (see Section 4)	0.0384
SVM-based affiliation	0.6188
ANN-based affiliation	0.4948

Table 1. Average distance of the affiliation of new data to the true path.

All estimators are then used to predict the next 10 time steps, i.e., $N_{\text{pred}} = 10$, according to the following algorithm:

input : data $\pi^{\text{syn}}(t, j)$, maximal prediction depth N_{pred} and the affiliation $\Gamma^*(t, j)$ for $t \in \{1, \dots, N_{T_{\text{train}}}\}$
output : $\hat{\pi}^{\text{Markov}}(t, j)$ and $\varpi(t, j, \tau)$ for $t \in \{N_{T_{\text{train}}} + 1, \dots, N_T\}$, $j \in \{1, \dots, N_J\}$ and $\tau \in \{1, \dots, N_{\text{pred}}\}$

- 1 Set start of test data $N_{T_{\text{train}}} = 9000$.
- 2 $\hat{\Gamma}(N_{T_{\text{train}}}, j) := \Gamma^*(N_{T_{\text{train}}}, j) \forall j$
- 3 **for** $j = 1 : N_J$ **do**
- 4 **for** $t = N_{T_{\text{train}}} : N_T - N_{\text{pred}}$ **do**
- 5 **for** $\tau = 1 : N_{\text{pred}}$ **do**
- 6 $\hat{\Gamma}(t + \tau, j) = \hat{\Gamma}(t, j) \prod_{h=0}^{\tau-1} P^\Gamma(u(t+h, j))$ (see (53))
- 7 Generate $\hat{\pi}(t + \tau, j)$ employing [Algorithm 3](#) (lines 3 to 10) using $\hat{\Gamma}(t + \tau, j)$.
- 8 $\varpi(t, j, \tau) = \|\pi_1^{\text{syn}}(t + \tau, j) - \hat{\pi}_1^{\text{Markov}}(t + \tau, j)\|_2^2$
- 9 Incorporate the observation $\pi^{\text{syn}}(t + 1, j)$ into the data set, and estimate its affiliation $\hat{\Gamma}(t + 1, j)$ for all j .

Algorithm 5: Prediction.

The quality of the prediction is measured by $\varpi(t, j, \tau)$, the squared distance of the synthetic data and the predicted probability for observing one (see line 8, [Algorithm 5](#)). These errors are then averaged for every τ over the 200 different realizations, the 10 locations and the prediction period.

As can be seen from [Table 1](#) and [Figure 10](#), the posterior estimators based on [Proposition 4.1](#) significantly outperforms other considered methods. Yet it should be noted that the process is rapidly mixing and thus hard to predict in the first place.

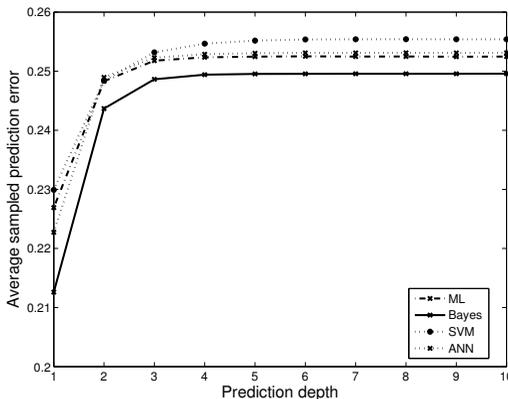


Figure 10. Mean of the sampled prediction errors for up to 10 time steps and four different assimilation schemes. The sample consisted of 200 different realizations of sets of 10 time series with length 1000.

This property increases the challenge all the assimilation methods have to face as the two different model-states are hard to separate even visually (see, e.g., the gray line in the left panel of [Figure 9](#)). Additionally, it should be noted that the average errors of the predictions for all four assimilation methods are rather similar; this is again a result of the low persistency of the rapidly mixing observed process. Nevertheless, the better assimilation of the missing information in the form of the affiliation function Γ (introduced in the current manuscript) leads to a reduction of the prediction error even for this very tough case, raising hope for better predictive models and better assimilation of the effects induced by the unresolved external factors as captured by the affiliation functions Γ .

6. Conclusion

The proposed nonstationary and nonhomogeneous regression framework represents a very promising way for modeling of spatiotemporal discrete jump processes under the presence of unobserved external impacts. As demonstrated in the current manuscript, it can capture the most significant impacts of the unobserved external factors described by [Proposition 2.1](#).

This was demonstrated by means of an example with additionally incorporated implicit external factors that were not made available for the calculation of the model parameters. Since incomplete data sets represent one of the central challenges in the field of time-series analysis, this property makes the presented methodology potentially useful in many areas of multiscale modeling and simulation, where discrete processes (e.g., associated with the phase transitions in physics) are subject to unresolved subgrid-scale effects.

Along the lines of traditional data assimilation, a new Bayesian algorithm to assimilate the model affiliation function $\widehat{\Gamma}(t, j)$ (capturing an impact of the unresolved external factors) was introduced and shows promising results. The proposed Bayesian algorithm for discrete data assimilation provides considerably better results than the currently available standard methods (i.e., maximum-likelihood assimilation, ANN and SVM) for the considered “tough” example of a rapidly switching nonstationary and nonhomogeneous discrete process.

It should be stressed that the adequacy of the presented models is largely relying on the validity of the underlying assumptions in [Proposition 2.1](#) as well as on the validity of the stationary homogeneous Markov assumption for the model-affiliation process (capturing an impact of unresolved external factors).

Because of this reason, in some situations, it might be necessary to use a nonstationary model formulation for the affiliation process and to include the additional necessary variables in the validation of the modified information criterion. In other words, in such situations, the optimal fit given by the nonstationary discrete regression model parameters and parametrization of the optimal spatiotemporal model for the hidden process Γ (beyond the stationary approximation deployed in this work) should be approached simultaneously. Although this new direction will allow constructing more realistic models with less a priori assumptions, it would also require many more computational resources than the proposed numerical framework. Numerical complexity estimates presented in this paper demonstrate that the deployment of concepts from high-performance computing and supercomputing computational facilities will also be necessary to extend all of the considered methods to realistic numbers of spatial locations and lengths of the time series. This issue is also a matter of the ongoing research.

Appendix A: Proof of [Proposition 2.1\(3\)](#)

Proof. Without loss of generality, we can assume that the external factors are ordered such that the explicit factors are the first N_E entries of $\bar{u}(t, j)$. By performing a Taylor expansion on the transition matrix $P(\bar{u}(t, j))$ around the means $\mu(t, j) = [\mathbf{E}(\bar{u}_1(t, j)), \dots, \mathbf{E}(\bar{u}_{N_E+N_I}(t, j))] \in \mathbb{R}^{(N_E+N_I) \times 1}$, we obtain

$$\begin{aligned}
 P(\bar{u}(t, j)) &= P(\mu(t, j)) + \sum_{e=1}^{N_E+N_I} \frac{\partial P(\mu(t, j))}{\partial \bar{u}_e(t, j)} (\bar{u}_e(t, j) - \mu_e(t, j)) \\
 &+ \sum_{e,h=1}^{N_E+N_I} \frac{\partial^2 P}{\partial \bar{u}_e(t, j) \partial \bar{u}_h(t, j)} (\bar{u}_e(t, j) - \mu_e(t, j)) (\bar{u}_h(t, j) - \mu_h(t, j)) \\
 &+ \sum_{|\alpha|=3} R_\alpha(\bar{u}(t, j)) (\bar{u}(t, j) - \mu(t, j))^\alpha, \tag{1}
 \end{aligned}$$

where α is a multi-index and

$$R_\alpha(\bar{u}(t, j)) = \frac{3}{\alpha!} \int_0^1 (1-x) D^\alpha P(\mu(t, j) + x(\bar{u}(t, j) - \mu(t, j))) dx. \quad (2)$$

Note that $R_\alpha(\bar{u}(t, j))$ is bounded as the third derivative of P is assumed to be bounded. Resorting the terms and defining

$$\rho_h(t, j) = 2 \sum_{e=N_E+1}^{N_E+N_I} \frac{\partial^2 P}{\partial \bar{u}_h(t, j) \bar{u}_e(t, j)} (u_e(t, j) - \mu_e(t, j)), \quad h = 1, \dots, N_E, \quad (3)$$

$$\begin{aligned} \check{\varepsilon}(t, j) &= \sum_{e=N_E+1}^{N_E+N_I} \frac{\partial P(\mu(t, j))}{\partial \bar{u}_e(t, j)} (\bar{u}_e(t, j) - \mu_e(t, j)) \\ &\quad + \sum_{|\alpha|=3} R_\alpha(\bar{u}(t, j)) (\bar{u}(t, j) - \mu(t, j))^\alpha \\ &\quad + \sum_{e, h=1}^{N_E} \frac{\partial^2 P}{\partial \bar{u}_h(t, j) \bar{u}_e(t, j)} (u_e(t, j) - \mu_e(t, j)) (u_h(t, j) - \mu_h(t, j)) \\ &\quad - \sum_{e=1}^{N_E} \mu_e(t, j) \rho_h(t, j) + \sum_{e, h=N_E+1}^{N_E+N_I} \frac{\partial^2 P}{\partial \bar{u}_h(t, j) \bar{u}_e(t, j)} \\ &\quad \quad \quad * (u_h(t, j) - \mu_h(t, j)) (u_e(t, j) - \mu_e(t, j)), \end{aligned} \quad (4)$$

$$\varepsilon(t, j) = \check{\varepsilon}(t, j) - \mathbf{E}[\check{\varepsilon}(t, j)] \quad (5)$$

yields (11) whereas the definition of matrices $P_h(t, j)$ is given in (14) for all t and j and $h \geq 1$ and $P_0(t, j)$ is defined as in (16), and replacing the expectation in the formula by the expectation of $\check{\varepsilon}(t, j)$. Moreover, $\mathbf{E}[\varepsilon(t, j)] = 0$ and $\mathbf{E}[\rho_h(t, j)] = 0$. \square

Appendix B: Notation

The notation index is organized as follows. The numbers and sizes are listed separately as their notation is very similar. The remaining notations are listed in order of appearance in the manuscript. To improve readability, the titles of sections and subsections are indicated. Moreover, the abbreviations used in the manuscript are listed at the end of the notation index.

Numbers and sizes.

- N_S total number of states s_i (associated index i , p. 3).
- N_{ens} (associated index l , p. 3).
- N_J space dimension of observations $\pi(t, j)$ for all time steps t (associated index j , p. 3).
- N_T length of observed time series $\pi(t, j)$ for fixed location j (associated index t , p. 3).
- $N_{s_i}(t, j)$ number of cells j currently (at time t) in state s_i (p. 3).

- N_E total number of explicit external factors (associated index e , p. 5).
- N_I total number of implicit external factors (associated index e , p. 5).
- N_F total number external factors (associated index e , p. 5).
- N_M memory depth (p. 8).
- N_K total number of local stationary homogeneous models θ_k (associated index k , p. 13).
- N_C maximal number of allowed transitions of the affiliation processes $\gamma_k(t, j)$ for fixed j .
- $N_{\text{anneal}}^{\text{FEM}}$ total number of annealing steps used for the FEM framework (p. 16).
- N_{ϕ_k} degree of a polynomial of parametric (conditional) probability density function ϕ_k (p. 18).
- N_{pred} prediction depth (p. 20).
- $N_{T_{\text{train}}}$ time-wise length of training data (p. 22).
- N_C^{syn} artificially chosen maximal number of transitions of the synthetic affiliation processes $\gamma_k^{\text{syn}}(t, j)$ (p. 24).
- N_K^{syn} artificially chosen total number of local stationary homogeneous models θ_k^{syn} (p. 24).
- N_K^{FEM} number of local regimes considered for the general FEM framework (p. 26).
- N_C^{FEM} number of maximal transitions considered for the general FEM framework (p. 26).
- N_{dummy} auxiliary quantity of Algorithm 2 (p. 24).
- $N_C^*_{\text{Markov}}$ optimal in terms of the mAIC values (with respect to the data) maximal number of transitions for the parameters computed with the Markov regression framework (p. 26).
- $N_K^*_{\text{Markov}}$ optimal in terms of the mAIC values (with respect to the data) maximal number of local stationary models computed with the Markov regression framework (p. 26).
- $N_{\text{neurons}}^{\text{ANN}}$ total number of employed neurons for an ANN run (p. 27).
- $N_{\text{anneal}}^{\text{ANN}}$ total number of annealing steps used for an ANN run (p. 28).
- $N_{\text{anneal}}^{\text{Markov}}$ total number of annealing steps used for the Markov regression (p. 27).
- $N_{\text{weights}}^{\text{ANN}}$ is the number of ANN parameters (p. 30).
- $N_{\text{basis}}^{\text{FEM}}$ number of finite elements used for the discretization (p. 16).
- N_C^{SVM} regularization parameter of SVM (p. 30).

Ensemble data and exterior quantities.

- s_i discrete state (p. 3).
- $\omega(j, l)$ microscopic cell (p. 3).
- $\sigma(t, j, l)$ with $j \in \{1, \dots, N_J\}$ and $l \in \{1, \dots, N_{\text{ens}}\}$ dynamical process of a microscopic cell $\omega(j, l)$ (p. 3).
- $\tilde{\pi}_i(t, j)$ empirical probability for process $\sigma(t, j, l)$ to be in state s_i in location $\omega(j, l)$ at time t (Definition (1), p. 3).
- $N_{s_i}(t, j)$ total number of microscopic cells $\omega(j, t)$ in state s_i for fixed t and j (Definition (2), p. 3).
- $\delta_{s_i}(\cdot)$ the Kronecker delta for the value s_i (p. 4).
- $\tilde{\pi}(t, j) \in [0, 1]^{N_s \times 1}$ vector of empirical probabilities (Definition (3), p. 4).
- $\pi_i(t, j)$ probability for process $\sigma(t, j, l)$ to be in state s_i in location $\omega(j, l)$ at time t (Definition (4), p. 4).

- $\mathbb{P}[\cdot]$ probability function.
- $\pi(t, j) \in [0, 1]^{N_S \times 1}$ vector of states probabilities (Definition (5), p. 4).

Implicit external factors.

- $P(\bar{u}(t, j)) \in [0, 1]^{N_S \times N_S}$ transition matrix dependent on all external factors (p. 4).
- $\bar{u}(t, j) \in \mathbb{R}^{(N_E + N_I) \times 1}$ all influencing external factors (Definition (7), p. 5).
- \mathbb{R} real numbers.
- $u_e(t, j) \in \mathbb{R}$ explicit external factor.
- $u(t, j) \in \mathbb{R}^{N_E \times 1}$ vector of explicit external factors (Definition (8), p. 5).
- $\mathcal{U} \subset \mathbb{R}^{N_E \times 1}$ vector space of explicit external factors $u(t, l)$.
- $u_e^{\text{unres}}(t, j) \in \mathbb{R}$ implicit external factor.
- $u^{\text{unres}}(t, j) \in \mathbb{R}^{N_I \times 1}$ vector of implicit external factors (Definition (9), p. 5).
- $\varepsilon(t, j)$ error term associated with decomposition of transition matrix $P(\bar{u}(t, j))$ (Definition (15), p. 7).
- $\mathbf{E}(\cdot)$ expected value.
- $\rho_e(t, j)$ second noise process for decomposition of $P(\bar{u}(t, j))$ with second derivatives. (Definition (3), Appendix A, p. 39).
- $\mu(t, j) \in \mathbb{R}^{N_I \times 1}$ vector of expected values for each of the entries of vector $\bar{u}(t, l)$ (p. 7).
- $R_\alpha(\bar{u}(t, j))$ Taylor-expansion error component (Definition (13), p. 7).
- α a multi-index (p. 7).
- $P_e(t, j)$ matrix used in the linear combination equal to $P(t, l, u(t, l))$ corresponding to $u_e(t, j)$ for $e \in \{1, \dots, N_S\}$ (Definition (14), p. 7).
- $P_0(t, j)$ matrix used in the linear combination equal to $P(t, l, u(t, l))$ (Definition (16), p. 7).
- $P(t, j, u(t, j)) \in [0, 1]^{N_S \times N_S}$ equal to $P(\bar{u}(t, j))$ assuming the conditions of Proposition 2.1 are fulfilled.

Inverse problem formulation.

- $f(\cdot)$ a general direct mathematical model (Definition (17), p. 8).
- $\theta(\bar{u}(t, j))$ unknown model parameter dependent on all external factors (p. 8).
- Ω parameter space containing $\theta(\bar{u}(t, j))$ (p. 8).
- $\lambda(t, j)$ error term of simple model example (p. 9).
- $g(\cdot)$ model distance function (Definition (19), p. 9).
- $\mathbf{L}(\cdot)$ averaged clustering functional (Definition (20), p. 9).
- $d(\cdot, \cdot)$ metric (p. 9).
- $d_2(\cdot, \cdot)$ Euclidean metric (p. 9).
- f^{logit} logistic direct mathematical model function (p. 9).
- f^{Markov} Markov direct mathematical model function (p. 9).

Logistic regression.

- $\mathcal{C}_i[u(t, j), B^i(t, j)]$ utility measure (Definition (23), p. 10).

- $B^i(t, j) \in \mathbb{R}^{(N_E+1) \times 1}$ logistic model parameter corresponding to state s_i for $i \in \{1, \dots, N_S\}$ (Definition (24), p. 10).
- $\beta_e^i(t, j)$ e -th entry of vector $B^i(t, j)$.
- $\xi^i(t, j)$ error process of utility measure (p. 10).
- $B(t, j) \in \mathbb{R}^{(N_E+1) \times N_S}$ nonstationary nonhomogeneous logistic model parameter (p. 11).
- $\theta^{\text{logit}}(B(t, j), u(t, j))$ logistic model parameter (Definition (28), p. 11).
- $\zeta(t, j)$ error term of logistic model distance function (p. 11).

Interpolation.

- $\theta_k(u(t, j))$ stationary homogeneous model parameter (p. 13).
- $\gamma_k(t, j)$ weighting process corresponding to local model $\theta_k(u(t, j))$ (p. 13).
- $\Theta(u(t, j))$ vector of stationary homogeneous model parameters (p. 13).
- $\Gamma(t, j) \in [0, 1]^{1 \times N_K}$ vector of affiliation processes (p. 13).
- $\mathbf{L}(\cdot, \cdot)$ interpolated version of averaged clustering functional $\mathbf{L}(\cdot)$ (Definition (38), p. 13).
- $\mathbf{L}_j(\cdot, \cdot)$ one summand for a fixed location j of interpolated average clustering functional (Definition (39), p. 14).
- $B_k \in \mathbb{R}^{(N_E+1) \times N_S}$ local logit model parameter (p. 14).
- B_k^i i -th entry of stationary and homogeneous logit model parameter vector B_k (p. 14).
- $P^k(u(t, j))$ local Markov model parameter matrix (Definition (41), p. 14).
- $P(u(t, j)) \in \mathbb{R}^{N_S \times N_S N_K}$ vector of model matrices $P^k(u(t, j))$ (p. 14).
- $P_0^k, \dots, P_{N_E}^k$ for all k matrices used in the linear combination equal to $P^k(u(t, l))$ (p. 14).
- $\mathbf{1}$ auxiliary column vector containing only entries equal to one (p. 14).
- $\mathbf{0}$ auxiliary column vector containing only entries equal to zero (p. 14).
- $\{P_e^k\}_{n,m}$ entry of matrix P_e^k in n -th row and m -th column (p. 15).

Spatial and temporal persistence.

- $|\cdot|_{\text{BV}(1, N_T)}$ bounded variation (BV) half-norm (Definition (45), p. 15).

Numerical approach and computational complexity.

- $\Gamma^*(t, j) = [\gamma_1^*(t, j), \dots, \gamma_{N_K}^*(t, j)] \in [0, 1]^{1 \times N_K}$ global optimizer with respect to $\mathbf{L}(\Gamma(t, j), \Theta(u(t, j)))$ (p. 15).
- $\Theta^*(u(t, j))$ global optimizer with respect to $\mathbf{L}(\Gamma(t, j), \Theta(u(t, j)))$ (p. 15).
- Γ_r computed Γ process dependent on annealing index (p. 17).
- $\Gamma_r^{[h]}$ determined Γ process dependent on annealing index and optimization iteration index (p. 17).
- Θ_r computed model parameter Θ dependent on annealing index (p. 17).
- $\Theta_r^{[h]}$ determined model parameter Θ dependent on annealing index and optimization iteration index (p. 17).
- \mathbf{L}_{\min} auxiliary variable of Algorithm 1 (p. 17).
- κ auxiliary variable used to describe the order of the computational costs (p. 16).

Information criterion.

- $\text{mAIC}(\cdot, \cdot)$ modified version of Akaike information criterion for presented framework (Definition (46), p. 18).
- $\mathcal{L}(\cdot, \cdot)$ log-likelihood (Definition (47), p. 18).
- $\phi_k(\cdot, \dots, \cdot | N_{\phi_k})$ parametric (conditional) probability density function (PDF) (p. 18).
- $M(\cdot, \cdot)$ function computing total number of involved parameters (p. 18).
- $M^{\text{logit}}(\cdot, \cdot)$ function computing total number of involved parameters for a logistic model (Definition (48), p. 18).
- $M^{\text{Markov}}(\cdot, \cdot)$ function computing total number of involved parameters for Markov model (Definition (49), p. 18).
- \mathcal{S}_1 finite discrete set of different values for variable N_K (p. 19).
- \mathcal{S}_2 finite discrete set of different values for variable N_C (p. 19).
- $\Delta(\cdot, \cdot)$ mAIC model ranking (Definition (50), p. 19).
- $w(\cdot, \cdot)$ mAIC model weights (Definition (51), p. 19).

Prediction and assimilation of additional information.

- $\hat{\pi}(t, j)$ prediction of observation $\pi(t, j)$ (p. 20).
- $\hat{\Gamma}(t, j) = [\hat{\gamma}_1(t, j), \dots, \hat{\gamma}_{N_K}(t, j)] \in [0, 1]^{1 \times N_K}$ prediction of future affiliations (p. 20).
- $P^\Gamma(t, j)$ transition matrix characterizing $\Gamma^*(t, j)$ (p. 20).
- $P_0^\Gamma, \dots, P_{N_E}^\Gamma$ matrices used in linear combination equal to $P^\Gamma(t, j)$ (p. 20).
- $\dot{\gamma}(N_T + 1, j)$ posterior estimate based on the new observation $\pi(N_T + 1, j)$ (p. 21).
- $\dot{y}_k(N_T + 1, j)$ updated affiliation associated with local model θ_k (Definition (56), p. 21).

Numerical investigation.

- $\Gamma^{\text{syn}}(t, j)$ synthetic affiliation process (p. 23).
- $\gamma_k^{\text{syn}}(t, j)$ synthetic affiliation associated with θ_k^{syn} (p. 24).
- dummy1 auxiliary vector of Algorithm 2 containing only ones (p. 24).
- dummy0 auxiliary vector of Algorithm 2 containing only zeros (p. 24).
- $P^{\text{syn}}(t, j, u(t, j))$ synthetic transition matrix (Definition (57), p. 23).
- $P^{k \text{ syn}}(u(t, j))$ synthetic model parameter matrix associated with affiliation $\gamma_k^{\text{syn}}(t, j)$ (p. 23).
- $\sigma^{\text{syn}}(t, j, l)$ synthetic dynamical process (p. 23).
- $\pi^{\text{syn}}(t, j)$ synthetic data (p. 25).
- $P_0^{k \text{ syn}}, \dots, P_{N_E}^{k \text{ syn}}$ synthetic model matrices corresponding to explicit external factors $u(t, j)$ (p. 25).
- $P_{N_E+1}^{k \text{ syn}}, \dots, P_{N_E+N_I}^{k \text{ syn}}$ synthetic model matrices corresponding to implicit external factors $u^{\text{unres}}(t, j)$ (p. 25).
- $\pi_i^{\text{syn}}(t, j)$ i -th vector entry of synthetic data $\pi^{\text{syn}}(t, j)$ (p. 25).
- $\pi^{\text{ANN}(w)}(t, j)$ approximation of $\pi^{\text{syn}}(t, j)$ computed with an ANN based on a network with w neurons (p. 27).
- $\pi^{\text{ANN}}(t, j)$ approximation of $\pi^{\text{syn}}(t, j)$ computed with an ANN (p. 28).

- $\pi^{\text{Markov}}(t, j)$ approximation of $\pi^{\text{syn}}(t, j)$ computed with Markov regression framework (p. 27).
- $\varpi(j, \tau)$ prediction error term dependent on location j and prediction depth τ (p. 29).
- $\varpi_{\text{rel}}(\tau)$ relative mean prediction error (p. 29).
- $\varpi(t, j, \tau)$ prediction error term dependent on time t , location j and prediction depth τ (p. 36).
- $\check{\epsilon}(t, j)$ auxiliary process used in the proof of [Proposition 2.1](#) (p. 39).

Abbreviations.

- SVM support vector machines.
- ANN artificial neural networks.
- AIC Akaike information criterion.
- mAIC modified Akaike information criterion.
- GLM generalized linear models.
- PDEs partial differential equations.
- ODEs ordinary differential equations.
- FEM finite-element method.
- IIA independence of irrelevant alternatives.
- i.i.d. independent and identically distributed.

Acknowledgements

The authors thank the DFG SPP 1276 MetStroem “Meteorology and Turbulence Mechanics”, the Swiss National Science Foundation (project 131845 “AnaGraph”), the Center for Scientific Simulation (Freie Universität Berlin) and the graduate research school GEOSIM (GFZ Potsdam, Universität Potsdam, Freie Universität Berlin) for funding. Further, the authors thank Professor Rupert Klein (Freie Universität Berlin) for stating the question about spatial heterogeneity of nonstationary models that led to a formulation of the problem for discrete processes considered in this manuscript. Special thanks to the unknown referees for their many helpful comments and hints regarding the deployed notation.

References

- [1] H. Akaike, *A new look at the statistical model identification*, IEEE Trans. Automatic Control **19** (1974), no. 6, 716–723. [MR 54 #11691](#) [Zbl 0314.62039](#)
- [2] C. M. Bishop, *Neural networks for pattern recognition*, Clarendon Press, New York, 1995. [MR 97m:68172](#)
- [3] C. Blume, K. Matthes, and I. Horenko, *Supervised learning approaches to classify stratospheric warming events*, J. Atmos. Sci. **69** (2012), no. 6, 1824–1840.
- [4] A. Bordes, S. Ertekin, J. Weston, and L. Bottou, *Fast kernel classifiers with online and active learning*, J. Mach. Learn. Res. **6** (2005), 1579–1619. [MR 2249866](#) [Zbl 1222.68152](#)

- [5] K. P. Burnham and D. R. Anderson, *Model selection and multimodel inference: a practical information-theoretic approach*, 2nd ed., Springer, New York, 2002. MR 1919620 Zbl 1005.62007
- [6] S. Chib and E. Greenberg, *Understanding the Metropolis–Hastings algorithm*, Am. Stat. **49** (1995), no. 4, 327–335.
- [7] J. S. Chipman, *The foundations of utility*, Econometrica **28** (1960), no. 2, 193–224. MR 22 #9284 Zbl 0173.48001
- [8] N. Cristianini and J. Shawe-Taylor, *An introduction to support vector machines and other kernel-based learning methods*, Cambridge University Press, 2000. Zbl 0994.68074
- [9] J. de Wiljes, A. Majda, and I. Horenko, *An adaptive Markov chain Monte Carlo approach to time series clustering of processes with regime transition behavior*, Multiscale Model. Simul. **11** (2013), no. 2, 415–441. MR 3047436
- [10] A. J. Dobson and A. G. Barnett, *An introduction to generalized linear models*, 3rd ed., CRC Press, Boca Raton, 2008. MR 2010a:62003 Zbl 1165.62049
- [11] J. Fox, *Applied regression analysis, linear models, and related methods*, SAGE Publications, Thousand Oaks, 1997.
- [12] J. Gill, *Generalized linear models: a unified approach*, Quantitative Applications in the Social Sciences, no. 134, SAGE Publications, Thousand Oaks, 2001.
- [13] J. Hadamard, *Sur les problèmes aux dérivées partielles et leur signification physique*, Princeton University Bulletin **13** (1902), no. 4, 49–52.
- [14] M. T. Hagan and M. B. Menhaj, *Training feedforward networks with the Marquardt algorithm*, IEEE Trans. Neural Networks **5** (1994), no. 6, 989–993.
- [15] L. D. Haugh and G. E. P. Box, *Identification of dynamic regression (distributed lag) models connecting two time series*, J. Amer. Statist. Assoc. **72** (1977), no. 357, 121–130. MR 56 #4084
- [16] I. Horenko, *Nonstationarity in multifactor models of discrete jump processes, memory, and application to cloud modeling*, J. Atmos. Sci. **68** (2011), no. 7, 1493–1506.
- [17] ———, *On analysis of nonstationary categorical data time series: dynamical dimension reduction, model selection, and applications to computational sociology*, Multiscale Model. Simul. **9** (2011), no. 4, 1700–1726. MR 2012j:60196 Zbl 1244.60070
- [18] K. Hornik, M. Stinchcombe, and H. White, *Multilayer feedforward networks are universal approximators*, Neural Networks **2** (1989), no. 5, 359–366.
- [19] M. A. Katsoulakis, A. J. Majda, and A. Sopsakis, *Multiscale couplings in prototype hybrid deterministic/stochastic systems, I: Deterministic closures*, Commun. Math. Sci. **2** (2004), no. 2, 255–294. MR 2005m:76144 Zbl 1103.93013
- [20] ———, *Multiscale couplings in prototype hybrid deterministic/stochastic systems, II: Stochastic closures*, Commun. Math. Sci. **3** (2005), no. 3, 453–478. MR 2006j:34135 Zbl 1101.34042
- [21] ———, *Hybrid deterministic stochastic systems with microscopic look-ahead dynamics*, Commun. Math. Sci. **8** (2010), no. 2, 409–437. MR 2012c:60116 Zbl 1197.35336
- [22] S. Kirkpatrick, C. D. Gelatt, Jr., and M. P. Vecchi, *Optimization by simulated annealing*, Science **220** (1983), no. 4598, 671–680. MR 85f:90091 Zbl 1225.90162
- [23] V. Kurkova, *Kolmogorov’s theorem and multilayer neural networks*, Neural Networks **5** (1992), no. 3, 501–506.
- [24] J. Lawrence, *Introduction to neural networks: design, theory, and applications*, 6th ed., California Scientific Software Press, Nevada City, CA, 1994.

- [25] W. D. Li and C. A. McMahon, *A simulated annealing-based optimization approach for integrated process planning and scheduling*, Int. J. Comp. Integ. M. **20** (2007), no. 1, 80–95.
- [26] T. F. Liao, *Interpreting probability models: logit, probit, and other generalized linear models*, SAGE Publications, Thousand Oaks, 1994.
- [27] R. D. Luce, *Individual choice behavior: a theoretical analysis*, Wiley, New York, 1959. MR 21 #7127 Zbl 0093.31708
- [28] C. F. Manski and D. McFadden (eds.), *Structural analysis of discrete data with econometric applications*, MIT Press, Cambridge, MA, 1981. MR 83f:62158 Zbl 0504.00023
- [29] D. McFadden, *Conditional logit analysis of qualitative choice behaviour*, Frontiers in econometrics (P. Zarembka, ed.), Academic Press, New York, 1974, pp. 105–142.
- [30] P. Metzner, L. Putzig, and I. Horenko, *Analysis of persistent nonstationary time series and applications*, Commun. Appl. Math. Comput. Sci. **7** (2012), no. 2, 175–229. MR 3005737 Zbl 1275.62067
- [31] A. Y. Ng and M. I. Jordan, *On discriminative vs. generative classifiers: a comparison of logistic regression and naive Bayes*, Adv. Neural Inf. Process. Syst. (2002), 841–848.
- [32] A. Pankratz, *Forecasting with dynamic regression models*, Wiley, Hoboken, NJ, 1991.
- [33] C. P. Robert and G. Casella, *Monte Carlo statistical methods*, 2nd ed., Springer, New York, 2004. MR 2005d:62006 Zbl 1096.62003
- [34] S. Shalev-Shwartz and N. Srebro, *SVM optimization: inverse dependence on training set size*, Proceedings of the 25th International Conference on Machine Learning (A. McCallum and S. Roweis, eds.), 2008, pp. 928–935.
- [35] J. Shawe-Taylor and N. Cristianini, *Kernel methods for pattern analysis*, Cambridge University Press, 2004.
- [36] K. E. Train, *Discrete choice methods with simulation*, 2nd ed., Cambridge University Press, 2009. MR 2010m:91055 Zbl 1269.62073
- [37] S. A. Vavasis, *Quadratic programming is in NP*, Inform. Process. Lett. **36** (1990), no. 2, 73–77. MR 91m:68095 Zbl 0719.90052
- [38] J. von Neumann, *Various techniques used in connection with random digits*, National Bureau of Standards Applied Math Series **11** (1951), 36–38.
- [39] H. Yu and B. M. Wilamowski, *Levenberg–Marquardt training*, Intelligent systems (B. M. Wilamowski and J. D. Irwin, eds.), Industrial Electronics Handbook, no. 5, CRC Press, Boca Raton, 2nd ed., 2011.

Received November 29, 2012. Revised October 22, 2013.

JANA DE WILJES: jana.dewiljes@math.fu-berlin.de

Institute of Mathematics, Freie Universität Berlin, Arnimallee 6, D-14195 Berlin, Germany

LARS PUTZIG: lars.putzig@usi.ch

Institute of Computational Science, Università della Svizzera Italiana, Via Giuseppe Buffi 13, CH-6900 Lugano, Switzerland

ILLIA HORENKO: horenkoi@usi.ch

Institute of Computational Science, Università della Svizzera Italiana, Via Giuseppe Buffi 13, CH-6904 Lugano, Switzerland

LOW MACH NUMBER FLUCTUATING HYDRODYNAMICS OF DIFFUSIVELY MIXING FLUIDS

ALEKSANDAR DONEV, ANDY NONAKA, YIFEI SUN,
THOMAS G. FAI, ALEJANDRO L. GARCIA AND JOHN B. BELL

We formulate low Mach number fluctuating hydrodynamic equations appropriate for modeling diffusive mixing in isothermal mixtures of fluids with different density and transport coefficients. These equations represent a coarse-graining of the microscopic dynamics of the fluid molecules in both space and time and eliminate the fluctuations in pressure associated with the propagation of sound waves by replacing the equation of state with a local thermodynamic constraint. We demonstrate that the low Mach number model preserves the spatiotemporal spectrum of the slower diffusive fluctuations. We develop a strictly conservative finite-volume spatial discretization of the low Mach number fluctuating equations in both two and three dimensions and construct several explicit Runge–Kutta temporal integrators that strictly maintain the equation-of-state constraint. The resulting spatiotemporal discretization is second-order accurate deterministically and maintains fluctuation-dissipation balance in the linearized stochastic equations. We apply our algorithms to model the development of giant concentration fluctuations in the presence of concentration gradients and investigate the validity of common simplifications such as neglecting the spatial nonhomogeneity of density and transport properties. We perform simulations of diffusive mixing of two fluids of different densities in two dimensions and compare the results of low Mach number continuum simulations to hard-disk molecular-dynamics simulations. Excellent agreement is observed between the particle and continuum simulations of giant fluctuations during time-dependent diffusive mixing.

I. Introduction

Stochastic fluctuations are intrinsic to fluid dynamics because fluids are composed of molecules whose positions and velocities are random at thermodynamic scales. Because they span the whole range of scales from the microscopic to the macroscopic [23; 75], fluctuations need to be consistently included in all levels of description. Stochastic effects are important for flows in new microfluidic, nanofluidic, and microelectromechanical devices [7]; novel materials such as nanofluids [79]; and

MSC2010: primary 76T99; secondary 65M08.

Keywords: fluctuating hydrodynamics, low Mach expansion, molecular dynamics, giant fluctuations.

biological systems such as lipid membranes [57], Brownian molecular motors [64], and nanopores [20]; as well as processes where the effect of fluctuations is amplified by strong nonequilibrium effects such as ultraclean combustion, capillary dynamics [16; 68], and hydrodynamic instabilities [55; 14; 44].

One can capture thermal fluctuations using direct particle-level calculations. But even coarse-grained particle methods [59; 22; 23] are computationally expensive because the dynamics of individual particles has time scales significantly shorter than hydrodynamic time scales. Alternatively, thermal fluctuations can be included in the Navier–Stokes equations through stochastic forcing terms as proposed by Landau and Lifshitz [48] and later extended to fluid mixtures [61]. The basic idea of *fluctuating hydrodynamics* is to add a *stochastic flux* corresponding to each dissipative (irreversible, diffusive) flux [62]. This ensures that the microscopic conservation laws and thermodynamic principles are obeyed while also maintaining fluctuation–dissipation balance. Specifically, the equilibrium thermal fluctuations have the Gibbs–Boltzmann distribution dictated by statistical mechanics. Fluctuating hydrodynamics is a useful tool in understanding complex fluid flows far from equilibrium [61], but theoretical calculations are often only feasible after ignoring nonlinearities, inhomogeneities in density, temperature, and transport properties, surface dynamics, gravity, unsteady flow patterns, and other important effects. In the past decade, fluctuating hydrodynamics has been applied to study a number of nontrivial practical problems [31; 68; 71; 3]; however, the numerical methods used are far from the comparable state of the art for deterministic solvers.

Previous computational studies of the effect of thermal fluctuations in fluid mixtures [68; 6; 71] have been based on the compressible fluid equations and thus require small time steps to resolve fast sound waves (pressure fluctuations). Recently, some of us developed finite-volume methods for the incompressible equations of fluctuating hydrodynamics [73], which eliminate the stiffness arising from the separation of scales between the acoustic and vortical modes [47; 52]. For inhomogeneous fluids with nonconstant density, diffusive mass and heat fluxes create local expansion and contraction of the fluid, and the incompressibility constraint should be replaced by a “quasi-incompressibility” constraint [52; 50]. The resulting *low Mach number* equations have been used for some time to model deterministic flows with thermochemical effects [66; 52], and several conservative finite-volume techniques have been developed for solving equations of this type [63; 67; 15; 58; 56]. To our knowledge, thermal fluctuations have not yet been incorporated in low Mach number models.

In this work, we extend the staggered-grid, finite-volume approach developed in [73] to isothermal mixtures of fluids with unequal densities. The imposition of the quasi-incompressibility constraint poses several nontrivial mathematical and computational challenges. At the mathematical level, the traditional low Mach number asymptotic expansions [47; 52] assume spatiotemporal smoothness of the

flow and thus do not directly apply in the stochastic context. At the computational level, enforcing the quasi-incompressibility or equation-of-state (EOS) constraint in a conservative and stable manner requires specialized spatiotemporal discretizations. By careful selection of the analytical form of the EOS constraint and the spatial discretization of the advective fluxes, we are able to maintain strict local conservation and enforce the EOS to within numerical tolerances. In the present work, we employ an explicit projection-based temporal discretization because of the substantial complexity of designing and implementing semi-implicit discretizations of the momentum equation for spatially inhomogeneous fluids [10].

Thermal fluctuations exhibit unusual features in systems out of thermodynamic equilibrium. Notably, external gradients can lead to *enhancement* of thermal fluctuations and to *long-range* correlations between fluctuations [36; 53; 30; 60; 61]. Sharp concentration gradients present during diffusive mixing lead to the development of macroscopic or *giant fluctuations* [77; 72; 9] in concentration, which have been observed using light-scattering and shadowgraphy techniques [76; 12; 75]. These experimental studies have found good but imperfect agreement between the predictions of a simplified fluctuating hydrodynamic theory and experiments. Computer simulations are, in principle, an ideal tool for studying such complex time-dependent processes in the presence of nontrivial boundary conditions without making the sort of approximations necessary for analytical calculations such as assuming spatially constant density and transport coefficients and spatially uniform gradients. On the other hand, the multiscale (more precisely, *many-scale*) nature of the equations of fluctuating hydrodynamics poses many mathematical and computational challenges that are yet to be addressed. Notably, it is necessary to develop temporal integrators that can accurately and robustly handle the large separation of time scales between different physical processes such as mass and momentum diffusion. The computational techniques we develop here form the foundation for incorporating additional physics such as heat transfer and internal energy fluctuations, phase separation and interfacial dynamics, and chemical reactions.

We begin [Section II](#) by formulating the fluctuating low Mach number equations for an isothermal binary fluid mixture. We present both a traditional pressure (constrained) formulation and a gauge (unconstrained) formulation. We analyze the spatiotemporal spectrum of the thermal fluctuations in the linearized equations and demonstrate that the low Mach equations eliminate the fast (sonic) pressure fluctuations but maintain the correct spectrum of the slow (diffusive) fluctuations. In [Section III](#), we develop projected Runge–Kutta schemes for solving the spatially discretized equations, including a midpoint and a trapezoidal second-order predictor-corrector scheme and a third-order three-stage scheme. In [Section IV](#), we describe a spatial discretization of the equations that strictly maintains the equation-of-state constraint and also obeys a fluctuation-dissipation balance principle [29]. In

[Section V](#), we study the steady-state spectrum of giant concentration fluctuations in the presence of an applied concentration gradient in a mixture of two dissimilar fluids and test the applicability of common approximations that neglect spatial inhomogeneities. In [Section VI](#), we study the dynamical evolution of giant interface fluctuations during diffusive mixing of two dissimilar fluids, using both hard-disk molecular dynamics and low Mach number fluctuating hydrodynamics. We find excellent agreement between the two, providing a strong support for the usefulness of the fluctuating low Mach number equations as a coarse-grained model of complex fluid mixtures. In [Section VII](#), we offer some concluding remarks and point out several outstanding challenges for the future. Several technical calculations and procedures are detailed in the appendices.

II. Low Mach number equations

The compressible equations of fluctuating hydrodynamics were proposed some time ago [48] and have since been studied and applied successfully to a variety of situations [61]. The presence of rapid pressure fluctuations due to the propagation of sound waves leads to stiffness that makes it computationally expensive to solve the fully compressible equations numerically especially for typical liquids. It is therefore important to develop fluctuating hydrodynamics equations that capture the essential physics in cases where acoustics can be neglected.

It is important to note that the equations of fluctuating hydrodynamics are to be interpreted as a mesoscopic coarse-grained representation of the mass, momentum, and energy transport that occurs at microscopic scales through molecular interactions (collisions). As such, these equations implicitly contain a mesoscopic coarse-graining length and time scale that is larger than molecular scales [34]. While a coarse-graining scale does not appear explicitly in the formal stochastic partial differential equations (SPDEs) written in this section (but note that it can be if desired [26]), it does explicitly enter in the spatiotemporal discretization described in [Section IV](#) through the grid spacing (equivalently, the volume of the grid or, more precisely, the number of molecules per grid cell) and time-step size. This changes the appropriate interpretation of convergence of numerical methods to a continuum limit in the presence of fluctuations and nonlinearities [18]. Only for the linearized equations of fluctuating hydrodynamics [61] can the formal SPDEs be given a precise continuum meaning [29].

Developing coarse-grained models that only resolve the relevant spatiotemporal scales is a well-studied but still ad hoc procedure that requires substantial a priori physical insight [62]. More precise mathematical mode-elimination procedures [39; 40; 41; 45] are technically involved and often purely formal especially in the context of SPDEs [26]. Here we follow a heuristic approach to constructing

fluctuating low Mach number equations, starting from the well-known deterministic low Mach equations (which can be obtained via asymptotic analysis [47; 52]) and then adding fluctuations in a manner consistent with fluctuation-dissipation balance. Alternatively, our low Mach number equations can be seen as a formal asymptotic limit in which the noise terms are formally treated as smooth forcing terms; a more rigorous derivation is nontrivial and is deferred for future work.

II-A. Compressible equations. The starting point of our investigations is the system of isothermal compressible equations of fluctuating hydrodynamics for the density $\rho(\mathbf{r}, t)$, velocity $\mathbf{v}(\mathbf{r}, t)$, and mass concentration $c(\mathbf{r}, t)$ for a mixture of two fluids in d dimensions. In terms of mass and momentum densities, the equations can be written as conservation laws [62; 61; 6]

$$\begin{aligned} \partial_t \rho + \nabla \cdot (\rho \mathbf{v}) &= 0, \\ \partial_t (\rho \mathbf{v}) + \nabla \cdot (\rho \mathbf{v} \mathbf{v}^T) &= -\nabla P + \rho \mathbf{g} \\ &\quad + \nabla \cdot \left[\eta (\nabla \mathbf{v} + \nabla^T \mathbf{v}) + \left(\kappa - \frac{2}{d} \eta \right) (\nabla \cdot \mathbf{v}) \mathbf{I} + \boldsymbol{\Sigma} \right], \\ \partial_t (\rho_1) + \nabla \cdot (\rho_1 \mathbf{v}) &= \nabla \cdot [\rho \chi (\nabla c + K_P \nabla P) + \boldsymbol{\Psi}], \end{aligned} \quad (1)$$

where $\rho_1 = \rho c$ is the density of the first component, $\rho_2 = (1 - c)\rho$ is the density of the second component, $P(\rho, c; T)$ is the equation of state for the pressure at the reference temperature $T = T_0 = \text{const}$, and \mathbf{g} is the gravitational acceleration. Temperature fluctuations are neglected in this study but can be accounted for using a similar approach. The shear viscosity η , bulk viscosity κ , mass diffusion coefficient χ , and barodiffusion coefficient K_P , in general, depend on the state. The barodiffusion coefficient K_P above (denoted by k_P/P in [6]; see Equation (A.17) there) is not a transport coefficient but rather determined from thermodynamics [49] as

$$K_P = \frac{(\partial \mu / \partial P)_c}{(\partial \mu / \partial c)_P} = -\rho^{-2} \frac{(\partial \rho / \partial c)_P}{(\partial \mu / \partial c)_P} = \frac{(\partial P / \partial c)_\rho}{\rho^2 c_T^2 \mu_c}, \quad (2)$$

where μ is the chemical potential of the mixture at the reference temperature, $\mu_c = (\partial \mu / \partial c)_P$, and $c_T^2 = (\partial P / \partial \rho)_c$ is the isothermal speed of sound. The capital Greek letters denote stochastic momentum and mass fluxes that are formally modeled as [73]

$$\begin{aligned} \boldsymbol{\Sigma} &= \sqrt{\eta k_B T} \left(\boldsymbol{\mathcal{W}} + \boldsymbol{\mathcal{W}}^T - \frac{2}{d} \text{Tr} \boldsymbol{\mathcal{W}} \right) + \sqrt{\frac{2\kappa k_B T}{d}} \text{Tr} \boldsymbol{\mathcal{W}}, \\ \boldsymbol{\Psi} &= \sqrt{2\chi \rho \mu_c^{-1} k_B T} \tilde{\boldsymbol{\mathcal{W}}}, \end{aligned} \quad (3)$$

where k_B is Boltzmann's constant and $\mathcal{W}(\mathbf{r}, t)$ and $\tilde{\mathcal{W}}(\mathbf{r}, t)$ are standard zero-mean, unit-variance random Gaussian tensor and vector fields with uncorrelated components

$$\langle \mathcal{W}_{ij}(\mathbf{r}, t) \mathcal{W}_{kl}(\mathbf{r}', t') \rangle = \delta_{ik} \delta_{jl} \delta(t - t') \delta(\mathbf{r} - \mathbf{r}')$$

and similarly for $\tilde{\mathcal{W}}$.

II-B. Low Mach equations. At mesoscopic scales, in typical liquids, sound waves are much faster than momentum diffusion and can usually be eliminated from the fluid-dynamics description. Formally, this corresponds to taking the zero-Mach-number singular limit $c_T \rightarrow \infty$ of the system (1) by performing an asymptotic analysis as the Mach number $\text{Ma} = U/c_T \rightarrow 0$, where U is a reference flow velocity. The limiting dynamics can be obtained by performing an asymptotic expansion in the Mach number [47]. In a deterministic setting, this analysis shows that the pressure can be written in the form

$$P(\mathbf{r}, t) = P_0(t) + \pi(\mathbf{r}, t),$$

where $\pi = O(\text{Ma}^2)$. The low Mach number equations can then be obtained by making the ansatz that the thermodynamic behavior of the system is captured by the reference pressure, P_0 , and π captures the mechanical behavior while not affecting the thermodynamics. We note that, when the system is sufficiently large or the gravitational forcing is sufficiently strong, assuming a spatial constant reference pressure is not valid. In those cases, the reference pressure represents a global hydrostatic balance $\nabla P_0 = \rho_0 \mathbf{g}$ (see [32] for details of the construction of these types of models). Here, however, we will restrict consideration to cases where gravity causes negligible changes in the thermodynamic state across the domain.

In this case, the reference pressure constrains the system so that the evolution of ρ and c remains consistent with the thermodynamic equation of state

$$P(\rho(\mathbf{r}, t), c(\mathbf{r}, t); T) = P_0(t). \quad (4)$$

This constraint means that any change in concentration (equivalently, ρ_1) must be accompanied by a corresponding change in density as would be observed in a system at thermodynamic equilibrium held at the fixed reference pressure and temperature. This implies that variations in density are coupled to variations in composition. Note that we do not account for temperature variations in our isothermal model.

The equation for ρ_1 can be written in primitive (nonconservation) form as the concentration equation

$$\rho \frac{Dc}{Dt} = \rho D_t c = \rho (\partial_t c + \mathbf{v} \cdot \nabla c) = \nabla \cdot \mathbf{F}, \quad (5)$$

where the nonadvective (diffusive and stochastic) fluxes are denoted with

$$\mathbf{F} = \rho\chi\nabla c + \Psi.$$

Note that there is no barodiffusion flux because barodiffusion is of thermodynamic origin (as seen from (2) [61]) and involves the gradient of the *thermodynamic* pressure $\nabla P_0 = 0$. By differentiating the EOS constraint along a Lagrangian trajectory, we obtain

$$\frac{D\rho}{Dt} = \beta\rho\frac{Dc}{Dt} = \beta\nabla\cdot\mathbf{F} = \partial_t\rho + \mathbf{v}\cdot\nabla\rho = -\rho\nabla\cdot\mathbf{v}, \quad (6)$$

where the solutal expansion coefficient

$$\beta(c) = \frac{1}{\rho}\left(\frac{\partial\rho}{\partial c}\right)_{P_0}$$

is determined by the specific form of the EOS.

Equation (6) shows that the EOS constraint can be rewritten as a constraint on the divergence of velocity,

$$\rho\nabla\cdot\mathbf{v} = -\beta\nabla\cdot\mathbf{F}. \quad (7)$$

Note that the usual incompressibility constraint is obtained when the density is not affected by changes in concentration, $\beta = 0$. When $\beta \neq 0$, changes in composition (concentration) due to diffusion cause local expansion and contraction of the fluid and thus a nonzero $\nabla\cdot\mathbf{v}$. It is important at this point to consider the boundary conditions. For a closed system, such as a periodic domain or a system with rigid boundaries, we must ensure that the integral of $\nabla\cdot\mathbf{v}$ over the domain is zero. This is consistent with (7) if β/ρ is constant so that we can rewrite (7) in the form $\nabla\cdot\mathbf{v} = -\nabla\cdot((\beta/\rho)\mathbf{F})$. In this case, P_0 does not vary in time. If β/ρ is not constant, then for a closed system the reference pressure P_0 must vary in time to enforce that the total fluid volume remains constant. Here we will assume that $\beta/\rho = \text{const}$, and we will give a specific example of an EOS that obeys this condition.

The asymptotic low Mach analysis of (1) is standard and follows the procedure outlined in [47], formally treating the stochastic forcing as smooth. This analysis leads to the *isothermal low Mach number* equations for a binary mixture of fluids in conservation form,

$$\partial_t(\rho\mathbf{v}) + \nabla\pi = -\nabla\cdot(\rho\mathbf{v}\mathbf{v}^T) + \nabla\cdot[\eta(\nabla\mathbf{v} + \nabla^T\mathbf{v}) + \Sigma] + \rho\mathbf{g} \equiv \mathbf{f}(\rho, \mathbf{v}, c, t), \quad (8)$$

$$\partial_t(\rho_1) = -\nabla\cdot(\rho_1\mathbf{v}) + \nabla\cdot\mathbf{F} \equiv h(\rho, \mathbf{v}, c, t), \quad (9)$$

$$\partial_t(\rho_2) = -\nabla\cdot(\rho_2\mathbf{v}) - \nabla\cdot\mathbf{F}, \quad (10)$$

$$\text{such that } \nabla\cdot\mathbf{v} = -(\rho^{-1}\beta)\nabla\cdot\mathbf{F} \equiv S(\rho, c, t). \quad (11)$$

The gradient of the nonthermodynamic component of the pressure π (Lagrange multiplier) appears in the momentum equation as a driving force that ensures the EOS constraint (11) is obeyed. We note that the bulk viscosity term gives a gradient term that can be absorbed in π and therefore does not explicitly need to appear in the equations. By adding the two density equations (9) and (10), we get the usual continuity equation for the total density,

$$\partial_t \rho = -\nabla \cdot (\rho \mathbf{v}). \quad (12)$$

Our conservative numerical scheme is based on (8), (9), (11), and (12).

In Appendix A, we apply the standard linearized fluctuating hydrodynamics analysis to the low Mach number equations. This gives expressions for the equilibrium and nonequilibrium static and dynamic covariances (spectra) of the fluctuations in density and concentration as a function of wavenumber and wave frequency. Specifically, the dynamic structure factor in the low Mach number approximation has the form

$$S_{\rho, \rho}(\mathbf{k}, \omega) = \langle (\widehat{\delta\rho})(\widehat{\delta\rho})^* \rangle = \beta^2 (\rho \mu_c^{-1} k_B T) \frac{2\chi k^2}{\omega^2 + \chi^2 k^4}.$$

The linearized analysis shows that the low Mach number equations reproduce the slow fluctuations (small ω) in density and concentration (central Rayleigh peak in the dynamic structure factor [61; 29]) as in the full compressible equations (see Section A.1) while eliminating the fast isentropic pressure fluctuations (side Brillouin peaks) from the dynamics.

The fluctuations in velocity, however, are different between the compressible and low Mach number equations. In the compressible equations, the dynamic structure factor for the longitudinal component of velocity decays to zero as $\omega \rightarrow \infty$ because it has two sound (Brillouin) peaks centered around $\omega \approx \pm c_T k$ in addition to the central diffusive (Rayleigh) peak. The low Mach number equations reproduce the central peak (slow fluctuations) correctly, replacing the side peaks with a flat spectrum for large ω , which is unphysical as it formally makes the velocity white in time. The low Mach equations should therefore be used only for time scales larger than the sound propagation time.

The fact that the velocity fluctuations are white in space and in time poses a further challenge in interpreting the nonlinear low Mach number equations, and in particular, numerical schemes may not converge to a sensible limit as the time step goes to zero. In practice, just as the spatial discretization of the equations imposes a spatial smoothing or regularization of the fluctuations, the temporal discretization of the equations imposes a temporal smoothing and filters the problematic large frequencies. In the types of problems we study in this work, the problem concentration fluctuations can be neglected, $\widehat{\Psi} \approx \mathbf{0}$, because the

concentration fluctuations are dominated by nonequilibrium effects. If $\widehat{\Psi} = \mathbf{0}$, the problematic white-in-time longitudinal component of velocity disappears.

Model equation of state. In general, the EOS constraint (4) is a nonlinear constraint. In this work, we consider a specific linear EOS,

$$\frac{\rho_1}{\bar{\rho}_1} + \frac{\rho_2}{\bar{\rho}_2} = \frac{c\rho}{\bar{\rho}_1} + \frac{(1-c)\rho}{\bar{\rho}_2} = 1, \quad (13)$$

where $\bar{\rho}_1$ and $\bar{\rho}_2$ are the densities of the pure component fluids ($c = 1$ and $c = 0$, respectively), giving

$$\beta = \rho \left(\frac{1}{\bar{\rho}_2} - \frac{1}{\bar{\rho}_1} \right) = \frac{\bar{\rho}_1 - \bar{\rho}_2}{c\bar{\rho}_2 + (1-c)\bar{\rho}_1}. \quad (14)$$

It is important that for this specific form of the EOS β/ρ is a material constant independent of the concentration. The density dependence (14) on concentration arises if one assumes that the two fluids do not change volume upon mixing. This is a reasonable assumption for liquids that are not too dissimilar at the molecular level. Surprisingly, the EOS (13) is also valid for a mixture of ideal gases since

$$P = P_1 + P_2 = P_0 = nk_B T = (n_1 + n_2)k_B T = \left(\frac{\rho_1}{m_1} + \frac{\rho_2}{m_2} \right) k_B T,$$

where m is molecular mass and $n = \rho/m$ is the number density. This is exactly of the form (13) with $\bar{\rho}_1 = m_1 P_0 / (k_B T) = nm_1$ and $\bar{\rho}_2 = nm_2$.

Even if the specific EOS (13) is not a very good approximation over the entire range of concentration $0 \leq c \leq 1$, (13) may be a very good approximation over the range of concentrations of interest if $\bar{\rho}_1$ and $\bar{\rho}_2$ are adjusted accordingly. In this case, $\bar{\rho}_1$ and $\bar{\rho}_2$ are not the densities of the pure component fluids but rather fitting parameters that approximate the true EOS in the range of concentrations of interest. For small variations in concentration around some reference concentration \bar{c} and density $\bar{\rho}$, one can approximate $\beta \approx \bar{\rho}^{-1} (\partial \rho / \partial c)_{\bar{c}}$ by a constant and determine appropriate values of $\bar{\rho}_1$ and $\bar{\rho}_2$ from (14) and the EOS (13) evaluated at the reference state. Our specific form choice of the EOS will aid significantly in the construction of simple conservative spatial discretizations that strictly maintain the EOS without requiring complicated nonlinear iterative corrections.

Boundary conditions. Several different types of boundary conditions can be imposed for the low Mach number equations just as for the more familiar incompressible equations. The simplest case is when periodic boundary conditions are used for all of the variables. We briefly describe the different types of conditions that can be imposed at a physical boundary with normal direction n .

For the concentration (equivalently, ρ_1), either Neumann (zero mass flux) or Dirichlet (fixed concentration) boundary conditions can be imposed. Physically,

a Neumann condition corresponds to a physical boundary that is impermeable to mass while Dirichlet conditions correspond to a permeable membrane that connects the system to a large reservoir held at a specified concentration. In the case of Neumann conditions for concentration, both the normal component of the diffusive flux $F_n = 0$ and the advective flux $\rho_1 v_n = 0$ vanish at the boundary, implying that the normal component of velocity must vanish, $v_n = 0$. For Dirichlet conditions on the concentration, however, there will, in general, be a nonzero normal diffusive flux F_n through the boundary. This diffusive flux for concentration will induce a corresponding mass flux as required to maintain the equation of state near the boundary. From the condition (11), we infer the proper boundary condition for the normal component of velocity to be

$$v_n = -(\rho^{-1}\beta)F_n. \quad (15)$$

This condition expresses the notion that there is no net volume change for the fluid in the domain. Note that no additional boundary conditions can be specified for ρ since its boundary conditions follow from those on c via the EOS constraint.

For the tangential component of velocity \mathbf{v}_τ , we either impose a no-slip condition $\mathbf{v}_\tau = 0$ or a free-slip boundary condition in which the tangential component of the normal viscous stress vanishes,

$$\eta \left(\frac{\partial v_n}{\partial \boldsymbol{\tau}} + \frac{\partial \mathbf{v}_\tau}{\partial n} \right) = \mathbf{0}.$$

In the case of zero normal mass flux, $v_n = 0$, the free-slip condition simplifies to a Neumann condition for the tangential velocity, $\partial \mathbf{v}_\tau / \partial n = 0$.

II-C. Gauge formalism. The low Mach number system of equations (8), (9), (11), and (12) is a *constrained* problem. For the purposes of analysis and in particular for constructing higher-order temporal integrators, it is useful to rewrite the constrained low Mach number equations as an *unconstrained* initial-value problem. In the incompressible case, $\nabla \cdot \mathbf{v} = 0$, we can write the constrained Navier–Stokes equations as an unconstrained system by eliminating the pressure using a projection-operator formalism. The constraint $\nabla \cdot \mathbf{v} = 0$ is a constant linear constraint and independent of the state and of time. However, in the low Mach number equations, the velocity-divergence constraint $\nabla \cdot \mathbf{v} = -\beta D_t c$ depends on concentration and also on time when there are additional (stochastic or deterministic) forcing terms in the concentration equation. Treating this type of system requires a more general vector-field decomposition. This more general vector-field decomposition provides the basis for a projection-based discretization of the constrained system. We also introduce a gauge formulation of the system [33] that casts the evolution as a nonlocal unconstrained system that is analytically equivalent to the original

constrained evolution. The gauge formulation allows us to develop higher-order method-of-lines temporal integration algorithms.

Vector-field decomposition. The velocity in the low Mach number equations can be split into two components,

$$\mathbf{v} = \mathbf{u} + \nabla\zeta,$$

where $\nabla \cdot \mathbf{u} = 0$ is a divergence-free (solenoidal or vortical) component, and therefore,

$$\nabla \cdot \mathbf{v} = \nabla^2\zeta = S(\rho, c, t).$$

This is a Poisson problem for ζ that is well-posed for appropriate boundary conditions on \mathbf{v} . Specifically, periodic boundary conditions on \mathbf{v} imply periodic boundary conditions for \mathbf{u} and ζ . At physical boundaries where a Dirichlet condition (15) is specified for the normal component of the velocity, we set $u_n = 0$ and use Neumann conditions for the Poisson solve, $\partial\zeta/\partial n = v_n$.

We can now define a more general vector-field decomposition that plays the role of the Hodge decomposition in incompressible flow. Given a vector field $\tilde{\mathbf{v}}$ and a density ρ , we can decompose $\tilde{\mathbf{v}}$ into three components

$$\tilde{\mathbf{v}} = \mathbf{u} + \nabla\zeta + \rho^{-1}\nabla\psi.$$

This decomposition can be obtained by using the condition $\nabla \cdot \mathbf{u} = 0$ and $\nabla^2\zeta = S$, which allows us to define a density-weighted Poisson equation for ψ ,

$$\nabla \cdot (\rho^{-1}\nabla\psi) = -\nabla \cdot (\tilde{\mathbf{v}} - \nabla\zeta) = -\nabla \cdot \tilde{\mathbf{v}} + S(\rho, c, t).$$

Let L_ρ^{-1} denote the solution operator to the density-dependent Poisson problem, formally,

$$L_\rho^{-1} = [\nabla \cdot (\rho^{-1}\nabla)]^{-1},$$

and also define a density-dependent projection operator \mathcal{P}_ρ defined through its action on a vector field \mathbf{w} ,

$$\mathcal{P}_\rho \mathbf{w} = \mathbf{w} - \rho^{-1}\nabla[L_\rho^{-1}(\nabla \cdot \mathbf{w})].$$

This is a well-known variable-density generalization [2] of the constant-density projection operator $\mathcal{P}\mathbf{w} = \mathbf{w} - \nabla[\nabla^{-2}(\nabla \cdot \mathbf{w})]$. We can now write

$$\mathbf{u} = \mathcal{P}_\rho(\tilde{\mathbf{v}} - \nabla\zeta) = \mathcal{P}_\rho\tilde{\mathbf{v}} + \rho^{-1}\nabla[L_\rho^{-1}S(\rho, c, t)] - \nabla\zeta.$$

This gives

$$\mathbf{v} = \mathbf{u} + \nabla\zeta = \mathcal{R}_S(\tilde{\mathbf{v}}),$$

where we have introduced an affine transformation $\mathcal{R}_S(\rho, c, t)$ that depends on ρ , c , and t through $S(\rho, c, t)$ and is defined via its action on a vector field \mathbf{w} ,

$$\mathcal{R}_S(\mathbf{w}) = \mathbf{w} - \rho^{-1} \nabla [L_\rho^{-1}(\nabla \cdot \mathbf{w} - S)]. \quad (16)$$

Note that application of \mathcal{R}_S requires only one Poisson solve and does not actually require computing ζ .

Gauge formulation. The low Mach number system (8), (9), (11), and (12) has the form

$$\begin{aligned} \partial_t \rho &= -\nabla \cdot (\rho \mathbf{v}), \\ \partial_t \mathbf{m} + \nabla \pi &= \mathbf{f}(c, \mathbf{v}, t), \\ \partial_t \rho_1 &= h(c, \mathbf{v}, t), \\ \nabla \cdot \mathbf{v} &= S(\rho, c, t), \end{aligned} \quad (17)$$

where $\mathbf{m} = \rho \mathbf{v}$ is the momentum density and \mathbf{f} , h , and S are as defined in (8), (9), and (11). At present, we will assume that these functions are smooth functions of time, which is only justified in the presence of stochastic forcing terms in a linearized setting. We note that, for the constrained system, ρ is not an independent variable because of the EOS constraint (13); however, we will retain the evolution of ρ with the implicit understanding that the evolution must be constrained so that ρ and c remain consistent with (13).

To define the gauge formulation, we introduce a new variable

$$\tilde{\mathbf{m}} = \rho \tilde{\mathbf{v}} = \mathbf{m} + \nabla \psi,$$

where ψ is a *gauge* variable. We note that ψ is not uniquely determined; however, the specific choice does not matter. If we choose the gauge so that $\partial_t \psi = \pi$, then the momentum equation in (17) is equivalent to

$$\partial_t \tilde{\mathbf{m}} = \mathbf{f}(\rho, \mathbf{v}, c, t).$$

The appropriate boundary conditions for ψ are linked to the boundary conditions on \mathbf{v} ; we set ψ to be periodic if \mathbf{v} is periodic and employ a homogeneous Neumann (natural) boundary condition $\partial \psi / \partial n = 0$ if a Dirichlet condition (15) is specified for the normal component of the velocity v_n . Note that, in the spatially discrete staggered formulation that we employ, the homogeneous Neumann condition follows automatically from the boundary conditions on velocity used to define the appropriate divergence and gradient operators in the interior of the domain.

If we know $\tilde{\mathbf{m}}$ and ρ , we can then define $\tilde{\mathbf{v}} = \tilde{\mathbf{m}} / \rho$ and compute $\mathbf{v} = \mathcal{R}_S(\tilde{\mathbf{v}})$, where \mathcal{R}_S is defined in (16). Thus, by using the gauge formulation, we can formally

write the low Mach number equations in the form of an unconstrained initial value problem

$$\partial_t \tilde{\mathbf{m}} = \mathbf{f}(\rho(c), \mathcal{R}_S(\tilde{\mathbf{v}}), c, t), \quad (18)$$

$$\partial_t \rho_1 = h(\rho(c), \mathcal{R}_S(\tilde{\mathbf{v}}), c, t). \quad (19)$$

The utility of the gauge formulation is that, in fact, we do not need to know ψ in order to determine \mathbf{v} . Therefore, the time-evolution equation for ψ does not actually need to be solved, and in particular, π does not need to be computed. Furthermore, by adopting the gauge formulation, we can directly use a method-of-lines approach for spatially discretizing the system (18)–(19) and then apply standard Runge–Kutta temporal integrators to the resulting system of ordinary (stochastic) differential equations.

It is important to emphasize that the actual independent physical variables in the low Mach formulation (18)–(19) are the vortical (solenoidal) component of velocity \mathbf{u} and the concentration c . The density $\rho = \rho(c)$ and the velocity $\mathbf{v} = \mathbf{u} + \nabla[\nabla^{-2}S(\rho, c, t)]$ are determined from \mathbf{u} and c and the constraints; hence, they can formally be eliminated from the system as can be seen in the linearized analysis in [Appendix A](#), which shows that fluctuations in the vortical velocity modes are decoupled from the longitudinal fluctuations.

III. Temporal integration

Our spatiotemporal discretization follows a “method-of-lines” approach in which we first discretize the equations (8), (9), (11), and (12) in space and then integrate the resulting semicontinuum equations in time. Our uniform staggered-grid spatial discretization of the low Mach number equations is relatively standard and is described in [Section IV](#). The main difficulty is the temporal integration of the resulting equations in the presence of the EOS constraint. Our temporal integrators are based on the gauge formulation (18)–(19) of the low Mach equations. The gauge formulation is unconstrained and enables us to use standard temporal integrators for initial-value problems. In the majority of this section, we assume that all of the fields and differential operators have already been spatially discretized and focus on the temporal integration of the resulting initial-value problem.

Because in the present schemes we handle both diffusive and advective fluxes explicitly, the time-step size Δt is restricted by well-known CFL conditions. For fluctuating hydrodynamics applications, the time step is typically limited by momentum diffusion,

$$\alpha_v = \frac{v\Delta t}{\Delta x^2} < \frac{1}{2d},$$

where d is the number of spatial dimensions and Δx is the grid spacing. The design and implementation of numerical methods that handle momentum diffusion semi-implicitly, as done in [73] for incompressible flow, is substantially more difficult for the low Mach number equations because it requires a variable-coefficient implicit fluid solver. We have recently developed an efficient Stokes solver for solving variable-density and variable-viscosity time-dependent and steady Stokes problems [10], and in future work, we will employ this solver to construct a semi-implicit temporal integrator for the low Mach number equations.

Our temporal discretization will make use of the special form of the EOS and the discretization of mass advection described in Section IV-C in order to strictly maintain the EOS relation (13) between density and concentration in each cell at *all* intermediate values. Therefore, no additional action is needed to enforce the EOS constraint after an update of ρ_1 and ρ . This is, however, only true to within the accuracy of the Poisson solver and also roundoff, and it is possible for a slow drifting off the EOS to occur over many time steps. In Section III-C, we describe a correction that prevents such drifting and ensures that the EOS is obeyed at all times to essentially roundoff tolerance. For simplicity, we will often omit the explicit update for the density ρ and instead focus on updating ρ_1 and the momentum density $\mathbf{m} = \rho \mathbf{v}$ with the understanding that ρ is updated whenever ρ_1 is.

III-A. Euler scheme. The foundation for our higher-order explicit temporal integrators is the first-order Euler method applied to the gauge formulation (18)–(19).

Gauge-free Euler update. We use a superscript to denote the time step and the point in time where a given term is evaluated, e.g., $f^n \equiv f_D(\rho^n, \mathbf{v}^n, c^n, t^n)$, where f_D denotes the spatial discretization of f with analogous definitions for h^n and S^n . We also denote the time-step size with $\Delta t = t^{n+1} - t^n$. Assume that at the beginning of time step n we know $\tilde{\mathbf{m}}^n$ and we can then compute

$$\mathbf{v}^n = \mathcal{R}_S^n(\tilde{\mathbf{v}}^n)$$

by enforcing the constraint (17). Here \mathcal{R}_S^n denotes the affine transformation (16) with all terms evaluated at the beginning of the time step so that $\nabla \cdot \mathbf{v}^n = S^n$. An Euler step for the low Mach equations then consists of the update

$$\begin{aligned} \rho_1^{n+1} &= \rho_1^n + \Delta t h^n, \\ \tilde{\mathbf{m}}^{n+1} &= \tilde{\mathbf{m}}^n + \Delta t \mathbf{f}^n \end{aligned} \quad (20)$$

together with an update of the density ρ^{n+1} consistent with ρ_1^{n+1} .

At the beginning of the next time step, \mathbf{v}^{n+1} will be calculated from $\tilde{\mathbf{m}}^{n+1}$ by applying \mathcal{R}_S^{n+1} , and it is only \mathbf{v}^{n+1} that will actually be used during time step $n+1$. We therefore do not need to explicitly store $\tilde{\mathbf{m}}^{n+1}$ and can instead replace it with

$\mathbf{m}^{n+1} = \rho^{n+1} \mathbf{v}^{n+1}$ without changing any of the observable results. This is related to the fact that the gauge is de facto arbitrary and, in the present setting, the gauge formulation is simply a formalism to put the equations in an unconstrained form suitable for method of lines discretization. The difference between $\tilde{\mathbf{m}}$ and \mathbf{m} is a (discrete) gradient of a scalar. Since our temporal integrators only use linear combinations of the intermediate values, the difference between the final result for $\tilde{\mathbf{m}}^{n+1}$ and \mathbf{m}^n is also a gradient of a scalar and replacing $\tilde{\mathbf{m}}^{n+1}$ with \mathbf{m}^{n+1} simply amounts to redefining the (arbitrary) gauge variable. For these reasons, the Euler advance,

$$\begin{aligned} \rho_1^{n+1} &= \rho_1^n + \Delta t h^n, \\ \mathbf{m}^{n+1} &= \rho^{n+1} \mathcal{R}_S^{n+1} [(\rho^{n+1})^{-1} (\mathbf{m}^n + \Delta t \mathbf{f}^n)], \end{aligned} \quad (21)$$

is analytically equivalent to (20). We will use this form as the foundation for our temporal integrators. The equivalence to the gauge form implies that the update specified by (21) can be viewed as an explicit update in spite of the formal dependence of the update on the solution at both old and new time levels.

Stochastic forcing. Thermal fluctuations cannot be straightforwardly incorporated in (21) because it is not clear how to define \mathcal{R}_S^{n+1} . In the deterministic setting, S is a function of concentration and density and can be evaluated pointwise at time level $n+1$. When the white-in-time stochastic concentration flux Ψ is included, however, S cannot be evaluated at a particular point of time. Instead, one must think of Ψ as representing the *average* stochastic flux over a given time interval δt , which can be expressed in terms of the increments $\sqrt{\delta t} \tilde{\mathbf{W}}$ of the underlying Wiener processes,

$$\Psi(\delta t, \tilde{\mathbf{W}}) = \sqrt{\frac{2\chi\rho\mu_c^{-1}k_B T}{\delta t \Delta V}} \tilde{\mathbf{W}},$$

where $\tilde{\mathbf{W}}$ is a collection of normal variates generated using a pseudorandom number generator and ΔV is the volume of the hydrodynamic cells. Similarly, the average stochastic momentum flux over a time step is modeled as

$$\Sigma(\delta t, \mathbf{W}) = \sqrt{\frac{\eta k_B T}{\delta t \Delta V}} (\mathbf{W} + \mathbf{W}^T),$$

where \mathbf{W} are normal random variates. As described in more detail in [73], stochastic fluxes are spatially discretized by generating normal variates on the faces of the grid on which the corresponding variable is discretized independently at each time step. As mentioned earlier, the volume of the grid cell appears here because it expresses the spatial coarse-graining length scale (i.e., the degree of coarse-graining for which a fluid element with discrete molecules can be modeled by continuous density fields) implicit in the equations of fluctuating hydrodynamics. Similarly,

the time interval $\delta t \sim \Delta t$ expresses the typical time scale at which the mass and momentum transfer can be modeled with low Mach number hydrodynamics.

With this in mind, we first evaluate the velocity divergence associated with the constraint using the particular sample of Ψ ,

$$S = -(\rho^{-1}\beta)\nabla \cdot [\rho\chi\nabla c + \Psi(\delta t, \tilde{W})].$$

We then define a discrete affine operator $\mathcal{R}_F(\delta t, \tilde{W})$ in terms of its action on the momentum m

$$[\mathcal{R}_F(\delta t, \tilde{W})](m) = \rho\mathcal{R}_S(\rho^{-1}m).$$

Using this shorthand notation, the momentum update in (21) in the presence of thermal fluctuations can be written as

$$m^{n+1} = [\mathcal{R}_F^{n+1}(\Delta t, \tilde{W}^{n+1})](m^n + \Delta t f^n).$$

Observe that this is a conservative momentum update since the application of \mathcal{R}_F subtracts the (discrete) gradient of a scalar from the momentum. In actual implementation, it is preferable to apply \mathcal{R}_F^{n+1} at the beginning of the time step $n+1$, instead of at the end of time step n , once the value S^{n+1} is computed from the diffusive and stochastic fluxes for the concentration.

Euler–Maruyama update. Following the above discussion, we can write an Euler–Maruyama temporal integrator for the low Mach number equations in the shorthand notation,

$$\begin{aligned} m^n &= [\mathcal{R}_F^n(\Delta t, \tilde{W}^n)](\tilde{m}^n), \\ \rho_1^{n+1} &= \rho_1^n + \Delta t \bar{h}^n + \check{h}^n(\Delta t, \tilde{W}^n), \\ \tilde{m}^{n+1} &= m^n + \Delta t \bar{f}^n + \check{f}^n(\Delta t, \mathbf{W}^n), \end{aligned} \tag{22}$$

where \mathbf{W}^n and \tilde{W}^n are collections of standard normal variates generated using a pseudorandom number generator independently at each time step. Here the deterministic increments are written using the shorthand notation

$$\begin{aligned} \bar{f} &= \nabla \cdot [-\rho\mathbf{v}\mathbf{v}^T + \eta(\nabla\mathbf{v} + \nabla^T\mathbf{v})] + \rho\mathbf{g}, \\ \bar{h} &= \nabla \cdot (-\rho_1\mathbf{v} + \rho\chi\nabla c). \end{aligned}$$

The stochastic increments are written in terms of

$$\begin{aligned} \check{f}(\delta t, \mathbf{W}) &= [\nabla \cdot \Sigma(\delta t, \mathbf{W})]\delta t = \nabla \cdot \left[\sqrt{\frac{\eta(k_B T)\delta t}{\Delta V}}(\mathbf{W} + \mathbf{W}^T) \right], \\ \check{h}(\delta t, \tilde{W}) &= [\nabla \cdot \Psi(\delta t, \tilde{W})]\delta t = \nabla \cdot \left[\sqrt{\frac{2\chi\rho\mu_c^{-1}(k_B T)\delta t}{\Delta V}}\tilde{W} \right], \end{aligned}$$

where \tilde{W} and \mathbf{W} are vectors of standard Gaussian variables [18].

III-B. Higher-order temporal integrators. A good strategy for composing higher-order temporal integrators for the low Mach number equations is to use a linear combination of several projected Euler steps of the form (22). In this way, the higher-order integrators inherit the properties of the Euler step. In our case, this will be very useful in constructing conservative discretizations that strictly maintain the EOS constraint and only evaluate fluxes at states that strictly obey the EOS constraint.

The incorporation of stochastic forcing in the Runge–Kutta temporal integrators that we use is described in [29; 18]; here we only summarize the resulting schemes. We note that the stochastic terms should be considered additive noise even though we evaluate them using an instantaneous state like multiplicative noise [73].

Explicit trapezoidal rule. A weakly second-order temporal integrator for (18)–(19) is provided by the *explicit trapezoidal rule*, in which we first take a predictor Euler step

$$\begin{aligned} \mathbf{m}^n &= [\mathcal{R}_F^n(\Delta t, \tilde{\mathbf{W}}^n)](\tilde{\mathbf{m}}^n), \\ \rho_1^{*,n+1} &= \rho_1^n + \Delta t \bar{h}^n + \check{h}^n(\Delta t, \tilde{\mathbf{W}}^n), \end{aligned} \quad (23)$$

$$\tilde{\mathbf{m}}^{*,n+1} = \mathbf{m}^n + \Delta t \bar{\mathbf{f}}^n + \check{\mathbf{f}}^n(\Delta t, \mathbf{W}^n). \quad (24)$$

The corrector step is a linear combination of the predictor and another Euler update,

$$\begin{aligned} \mathbf{m}^{*,n+1} &= [\mathcal{R}_F^{*,n+1}(\Delta t, \tilde{\mathbf{W}}^n)](\tilde{\mathbf{m}}^{*,n+1}), \\ \rho_1^{n+1} &= \frac{1}{2}\rho_1^n + \frac{1}{2}[\rho_1^{*,n+1} + \Delta t \bar{h}^{*,n+1} + \check{h}^{*,n+1}(\Delta t, \tilde{\mathbf{W}}^n)], \end{aligned} \quad (25)$$

$$\tilde{\mathbf{m}}^{n+1} = \frac{1}{2}\mathbf{m}^n + \frac{1}{2}[\mathbf{m}^{*,n+1} + \Delta t \bar{\mathbf{f}}^{*,n+1} + \check{\mathbf{f}}^{*,n+1}(\Delta t, \mathbf{W}^n)], \quad (26)$$

and reuses the same random numbers \mathbf{W}^n and $\tilde{\mathbf{W}}^n$ as the predictor step.

Note that both the predicted and the corrected values for density and concentration obey the EOS. We numerically observe that the trapezoidal rule does exhibit a slow but systematic numerical drift in the EOS, and therefore, it is necessary to use the correction procedure described in Section III-C at the end of each time step. The analysis in [18] indicates that for the incompressible case the trapezoidal scheme exhibits second-order weak accuracy in the nonlinear and linearized settings.

Explicit midpoint rule. An alternative second-order scheme is the *explicit midpoint rule*, which can be summarized as follows. First we take a projected Euler step to estimate midpoint values (denoted here with superscript $\star, n + 1/2$),

$$\begin{aligned} \mathbf{m}^n &= [\mathcal{R}_F^n(\frac{1}{2}\Delta t, \tilde{\mathbf{W}}_1^n)](\tilde{\mathbf{m}}^n), \\ \rho_1^{*,n+1/2} &= \rho_1^n + \frac{1}{2}\Delta t \bar{h}^n + \check{h}^n(\frac{1}{2}\Delta t, \tilde{\mathbf{W}}_1^n), \\ \tilde{\mathbf{m}}^{*,n+1/2} &= \mathbf{m}^n + \frac{1}{2}\Delta t \bar{\mathbf{f}}^n + \check{\mathbf{f}}^n(\frac{1}{2}\Delta t, \mathbf{W}_1^n), \end{aligned} \quad (27)$$

and then we complete the time step with another Euler-like update

$$\begin{aligned} \mathbf{m}^{*,n+1/2} &= [\mathcal{R}_F^{n+1/2}(\Delta t, \tilde{\mathbf{W}}^n)](\tilde{\mathbf{m}}^{*,n+1/2}), \\ \rho_1^{n+1} &= \rho_1^n + \Delta t \bar{h}^{*,n+1/2} + \check{h}^{*,n+1/2}(\Delta t, \tilde{\mathbf{W}}^n), \\ \tilde{\mathbf{m}}^{n+1} &= \mathbf{m}^n + \Delta t \bar{\mathbf{f}}^{*,n+1/2} + \check{\mathbf{f}}^{*,n+1/2}(\Delta t, \mathbf{W}^n), \end{aligned} \quad (28)$$

where the standard Gaussian variates

$$\tilde{\mathbf{W}}^n = \frac{\tilde{\mathbf{W}}_1^n + \tilde{\mathbf{W}}_2^n}{\sqrt{2}}$$

and the vectors of standard normal variates $\tilde{\mathbf{W}}_1^n$ and $\tilde{\mathbf{W}}_2^n$ are independent and similarly for \mathbf{W}_1^n and \mathbf{W}_2^n . Note that $\tilde{\mathbf{W}}_1^n$ and \mathbf{W}_1^n are used in *both* the predictor and the corrector stages while $\tilde{\mathbf{W}}_2^n$ and \mathbf{W}_2^n are used in the corrector only. Physically, the random numbers $\mathbf{W}_1^n/\sqrt{2}$ (and similarly for $\tilde{\mathbf{W}}_1^n$) correspond to the increments of the underlying Wiener processes $\Delta\mathcal{B}_1 = \sqrt{\Delta t/2} \mathbf{W}_1^n$ over the first half of the time step, and the random numbers $\mathbf{W}_2^n/\sqrt{2}$ correspond to the Wiener increments for the second half of the time step [18].

Note that both the midpoint and the endpoint values for density and concentration obey the EOS. We numerically observe that the midpoint rule does not exhibit a systematic numerical drift in the EOS and can therefore be used without the correction procedure described in Section III-C. The analysis in [18] indicates that for the incompressible case the midpoint scheme exhibits second-order weak accuracy in the nonlinear setting. Furthermore, in the linearized setting, it reproduces the steady-state covariances of the fluctuating fields to third order in the time-step size.

Three-stage Runge–Kutta (RK3) rule. We have also tested and implemented the three-stage Runge–Kutta scheme that was used in [29; 73]. This scheme can be expressed as a linear combination of three Euler steps. The first stage is a predictor Euler step,

$$\begin{aligned} \mathbf{m}^n &= [\mathcal{R}_F^n(\Delta t, \tilde{\mathbf{W}}^n)](\tilde{\mathbf{m}}^n), \\ \rho_1^* &= \rho_1^n + \Delta t \bar{h}^n + \check{h}^n(\Delta t, \tilde{\mathbf{W}}^n), \end{aligned} \quad (29)$$

$$\tilde{\mathbf{m}}^* = \mathbf{m}^n + \Delta t \bar{\mathbf{f}}^n + \check{\mathbf{f}}^n(\Delta t, \mathbf{W}^n). \quad (30)$$

The second stage is a midpoint predictor

$$\begin{aligned} \mathbf{m}^* &= [\mathcal{R}_F^*(\Delta t, \tilde{\mathbf{W}}^{*,n})](\tilde{\mathbf{m}}^*), \\ \rho_1^{**} &= \frac{3}{4}\rho_1^n + \frac{1}{4}[\rho_1^* + \Delta t \bar{h}^* + \check{h}^*(\Delta t, \tilde{\mathbf{W}}^{*,n})], \end{aligned} \quad (31)$$

$$\tilde{\mathbf{m}}^{**} = \frac{3}{4}\mathbf{m}^n + \frac{1}{4}[\mathbf{m}^* + \Delta t \bar{\mathbf{f}}^* + \check{\mathbf{f}}^*(\Delta t, \mathbf{W}^{*,n})], \quad (32)$$

and a final corrector stage completes the time step

$$\mathbf{m}^{**} = [\mathcal{R}_F^{**}(\Delta t, \tilde{\mathbf{W}}^{**,n})](\tilde{\mathbf{m}}^{**}),$$

$$\rho_1^{n+1} = \frac{1}{3}\rho_1^n + \frac{2}{3}[\rho_1^{**} + \Delta t \bar{h}^{**} + \check{h}^{**}(\Delta t, \tilde{\mathbf{W}}^{**,n})], \quad (33)$$

$$\tilde{\mathbf{m}}^{n+1} = \frac{1}{3}\mathbf{m}^n + \frac{2}{3}[\mathbf{m}^{**} + \Delta t \bar{\mathbf{f}}^{**} + \check{\mathbf{f}}^{**}(\Delta t, \mathbf{W}^{**,n})]. \quad (34)$$

Here the stochastic fluxes between different stages are related to each other via

$$\begin{aligned} \mathbf{W}^n &= \mathbf{W}_1^n + \frac{2\sqrt{2} + \sqrt{3}}{5} \mathbf{W}_2^n, \\ \mathbf{W}^{*,n} &= \mathbf{W}_1^n + \frac{-4\sqrt{2} + 3\sqrt{3}}{5} \mathbf{W}_2^n, \\ \mathbf{W}^{**,n} &= \mathbf{W}_1^n + \frac{\sqrt{2} - 2\sqrt{3}}{10} \mathbf{W}_2^n, \end{aligned} \quad (35)$$

where \mathbf{W}_1^n and \mathbf{W}_2^n are independent and generated independently at each RK3 step (similarly for $\tilde{\mathbf{W}}$). The weights of \mathbf{W}_2^n are chosen to maximize the weak order of accuracy of the scheme while still using only two random samples of the stochastic fluxes per time step [18].

The RK3 method is third-order accurate deterministically and stable even in the absence of diffusion/viscosity (i.e., for advection-dominated flows). Note that the predicted, the midpoint, and the endpoint values for density and concentration all obey the EOS. We numerically observe that the RK3 scheme does exhibit a systematic numerical drift in the EOS, and therefore, it is necessary to use the correction procedure described in [Section III-C](#) at the end of each time step. The analysis in [18] indicates that for the incompressible case the RK3 scheme exhibits second-order weak accuracy in the nonlinear setting. In the linearized setting, it reproduces the steady-state covariances of the fluctuating fields to third order in the time-step size.

III-C. EOS drift. While in principle our temporal integrators should strictly maintain the EOS, roundoff errors and the finite tolerance employed in the iterative Poisson solver lead to a small drift in the constraint that can, depending on the specific scheme, lead to an exponentially increasing violation of the EOS over many time steps. In order to maintain the EOS at all times to within roundoff tolerance, we periodically apply a globally conservative L_2 projection of ρ and ρ_1 onto the linear EOS constraint.

This projection step consists of correcting ρ_1 in cell k using

$$(\rho_1)_k \leftarrow A(\rho_1)_k - B(\rho_2)_k - \frac{1}{N} \sum_{k'} [A(\rho_1)_{k'} - B(\rho_2)_{k'}] + \frac{1}{N} \sum_{k'} (\rho_1)_{k'},$$

where N is the number of hydrodynamic cells in the system and

$$A = \frac{\bar{\rho}_1^2}{\bar{\rho}_1^2 + \bar{\rho}_2^2} \quad \text{and} \quad B = \frac{\bar{\rho}_1 \bar{\rho}_2}{\bar{\rho}_1^2 + \bar{\rho}_2^2}.$$

Note that the above update, while nonlocal in nature, conserves the total mass $\sum_{k'} (\rho_1)_{k'}$. A similar update applies to ρ_2 , or equivalently, $\rho = \rho_1 + \rho_2$.

IV. Spatial discretization

The spatial discretization we employ follows closely the spatial discretization of the constant-coefficient incompressible equations described in [73]. Therefore, we focus here on the differences, specifically, the use of conserved variables, the handling of the variable-density projection and variable-coefficient diffusion, and the imposition of the low Mach number constraint. Note that the handling of the stochastic momentum and mass fluxes is identical to that described in [73].

For simplicity of notation, we focus on two-dimensional problems with straightforward generalization to three spatial dimensions. Our spatial discretization follows the commonly used MAC approach [43], in which the scalar conserved quantities ρ and ρ_1 are defined on a regular Cartesian grid. The vector conserved variables $\mathbf{m} = \rho \mathbf{v}$ are defined on a staggered grid such that the k -th component of momentum is defined on the faces of the scalar-variable Cartesian grid in the k -th direction; see Figure 1. For simplicity of notation, we often denote the different components of velocity as $\mathbf{v} = (u, v)$ in two dimensions and $\mathbf{v} = (u, v, w)$ in three dimensions. The terms “cell-centered”, “edge-centered”, and “face-centered” refer to spatial locations relative to the underlying scalar grid. Our discretization is based on calculating fluxes on the faces of a finite-volume grid and is thus locally conservative. It is important to note, however, that for the MAC grid different control volumes are used for the scalars and the components of the momentum; see Figure 1.

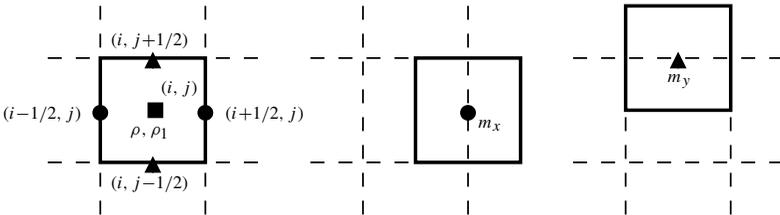


Figure 1. Staggered (MAC) finite-volume discretization on a uniform Cartesian two-dimensional grid. Left: control volume and flux discretization for cell-centered scalar fields such as densities ρ and ρ_1 . Middle: control volume for the x -component of face-centered vector fields such as m_x . Right: control volume for the y -component of face-centered vector fields such as m_y .

From the cell-centered ρ and ρ_1 , we can define other cell-centered scalar quantities, notably, the concentration $c_{i,j} = (\rho_1)_{i,j} / \rho_{i,j}$ and the transport quantities $\chi_{i,j}$ and $\eta_{i,j}$, which typically depend on the local density $\rho_{i,j}$ and concentration $c_{i,j}$ (and temperature for nonisothermal models) and can, in general, also depend on the spatial position of the cell $(x, y) = (i \Delta x, j \Delta y)$. In order to define velocities, we need to interpret the continuum relationship $\mathbf{m} = \rho \mathbf{v}$ on the staggered grid. This is done by defining face-centered scalar quantities obtained as an arithmetic average of the corresponding cell-centered quantities in the two neighboring cells. Specifically, we define

$$\rho_{i+1/2,j} = \frac{\rho_{i,j} + \rho_{i+1,j}}{2} \quad \text{and} \quad u_{i+1/2,j} = \frac{(m_x)_{i+1/2,j}}{\rho_{i+1/2,j}}, \quad (36)$$

except at physical boundaries, where the value is obtained from the imposed boundary conditions (see [Section IV-E](#)). Arithmetic averaging is only one possible interpolation from cells to faces [2]. In general, other forms of averaging such as a harmonic or geometric average or higher-order, wider stencils [1; 29] can be used. Most components of the spatial discretization can easily be generalized to other choices of interpolation. As we explain later, the use of linear averaging simplifies the construction of conservative advection.

IV-A. Diffusion. In this section, we describe the spatial discretization of the diffusive mass flux term $\nabla \cdot \rho \chi \nabla c$ in (9). The discretization is based on conservative centered differencing [29; 18]

$$\begin{aligned} (\nabla \cdot \rho \chi \nabla c)_{i,j} = & \Delta x^{-1} \left[\left(\rho \chi \frac{\partial c}{\partial x} \right)_{i+1/2,j} - \left(\rho \chi \frac{\partial c}{\partial x} \right)_{i-1/2,j} \right] \\ & + \Delta y^{-1} \left[\left(\rho \chi \frac{\partial c}{\partial y} \right)_{i,j+1/2} - \left(\rho \chi \frac{\partial c}{\partial y} \right)_{i,j-1/2} \right], \end{aligned} \quad (37)$$

where, for example,

$$\left(\rho \chi \frac{\partial c}{\partial x} \right)_{i+1/2,j} = (\rho_{i+1/2,j}) (\chi_{i+1/2,j}) \left(\frac{c_{i+1,j} - c_{i,j}}{\Delta x} \right) \quad (38)$$

and $\chi_{i+1/2,j}$ is an interpolated face-centered diffusion coefficient, for example, as done for ρ in (36),

$$\chi_{i+1/2,j} = \frac{\chi_{i,j} + \chi_{i+1,j}}{2},$$

except at physical boundaries, where the value is obtained from the imposed boundary conditions.

Regardless of the specific form of the interpolation operator, the same face-centered diffusion coefficient $\chi_{i+1/2,j}$ must be used when calculating the magnitude

of the stochastic mass flux on face $(i + 1/2, j)$,

$$(\Psi_x)_{i+1/2,j} = \sqrt{2\chi_{i+1/2,j}(\rho\mu_c^{-1})_{i+1/2,j}k_B T} \tilde{\mathcal{W}}_{i+1/2,j}.$$

This matches the covariance of the discrete stochastic mass increments $\nabla \cdot \Psi$ with the discretization of the diffusive dissipation operator $\nabla \cdot \rho\chi\nabla$ given in (37)–(38). This matching ensures discrete fluctuation-dissipation balance in the linearized setting [29]. Specifically, at thermodynamic equilibrium, the static covariance of the concentration is determined from the equilibrium value of $(\rho\mu_c^{-1})$ (thermodynamics) independently of the particular values of the transport coefficients (dynamics) as seen in (A-1) and dictated by statistical mechanics principles.

IV-B. Viscous terms. In [73], a Laplacian form of the viscous term $\eta\nabla^2\mathbf{v}$ is assumed, which is not applicable when viscosity is spatially varying and $\nabla \cdot \mathbf{v} = S \neq 0$. In two dimensions, the divergence of the viscous stress tensor in the momentum equation (8), neglecting bulk viscosity effects, is

$$\nabla \cdot [\eta(\nabla\mathbf{v} + \nabla^T\mathbf{v})] = \begin{bmatrix} 2\frac{\partial}{\partial x}(\eta\frac{\partial u}{\partial x}) + \frac{\partial}{\partial y}(\eta\frac{\partial u}{\partial y} + \eta\frac{\partial v}{\partial x}) \\ 2\frac{\partial}{\partial y}(\eta\frac{\partial v}{\partial y}) + \frac{\partial}{\partial x}(\eta\frac{\partial v}{\partial x} + \eta\frac{\partial u}{\partial y}) \end{bmatrix}. \quad (39)$$

The discretization of the viscous terms requires η at cell centers and edges (note that in two dimensions the edges are the same as the nodes $(i + 1/2, j + 1/2)$ of the grid). The value of η at a node is interpolated as the arithmetic average of the four neighboring cell centers,

$$\eta_{i+1/2,j+1/2} = \frac{1}{4}(\eta_{i,j} + \eta_{i+1,j+1} + \eta_{i+1,j} + \eta_{i,j+1}),$$

except at physical boundaries, where the values are obtained from the prescribed boundary conditions. The different viscous friction terms are discretized by straightforward centered differences. Explicitly, for the x -component of momentum,

$$\left[\frac{\partial}{\partial x} \left(\eta \frac{\partial u}{\partial x} \right) \right]_{i+1/2,j} = \Delta x^{-1} \left[\left(\eta \frac{\partial u}{\partial x} \right)_{i+1,j} - \left(\eta \frac{\partial u}{\partial x} \right)_{i,j} \right]$$

with

$$\left(\eta \frac{\partial u}{\partial x} \right)_{i,j} = \eta_{i,j} \left(\frac{u_{i+1/2,j} - u_{i-1/2,j}}{\Delta x} \right).$$

Similarly, for the term involving a second derivative in y ,

$$\left[\frac{\partial}{\partial y} \left(\eta \frac{\partial u}{\partial y} \right) \right]_{i+1/2,j} = \Delta y^{-1} \left[\left(\eta \frac{\partial u}{\partial y} \right)_{i+1/2,j+1/2} - \left(\eta \frac{\partial u}{\partial y} \right)_{i+1/2,j-1/2} \right]$$

with

$$\left(\eta \frac{\partial u}{\partial y} \right)_{i+1/2,j+1/2} = \eta_{i+1/2,j+1/2} \left(\frac{u_{i+1/2,j+1} - u_{i+1/2,j}}{\Delta y} \right).$$

A similar construction is used for the mixed-derivative term

$$\left[\frac{\partial}{\partial y} \left(\eta \frac{\partial v}{\partial x} \right) \right]_{i+1/2, j} = \Delta y^{-1} \left[\left(\eta \frac{\partial v}{\partial x} \right)_{i+1/2, j+1/2} - \left(\eta \frac{\partial v}{\partial x} \right)_{i+1/2, j-1/2} \right]$$

with

$$\left(\eta \frac{\partial v}{\partial x} \right)_{i+1/2, j+1/2} = \eta_{i+1/2, j+1/2} \left(\frac{v_{i+1, j+1/2} - v_{i, j+1/2}}{\Delta x} \right).$$

The stochastic-stress-tensor discretization is described in more detail in [73] and applies in the present context as well. For the low Mach number equations, just as for the compressible equations, the symmetric form of the stochastic stress tensor must be used in order to ensure discrete fluctuation-dissipation balance between the viscous dissipation and stochastic forcing. Additionally, when η is not spatially uniform, the same interpolated viscosity $\eta_{i+1/2, j+1/2}$ as used in the viscous terms must be used when calculating the amplitude in the stochastic forcing $\sqrt{\eta k_B T}$ at the edges (nodes) of the grid.

IV-C. Advection. It is challenging to construct spatiotemporal discretizations that conserve the total mass while remaining consistent with the equation of state [67; 65; 58] as ensured in the continuum context by the constraint (11). We demonstrate here how the special linear form of the constraint (13) can be exploited in the discrete context. Following [73], we spatially discretize the advective terms in (9) using a centered (skew-adjoint [54]) discretization

$$\begin{aligned} [\nabla \cdot (\rho_1 \mathbf{v})]_{i, j} &= \Delta x^{-1} [(\rho_1)_{i+1/2, j} u_{i+1/2, j} - (\rho_1)_{i-1/2, j} u_{i-1/2, j}] \\ &\quad + \Delta y^{-1} [(\rho_1)_{i, j+1/2} v_{i, j+1/2} - (\rho_1)_{i, j-1/2} v_{i, j-1/2}] \end{aligned} \quad (40)$$

and similarly for (12). We would like this discrete advection to maintain the equation of state (13) at the discrete level, that is, maintain the constraint relating $(\rho_1)_{i, j}$ and $(\rho_2)_{i, j}$ in every cell (i, j) .

Because the different dimensions are decoupled and the divergence is simply the sum of the one-dimensional difference operators, it is sufficient to consider (9) in one spatial dimension. The method-of-lines discretization is given by the system of ODEs, one differential equation per cell i ,

$$(\partial_t \rho_1)_i = \Delta x^{-1} (F_{i+1/2} - F_{i-1/2}) - \Delta x^{-1} [(\rho_1)_{i+1/2} u_{i+1/2} - (\rho_1)_{i-1/2} u_{i-1/2}]$$

and similarly for $(\partial_t \rho_2)_i$. As a shorthand, denote the quantity that appears in (13) with

$$\delta = \frac{\rho_1}{\bar{\rho}_1} + \frac{\rho_2}{\bar{\rho}_2} = 1.$$

If we use the linear interpolation (36) to calculate face-centered densities, then because of the linearity of the EOS the face-centered densities obey the EOS if the

cell-centered ones do since $\delta_{i+1/2} = (\delta_i + \delta_{i+1})/2 = 1$. The rate of change of δ in cell i is

$$\begin{aligned}\Delta x(\partial_t \delta)_i &= (\rho^{-1} \beta)(F_{i+1/2} - F_{i-1/2}) - [\delta_{i+1/2} u_{i+1/2} - \delta_{i-1/2} u_{i-1/2}] \\ &= (\rho^{-1} \beta)(F_{i+1/2} - F_{i-1/2}) - (u_{i+1/2} - u_{i-1/2}) = 0.\end{aligned}$$

This simple calculation shows that the EOS constraint $\delta = 1$ is obeyed discretely in each cell at all times if it is initially satisfied and the velocities used to advect mass obey the discrete version of the constraint (11),

$$\begin{aligned}S_{i,j} &= \Delta x^{-1}(u_{i+1/2,j} - u_{i-1/2,j}) + \Delta y^{-1}(v_{i,j+1/2} - v_{i,j-1/2}) \\ &= \left(\frac{1}{\rho_1} - \frac{1}{\rho_2}\right) [\Delta x^{-1}(F_{i+1/2,j} - F_{i-1/2,j}) + \Delta y^{-1}(F_{i,j+1/2} - F_{i,j-1/2})],\end{aligned}\quad (41)$$

in two dimensions. Our algorithm ensures that advective terms are always evaluated using a discrete velocity field that obeys this constraint. This is accomplished by using a discrete projection operator as we describe in the next section.

The spatial discretization of the advection terms in the momentum equation (8) is constructed using centered differences on the corresponding shifted (staggered) grid as described in [73]. For example, for the x -component of momentum $m_x = \rho u$,

$$\begin{aligned}[\nabla \cdot (m_x \mathbf{v})]_{i+1/2,j} &= \Delta x^{-1}[(m_x u)_{i+1,j} - (m_x u)_{i,j}] \\ &\quad + \Delta y^{-1}[(m_x v)_{i+1/2,j+1/2} - (m_x v)_{i+1/2,j-1/2}],\end{aligned}\quad (42)$$

where simple averaging is used to interpolate momenta to the cell centers and edges (nodes) of the grid, for example,

$$(m_x u)_{i,j} = (m_x)_{i,j} u_{i,j} = \frac{(m_x)_{i-1/2,j} + (m_x)_{i+1/2,j}}{2} \frac{u_{i-1/2,j} + u_{i+1/2,j}}{2}.\quad (43)$$

Because of the linearity of the interpolation procedure, the interpolated discrete velocity used to advect m_x obeys the constraint (41) on the shifted grid with a right-hand side $S_{i+1/2,j}$ interpolated using the same arithmetic average used to interpolate the velocities. In particular, in the incompressible case, all variables, including momentum, are advected using a discretely divergence-free velocity, ensuring discrete fluctuation-dissipation balance [73; 18].

It is well-known that the centered discretization of advection we employ here is not robust for advection-dominated flows, and higher-order limiters and upwinding schemes are generally preferred in the deterministic setting [5]. However, these more robust advection schemes add artificial dissipation, which leads to a violation of discrete fluctuation-dissipation balance [18]. In Appendix B, we describe an alternative filtering procedure that can be used to handle strong advection while continuing to use centered differencing.

IV-D. Discrete projection. We now briefly discuss the spatial discretization of the affine operator \mathcal{R}_S defined by (16) as used in our explicit temporal integrators. The discrete projection takes a face-centered (staggered) discrete velocity field $\tilde{\mathbf{v}} = (\tilde{u}, \tilde{v})$ and a velocity divergence S and projects $\mathbf{v} = \mathcal{R}_S(\tilde{\mathbf{v}})$ onto the constraint (41) in a conservative manner. Specifically, the projection consists of finding a cell-centered discrete scalar field ϕ such that

$$\rho \mathbf{v} = \rho \tilde{\mathbf{v}} - \nabla \phi \quad \text{and} \quad \nabla \cdot \mathbf{v} = S,$$

where the gradient is discretized using centered differences, e.g.,

$$v_{i+1/2,j} = \tilde{v}_{i+1/2,j} - \left(\frac{1}{\rho_{i+1/2,j}} \right) \left(\frac{\phi_{i+1,j} - \phi_{i,j}}{\Delta x} \right). \quad (44)$$

The pressure correction ϕ is the solution to the variable-coefficient discrete Poisson equation,

$$\begin{aligned} & \frac{1}{\Delta x} \left[\left(\frac{1}{\rho_{i+1/2,j}} \right) \left(\frac{\phi_{i+1,j} - \phi_{i,j}}{\Delta x} \right) - \left(\frac{1}{\rho_{i-1/2,j}} \right) \left(\frac{\phi_{i,j} - \phi_{i,j-1}}{\Delta x} \right) \right] \\ & + \frac{1}{\Delta y} \left[\left(\frac{1}{\rho_{i,j+1/2}} \right) \left(\frac{\phi_{i,j+1} - \phi_{i,j}}{\Delta y} \right) - \left(\frac{1}{\rho_{i,j-1/2}} \right) \left(\frac{\phi_{i,j} - \phi_{i,j-1}}{\Delta y} \right) \right] \\ & = S_{i,j} - \left[\left(\frac{\tilde{u}_{i+1/2,j} - \tilde{u}_{i-1/2,j}}{\Delta x} \right) + \left(\frac{\tilde{v}_{i,j+1/2} - \tilde{v}_{i,j-1/2}}{\Delta y} \right) \right], \quad (45) \end{aligned}$$

which can be solved efficiently using a standard multigrid approach [2].

IV-E. Boundary conditions. The handling of different types of boundary conditions is relatively straightforward when a staggered grid is used and the physical boundaries are aligned with the cell boundaries for the scalar grid. Interpolation is not used to obtain values for faces, nodes, or edges of the grid that lie on a physical boundary since this would require “ghost” values at cell centers lying outside of the physical domain. Instead, whenever a value of a physical variable is required at a face, node, or edge lying on a physical boundary, the boundary condition is used to obtain that value. Similarly, centered differences for the diffusive and viscous fluxes that require values outside of the physical domain are replaced by one-sided differences that only use values from the interior cell bordering the boundary and boundary values.

For example, if the concentration is specified at the face $(i+1/2, j)$, the diffusive flux discretization (38) is replaced with

$$\left(\rho \chi \frac{\partial c}{\partial x} \right)_{i+1/2,j} = (\rho_{i+1/2,j}) (\chi_{i+1/2,j}) \left(\frac{c_{i+1/2,j} - c_{i,j}}{\Delta x/2} \right),$$

where $c_{i+1/2,j}$ is the specified boundary value, the density $\rho_{i+1/2,j}$ is obtained from $c_{i+1/2,j}$ using the EOS constraint, and the diffusion coefficient $\chi_{i+1/2,j}$ is

calculated at the specified values of concentration and density. Similar straightforward one-sided differencing is used for the viscous fluxes. As discussed in [73], the use of second-order one-sided differencing is not required to achieve global second-order accuracy and would make the handling of the stochastic fluxes more complicated because it leads to a nonsymmetric discrete Laplacian. Note that for the nonlinear low Mach number equations our approach is subtly different from linearly extrapolating the value in the ghost cell $c_{i+1,j} = 2c_{i+1/2,j} - c_i$. Namely, the extrapolated value might be unphysical, and it might not be possible to evaluate the EOS or transport coefficients at the extrapolated concentration. For Neumann-type or zero-flux boundary conditions, the corresponding diffusive flux is set to zero for any faces of the corresponding control volume that lie on physical boundaries, and values in cells outside of the physical domain are never required. The corresponding handling of the stochastic fluxes is discussed in detail in [73].

The evaluation of advective fluxes for the scalars requires normal components of the velocity at the boundary. For faces of the grid that lie on a physical boundary, the normal component of the velocity is determined from the value of the diffusive mass flux at that face using (15). Therefore, these velocities are not independent variables and are not solved for or modified by the projection \mathcal{R}_S . Specifically, the discrete pressure ϕ is only defined at the cell centers in the interior of the grid, and the discrete Poisson equation (45) is only imposed on the interior faces of the grid. Therefore, no explicit boundary conditions for ϕ are required when the staggered grid is used, and the natural homogeneous Neumann conditions are implied. Advective momentum fluxes are only evaluated on the interior faces and thus do not use any values outside of the physical domain.

IV-F. Summary of Euler–Maruyama method. By combining the spatial discretization described above with one of the temporal integrators described in Section III, we can obtain a finite-volume solver for the fluctuating low Mach equations. For the benefit of the reader, here we summarize our implementation of a single Euler step (22). This forms the core procedure that the higher-order Runge–Kutta schemes employ several times during one time step.

- (1) Generate the vectors of standard Gaussian variates \mathbf{W}^n and $\tilde{\mathbf{W}}^n$.
- (2) Calculate diffusive and stochastic fluxes for ρ_1 using (38),

$$\mathbf{F}^n = (\rho\chi\nabla c)^n + \Psi^n(\Delta t, \tilde{\mathbf{W}}^n).$$

- (3) Solve the Poisson problem (45) with

$$S^n = -\left(\frac{1}{\bar{\rho}_1} - \frac{1}{\bar{\rho}_2}\right)\nabla \cdot \mathbf{F}^n$$

to obtain the velocity \mathbf{v}^n from $\tilde{\mathbf{v}}^n = \tilde{\mathbf{m}}^n/\rho^n$ using (44), enforcing $\nabla \cdot \mathbf{v}^n = S^n$.

- (4) Calculate viscous and stochastic momentum fluxes using (39),

$$\nabla \cdot [\eta(\nabla \mathbf{v} + \nabla^T \mathbf{v})]^n + \nabla \cdot [\boldsymbol{\Sigma}^n(\Delta t, \mathbf{W}^n)].$$

- (5) Calculate external forcing terms for the momentum equation such as the contribution $-\rho^n \mathbf{g}$ due to gravity.
- (6) Calculate advective fluxes for mass and momentum using (40) and (42).
- (7) Update mass and momentum densities, including advective, diffusive, stochastic, and external forcing terms, to obtain ρ^{n+1} , ρ_1^{n+1} , and $\tilde{\mathbf{m}}^{n+1}$. Note that this update preserves the EOS constraint as explained in Section IV-C.

We have tested and validated the accuracy of our methods and numerical implementation using a series of standard deterministic tests as well as by examining the equilibrium spectrum of the concentration and velocity fluctuations [29; 73; 18]. The next two sections present further verification and validation in the context of nonequilibrium systems.

V. Giant concentration fluctuations

Advection of concentration by thermal velocity fluctuations in the presence of large concentration gradients leads to the appearance of *giant fluctuations* of concentration, as has been studied theoretically and experimentally for more than a decade [77; 12; 75; 74]. These giant fluctuations were previously simulated in the absence of gravity in three dimensions by some of us in [73], and good agreement was found with experimental results [75]. In those previous studies, the incompressible equations were used; that is, it was assumed that concentration was a passively advected scalar. However, it is more physically realistic to account for the fact that the properties of the fluid, notably the density and the transport coefficients, depend on the concentration. In [12], a series of experiments were performed to study the temporal evolution of giant concentration fluctuations during the diffusive mixing of water and glycerol, starting with a glycerol mass fraction of $c = 0.39$ in the bottom half of the experimental domain and $c = 0$ in the top half. Because it is essentially impossible to analytically solve the full system of fluctuating equations in the presence of spatial inhomogeneity and nontrivial boundary conditions, the existing theoretical analysis of the diffusive mixing process [77] makes a quasiperiodic constant-coefficient incompressible approximation.

For simplicity, in this section, we focus on a time-independent problem and study the spectrum of steady-state concentration fluctuations in a mixture under gravity in the presence of a constant concentration gradient. This extends the study reported in [73] to account for the fact that the density, viscosity, and diffusion coefficient depend on the concentration. For simplicity, we do two-dimensional simulations since for this problem there is no difference between the spectra of

concentration fluctuations in two and three dimensions [73] (note, however, that in real space, unlike in Fourier space, the effect of the fluctuations on the transport is very different in two and three dimensions). Furthermore, in these simulations, we do not include a stochastic flux in the concentration equation; i.e., we set $\Psi = 0$ so that all fluctuations in the concentration arise from being out of thermodynamic equilibrium. With this approximation, we do not need to model the chemical potential of the mixture and obtain μ_c . This formulation is justified by the fact that it is known experimentally that the nonequilibrium fluctuations are much larger than the equilibrium ones for the conditions we consider [12].

In the simple linearized theory presented in Section A.2, several approximations are made. The first one is that a quasiperiodic approximation is used even though the actual system is not periodic in the y -direction. This source of error has already been studied numerically in [73]. We also use a Boussinesq approximation where it is assumed that $\bar{\rho}_1 = \rho_0 + \Delta\rho/2$ and $\bar{\rho}_2 = \rho_0 - \Delta\rho/2$, where $\Delta\rho$ is a small density difference between the two fluids, $\Delta\rho/\rho_0 \ll 1$, so that density is approximately constant and $\beta \ll 1$. More precisely, in the Boussinesq model, the gravity term in the velocity equation only enters through the product βg , so the approximation consists of taking the limit $\beta \rightarrow 0$ and $g \rightarrow \infty$ while keeping the product βg fixed. The final approximation made in the simple theory is that the transport coefficients, i.e., the viscosity and diffusion coefficients, are assumed to be constant. Here we evaluate the validity of the constant-coefficient constant-density approximation (ρ , η , and χ constant and $\beta \rightarrow 0$), as well as the constant-density (Boussinesq) approximation alone (ρ constant and $\beta \rightarrow 0$ but variable η and χ), by comparing with the solution to the complete low Mach number equations (ρ , η , χ , and β variable).

V-A. Simulation parameters. We base our parameters on the experimental studies of diffusive mixing in a water-glycerol mixture as reported in [12]. The physical domain is $1 \text{ cm} \times 0.25 \text{ cm}$ discretized on a uniform 128×32 two-dimensional grid with a thickness of 1 cm along the z -direction. Gravity is applied in the negative y - (vertical) direction. Reservoir boundary conditions (15) are applied in the y -direction and periodic boundary conditions in the x -direction. We set the concentration to $c = 0.39$ on the bottom boundary and $c = 0$ on the top boundary and apply no-slip boundary conditions for the velocity at both boundaries. The initial condition is $c(t = 0) = 0.39(y/0.25 - 1)$, which is close to the deterministic steady-state profile. A very good fit to the experimental equation of state (dependence of density on concentration at standard temperature and pressure) over the whole range of concentrations of interest is provided by the EOS (13) with the density of water set to $\bar{\rho}_2 = 1 \text{ g/cm}^3$ and the density of glycerol set to $\bar{\rho}_1 = 1.29 \text{ g/cm}^3$. In these simulations, the magnitude of the velocity fluctuations is very small, and we did not use filtering (see Appendix B).

Experimentally, the dependence of viscosity on glycerol mass fraction has been fit to an exponential function [12], which we approximate with a quadratic function over the range of concentrations of interest

$$\eta(c) = \rho(c)\nu(c) = \rho_0\nu_0 \exp(2.06c + 2.32c^2) \approx \rho_0\nu_0(1.0 + 0.66c + 12c^2), \quad (46)$$

where $\rho_0 = 1 \text{ g/cm}^3$ and experimental measurements estimate $\nu_0 \approx 10^{-2} \text{ cm}^2/\text{s}$. The dependence of the diffusion coefficient on the concentration has been studied experimentally [19], but it is strongly affected by thermal fluctuations and spatial confinement [24; 21; 26]. We approximate the dependence assuming a Stokes–Einstein relation [25], which is in reasonable agreement with the experimental results in [19] over the range of concentrations of interest here; we can write it as

$$\chi(c) = \frac{\chi_0\eta_0}{\eta(c)} \approx \chi_0(1.0 - 2.2c + 1.2c^2), \quad (47)$$

where experimental estimates for water-glycerol mixtures give $\chi_0 \approx 10^{-5} \text{ cm}^2/\text{s}$, with a Schmidt number $\text{Sc} = \nu_0/\chi_0 \approx 10^3$. This very large separation of scales between mass and momentum diffusion is not feasible to simulate with our explicit temporal integration methods. Referring back to the simplified theory (A-7), which in this case can be simplified further to

$$S_{c,c}(k_x, k_y = 0) = \langle (\widehat{\delta c})(\widehat{\delta c})^* \rangle \approx \frac{\nu}{\nu + \chi} \frac{k_B T}{(\chi \eta k_x^4 + h_{\parallel} \rho g \beta)} h_{\parallel}^2, \quad (48)$$

we see that for $\nu \gg \chi$ the shape of the spectrum of the steady-state concentration fluctuations, and in particular the cutoff wavenumber due to gravity, is determined from the product $\chi\nu$ and not χ and ν individually. Therefore, as also done in [73], we choose χ_0 and ν_0 so that $\chi(\bar{c})\nu(\bar{c})$ is kept at the physical value of $10^{-7} \text{ g} \cdot \text{cm}/\text{s}^2$, but the Schmidt number is reduced by two orders of magnitude, $S_c = \rho_0^{-1}\eta(\bar{c})/\chi(\bar{c}) = 10$, where $\bar{c} = 0.39/2$ is an estimate of the average concentration. The condition $\eta(\bar{c}) \approx 10^{-3} \text{ g}/(\text{cm} \cdot \text{s})$ and $\chi(\bar{c}) \approx 10^{-4} \text{ cm}^2/\text{s}$ gives our simulation parameters $\nu_0 \approx 6.1 \times 10^{-4} \text{ cm}^2/\text{s}$ and $\chi_0 \approx 1.6 \times 10^{-4} \text{ cm}^2/\text{s}$.

The physical value for gravity is $g \approx 10^3 \text{ cm}/\text{s}^2$, and the solutal expansion coefficient $\beta(\bar{c}) \approx 0.234$ follows from $\bar{\rho}_1$ and $\bar{\rho}_2$. When employing the Boussinesq approximation, in which gravity only enters through the product βg , we set $\rho_1 = 1.054$ and $\rho_2 = 1.044$ so that $\beta = 0.01$ and increase gravity by the corresponding factor to $g = 2.34 \cdot 10^4 \text{ cm}/\text{s}^2$ in order to keep βg fixed at the physical value. We also performed simulations with a weaker gravity, $g \approx 10^2 \text{ cm}/\text{s}^2$, which enhances the giant fluctuations.

V-B. Results. We employ the explicit midpoint temporal integrator (which we recall is third-order accurate for static covariances) and set $\Delta t = 0.005 \text{ s}$, which results in a diffusive Courant number $\nu\Delta t/\Delta x^2 \approx 0.1$. We skip the first 50,000

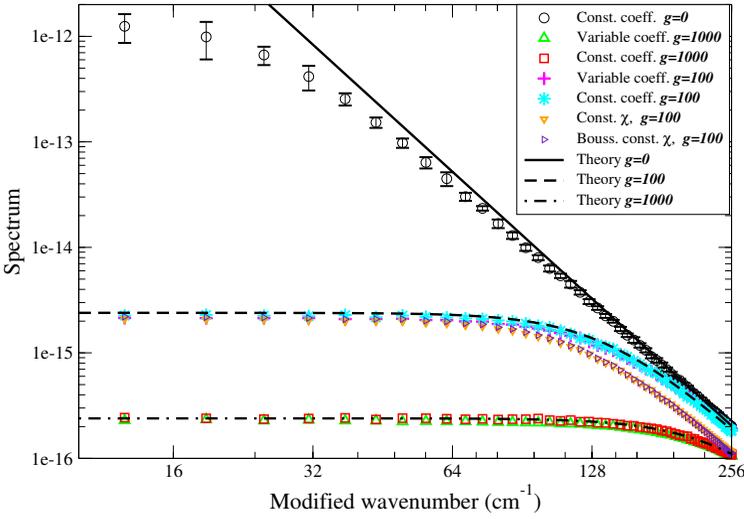


Figure 2. Comparison between the simple theory (A-7) (lines) and numerical results (symbols). Results are shown for standard gravity $g \approx 10^3 \text{ cm/s}^2$ (the cutoff wavenumber $k_g \approx 246 \text{ cm}^{-1}$) for the complete variable-coefficient variable-density low Mach model (green upward triangles) and the constant-coefficient constant-density approximation (red squares). Also shown are results for a weaker gravity, $g \approx 10^2 \text{ cm/s}^2$ (the cutoff wavenumber $k_g \approx 138 \text{ cm}^{-1}$), for the complete low Mach model (magenta pluses) and the constant-coefficient constant-density approximation (cyan stars). For comparison, results for $g \approx 10^2 \text{ cm/s}^2$ with variable viscosity $\eta(c)$ but constant diffusion coefficient $\chi(c) = \chi_0$ are also shown, for variable density (orange downward triangles) and the constant-density (Boussinesq) approximation (indigo right-facing triangles). Finally, results for no gravity are shown in the constant-coefficient approximation (black circles).

time steps (about five diffusion crossing times) and then collect samples from the subsequent 50,000 time steps. We repeat this eight times to increase the statistical accuracy and estimate error bars. To compare to the theory (A-7), we set the concentration gradient to $h_{\parallel} = 0.39/0.25 \text{ cm}^{-1}$ and evaluate $\rho \approx 1.05 \text{ g/cm}^3$ at $c = 0.39/2$ from the equation of state. When computing the theory, we account for errors in the discrete approximation to the continuum Laplacian by using the effective wavenumber

$$k_{\perp} = k_x \frac{\sin(k_x \Delta x/2)}{(k_x \Delta x/2)} \quad (49)$$

instead of the actual discrete wavenumber k_x [73].

The results for the static spectrum of concentration fluctuations $S_{c,c}(k_x, k_y = 0) = \langle (\widehat{\delta c})(\widehat{\delta c})^* \rangle$ as a function of the modified wavenumber k_{\perp} (49) are shown in Figure 2. When there is no gravity, we see the characteristic giant-fluctuation power-law spectrum of the fluctuations, modulated at small wavenumbers due to the presence of the physical boundaries [73]. When gravity is present, fluctuations at wavenumber

below the cutoff $k_g = [h_{\parallel} \rho g \beta / (\eta \chi)]^{1/4}$ are suppressed. If we use a constant-coefficient approximation, in which we reduce $\beta = 0.01$ so that $\rho \approx \rho(\bar{c})$ and also fix the transport coefficients at $\eta(c) = \eta(\bar{c})$ and $\chi(c) = \chi(\bar{c})$, we observe good agreement with the quasiperiodic theory (A-7). When we make the transport coefficients dependent on the concentration as in (46) and (47), we observe a rather small change in the spectrum. This is perhaps not unexpected because the simplified theory (48) shows that only the product $\chi \eta$, and not χ and η individually, matters. Since we used the Stokes–Einstein relation $\chi(c)\eta(c) = \rho_0 \chi_0 \nu_0 = \text{constant}$ to select the concentration dependence of the diffusion coefficient, the value of $\chi \eta$ is constant throughout the physical domain. For comparison, in Figure 2 we show results from a simulation where we keep the concentration dependence of the viscosity (46) but set the diffusion coefficient to a constant value, $\chi(c) = \chi_0$, and we observe a more significant change in the spectrum. Further employing the Boussinesq approximation makes little difference, showing that the primary effect here comes from the dependence of the transport coefficients on concentration.

This shows that, under the sort of parameters present in the experiments on diffusive mixing in water-glycerol mixture, it is reasonable to make the Boussinesq incompressible approximation; however, the spatial dependence of the viscosity and diffusion coefficient cannot in general be ignored if quantitative agreement is desired. In particular, time-dependent quantities such as dynamic spectra [76; 11] depend on the values of χ and η and not just their product, and are thus expected to be more sensitive to the details of their concentration dependence. Even though the constant-coefficient approximation gives qualitatively the correct shape and a better choice of the constant transport coefficients may improve its accuracy, there is no simple procedure to a priori estimate what parameters should be used (but see [77] for a proposal to average the constant-coefficient theory over the domain). A direct comparison with experimental results is not possible until multiscale temporal integrators capable of handling the extreme separation of time scales between mass and momentum diffusion are developed. At present, this has only been accomplished in the constant-coefficient incompressible limit ($\beta = 0$) [26], and it remains a significant challenge to accomplish the same for the complete low Mach number system.

VI. Diffusive mixing in hard-disk and hard-sphere fluids

In this section, we study the appearance of giant fluctuations during *time-dependent* diffusive mixing. As a validation of the low Mach number fluctuating equations and our algorithm, we perform simulations of diffusive mixing of two fluids of different densities in two dimensions. We find excellent agreement between the results of low Mach number (continuum) simulations and hard-disk molecular dynamics (particle)

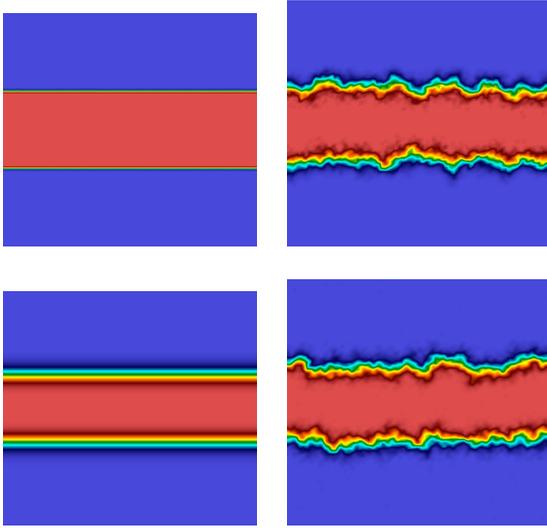


Figure 3. Diffusive mixing between two fluids of unequal densities, $R = \rho_2/\rho_1 = 4$, with coloring based on concentration: red for the pure first component, $c = 1$, and blue for the pure second component, $c = 0$. A smoothed shading is used for the coloring to eliminate visual discretization artifacts. The simulation domain is periodic and contains 128^2 hydrodynamic (finite-volume) cells. The top left panel shows the initial configuration, which is the same for all simulations reported here. The top right panel shows the final configuration at time $t = 5800$ as obtained using molecular dynamics. The bottom left panel shows the final configuration obtained using deterministic hydrodynamics while the bottom right panel shows the final configuration obtained using fluctuating hydrodynamics.

simulations. This nontrivial test clearly demonstrates the usefulness of low Mach number models as a coarse-grained mesoscopic model for problems where sound waves can be neglected.

Our simulation setup is illustrated in Figure 3. We consider a periodic square box of length L along both the x - (horizontal) and y - (vertical) directions and initially place all of the fluid of species one (colored red) in the middle third of the domain; i.e., we set $c = 1$ for $L/3 \leq y \leq 2L/3$, and $c = 0$ otherwise, as shown in the top left panel of the figure. The two fluids mix diffusively, and at the end of the simulation, the concentration field shows a *rough diffusive interface* as confirmed by molecular dynamics simulations shown in the top right panel of the figure. The deterministic equations of diffusive mixing reduce to a one-dimensional model due to the translational symmetry along the x -axis and would yield a *flat* diffusive interface as illustrated in the bottom left panel of the figure. However, fluctuating hydrodynamics correctly reproduces the interface roughness as illustrated in the bottom right panel of the figure and demonstrated quantitatively below.

We consider here a binary hard-disk mixture in two dimensions. We use arbitrary (molecular) units of length, time, and mass for convenience. All hard disks had

a diameter $\sigma = 1$ in arbitrary units, and we set the temperature at $k_B T = 1$. The molecular mass for the first fluid component was fixed at $m_1 = 1$ and for the second component at $m_2 = R m_1$. For mass ratio $R = 1$, the two types of disks are mechanically identical, and therefore, the species label is simply a red-blue coloring of the particles. In this case, $\bar{\rho}_2 = \bar{\rho}_1$ and the low Mach number equations reduce to the incompressible equations of fluctuating hydrodynamics with a passively advected concentration field. For the case of unequal particle masses, mechanical equilibrium is obtained if the pressures in the two fluid components are the same. It is well-known from statistical mechanics that for hard disks or hard spheres the pressure is

$$P = Y(\phi) \cdot n \cdot k_B T,$$

where $n = N/V$ is the number density and $Y(\phi)$ is a prefactor that only depends on the packing fraction $\phi = n(\pi\sigma^2/4)$ and not on the molecular mass. Therefore, for a mixture of disks or spheres with equal diameters, at constant pressure, the number density and the packing fraction ϕ are constant independent of the composition. The equation of state at constant pressure and temperature is therefore

$$1 = \frac{n_1}{n} + \frac{n_2}{n} = \frac{\rho_1}{nm_1} + \frac{\rho_2}{nm_2},$$

which is exactly of the form (13) with $\bar{\rho}_1 = nm_1$ and $\bar{\rho}_2 = nm_2$. The chemical potential of such a mixture has the same concentration dependence as a low-density gas mixture [49],

$$\mu_c^{-1} k_B T = c(1 - c)[cm_2 + (1 - c)m_1].$$

VI-A. Hard-disk molecular dynamics. In order to validate the predictions of our low Mach number model, we performed hard-disk molecular dynamics (HDMD) simulations of diffusive mixing using a modification of the public-domain code developed by the authors of [70]. We used a packing fraction of $\phi = 0.6$ for all simulations reported here. This packing fraction is close to the freezing transition point but is known to be safely in the (dense) gas phase (there is no liquid phase for a hard-disk fluid). The initial particle positions were generated using a nonequilibrium molecular-dynamics simulation as in the hard-particle packing algorithm described in [27; 28]. After the initial configuration was generated, the disks were assigned a species according to their y -coordinate, and the mixing simulation was performed using event-driven molecular dynamics.

In order to convert the particle data to hydrodynamic data comparable to that generated by the fluctuating hydrodynamics simulations, we employed a grid of N_c^2 hydrodynamic cells that were each a square of linear dimension $L_c = 10\sigma$. At the chosen packing fraction $\phi = 0.6$, this corresponds to about 76 disks per hydrodynamic cell, which is deemed a reasonable level of coarse-graining for the equations

of fluctuating hydrodynamics to be a reasonably accurate model while still keeping the computational demands of the simulations manageable. We performed HDMD simulations for systems of size $N_c = 64$ and $N_c = 128$ cells and simulated the mixing process to a final simulation time of $t = 5800$ units. The largest system simulated had about 1.25 million disks (each simulation took about five days of CPU time), which is well into the “hydrodynamic” rather than “molecular” scale.

Every 58 units of time, particle data was converted to hydrodynamic data for the purposes of analysis and comparison to hydrodynamic calculations. There is not a unique way of coarse-graining particle data to hydrodynamic data [78; 35]; however, we believe that the large-scale (giant) concentration fluctuations studied here are *not* affected by the particular choice. We therefore used a simple method consistent with the philosophy of finite-volume conservative discretizations. Specifically, we coarse-grained the particle information by sorting the particles into hydrodynamic cells based on the position of their centroid as if they were point particles. We then calculated ρ_1 and ρ_2 in each cell based on the total mass of each species contained inside the given cell. Since all particles have equal diameter, other definitions that take into account the particle shape and size give similar results.

VI-B. Hard-disk hydrodynamics. We now turn to hydrodynamic simulations of the diffusive mixing of hard disks. Our hydrodynamic calculations use the same grid of cells used to convert particle to hydrodynamic data. The only input required for the hydrodynamic calculations, in addition to those provided by equilibrium statistical mechanics, are the transport coefficients of the fluid as a function of concentration, specifically, the shear viscosity η and the diffusion coefficient χ .

The values for the transport coefficients used in the spatiotemporal discretization, as explained in [24; 26] and detailed in Appendix C, are not material constants independent of the discretization. Rather, they are *bare* transport values η_0 and χ_0 measured at the length scales of the grid size. We assumed that the bare transport coefficients obey the same scaling with the mass ratio R as predicted by Enskog kinetic theory (C-1)–(C-2). As explained in Appendix C, theoretical arguments and molecular-dynamics results suggest that renormalization effects for viscosity are small and can be safely neglected. We have therefore fixed the viscosity in the hydrodynamic calculations based on the molecular-dynamics estimate $\eta_0 = 2.5$ for the pure fluid with molecular mass $m = 1$ (see Section C.1). However, the bare diffusion coefficient is strongly dependent on the size of the hydrodynamic cells (held fixed in our calculations at $\Delta x = \Delta y = 10$) and on whether filtering (see Appendix B) is used. Therefore, the value of χ_0 needs to be adjusted based on the spatial discretization in such a way as to match the behavior of the molecular-dynamics simulations at length scales much larger than the grid spacing. We describe the exact procedure we used to accomplish this in Section C.2.

The time step in our explicit algorithm is limited by the viscous CFL number $\alpha_v = \nu \Delta t / \Delta x^2 < 1/4$. Since the hydrodynamic calculations are much faster compared to the particle simulations, we used the more expensive RK3 temporal integrator with a relatively small time step $\Delta t = 1.45$, corresponding to $\alpha_v \approx 0.05$ for $c = 1$. For $R = 1$ and $N_c = 64$, we employed a larger time step, $\Delta t = 3.625$ ($\alpha_v \approx 0.125$), with no measurable temporal discretization artifacts for the quantities studied here. We are therefore confident that the discretization errors in this study are dominated by spatial discretization artifacts. In future work, we will explore semi-implicit discretizations and study the effect of taking larger time steps on temporal accuracy. Note that at these parameters for $c = 1$ the isothermal speed of sound is $c_T \approx 5.1$ so that a compressible scheme would require a time step on the order of $\Delta t \sim 1$ (corresponding to advective CFL of about $1/2$). By contrast, the explicit low Mach number algorithm is stable for $\Delta t \lesssim 7.5$. This modest gain is due to the small hydrodynamic cell we use here in order to compare to molecular dynamics. For mesoscopic hydrodynamic cells, the gain in time-step size afforded by the low Mach formulation will be several orders of magnitude larger.

For mass ratio $R = 1$ and $R = 2$, the hydrodynamic calculations were initialized using statistically identical configurations as would be obtained by coarse-graining the initial particle configuration. This implies a sharp, step-like jump in concentration at $y = L/3$ and $y = 2L/3$. Since our spatiotemporal discretization is not strictly monotonicity-preserving, such sharp concentration gradients combined with a small diffusion coefficient χ_0 lead to a large cell Peclet number. This may in turn lead to large deviations of concentration outside of the allowed interval $0 \leq c \leq 1$ for larger mass ratios. Therefore, for $R = 4$, we smoothed the initial condition slightly so that the sharp jump in concentration is spread over a few cells and also employed a nine-point filter for the advection velocity ($w_F = 4$; see [Appendix B](#)). We verified that for $R = 2$ using filtering only affects the large wavenumbers and does not appear to affect the small wavenumbers we study here, provided the bare diffusion coefficient χ_0 is adjusted based on the specific filtering width w_F .

VI-C. Comparison between molecular-dynamics and fluctuating hydrodynamics simulations. In order to compare the molecular-dynamics and the hydrodynamic simulations, we calculated several statistical quantities:

(1) The averages of ρ_1 along the directions perpendicular to the concentration gradient,

$$\rho_1^{(h)}(y) = L^{-1} \int_{x=0}^L \rho_1(x, y) dx, \quad (50)$$

where the integral is discretized as a direct sum over the hydrodynamic cells. Note that it is statistically better to use conserved quantities for such macroscopic averages than to use nonconserved variables such as concentration [37].

(2) The spectrum of the concentration averaged along the direction of the gradient by computing the average

$$c_v(x) = L^{-1} \int_{y=0}^L c(x, y) dy$$

and then taking the discrete Fourier transform. Intuitively, c_v is a measure of the thickness of the red strip in [Figure 3](#) and corresponds closely to what is measured in light-scattering and shadowgraphy experiments [\[61; 12\]](#).

(3) The discrete Fourier spectrum of the y -coordinate of the “center of mass” of concentration along the direction perpendicular to the gradient,

$$h_c(x) = L^{-1} \int_{y=0}^L y \cdot c(x, y) dy.$$

Intuitively, h_c is a measure of the height of the centerline of the red strip in [Figure 3](#).

All quantities were sampled at certain prespecified time points in a number of statistically independent simulations N_s and then means and standard deviations calculated from the N_s data points. For systems of size $N_c = 64$ cells, we used $N_s = 64$ simulations, and for systems of size $N_c = 128$, we used $N_s = 32$ simulations. By far the majority of the computational cost was in performing the HDMD simulations.

Average concentration profiles. Once χ_0 and χ_{eff} were estimated based on simulations of a constant-density ($R = 1$) fluid (see [Section C.2](#)), kinetic theory, i.e., Equations [\(C-1\)](#) and [\(C-2\)](#), can be used to estimate them for different density ratios. In [Figure 9](#) (page 99), we show $\rho_1^{(h)}(y)$ for mass ratio $R = 2$, showing good agreement between HDMD and hydrodynamics especially when fluctuations are accounted for. For $R = 4$, a direct comparison is difficult because the initial condition was slightly different in the hydrodynamic simulations due to the need to smooth the sharp concentration gradient for numerical reasons as explained earlier. This difference strongly affects the shape of $\rho_1^{(h)}(y)$ at early times; however, it does not significantly modify the roughness of the interface, which we study next.

Interface roughness. The most interesting contribution of fluctuations to the diffusive mixing process is the appearance of giant concentration fluctuations in the presence of large concentration gradients as evidenced in the roughness of the interface between the two fluids during the early stages of the mixing in [Figure 3](#). In order to quantify this interface roughness, we used the one-dimensional power spectra

$$S_c(k_x) = \langle \hat{c}_v \hat{c}_v^* \rangle \quad \text{and} \quad S_h(k_x) = \langle \hat{h}_c \hat{h}_c^* \rangle.$$

Note that here we do not correct the discrete wavenumber for the spatial discretization artifacts and continue to use k_x instead of k_\perp .

The temporal evolution of the spectra S_c and S_h is shown in [Figure 4](#) for mass

ratio $R = 1$, and in Figure 5 for mass ratio $R = 4$, for both HDMD and low Mach number fluctuating hydrodynamics (note that deterministic hydrodynamics would give identically zero for any spectral quantity). We observe an excellent agreement between the two, including the correct initial evolution of the interface fluctuations.

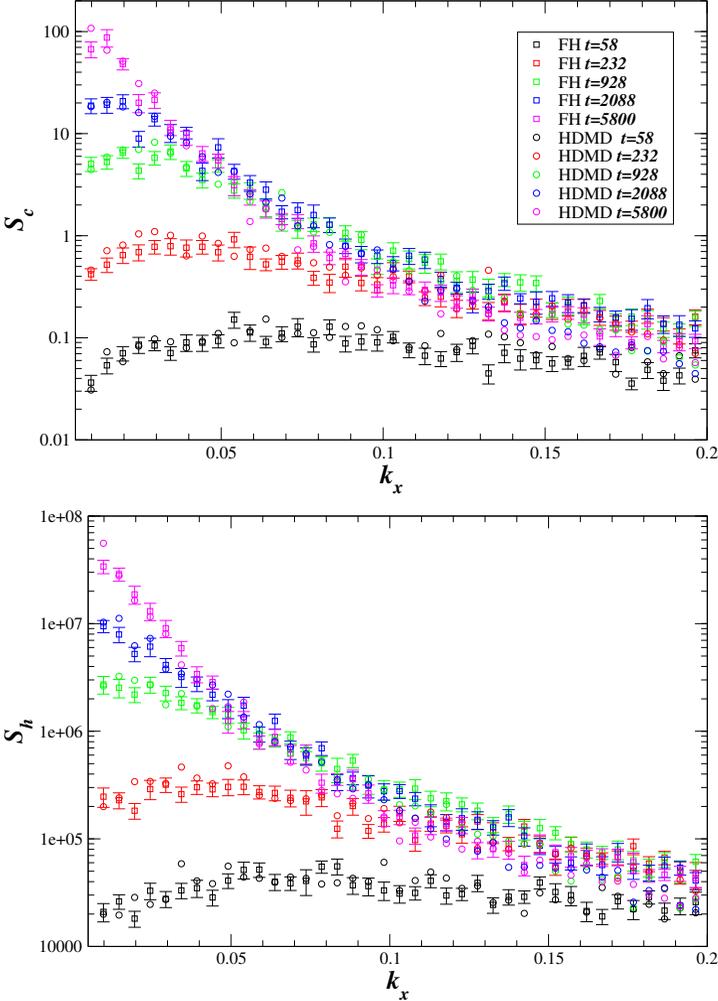


Figure 4. Discrete spatial spectrum of the interface fluctuations for $R = 1$ and $N_c = 128$ (averaged over 32 simulations) at several points in time (drawn with different colors as indicated in the legend) for fluctuating hydrodynamics (FH, squares with error bars) and HDMD (circles, error bars comparable to those for squares). Note that the largest wavenumber supported by the grid is $k_{\max} = \pi/\Delta x \approx 0.314$. The larger wavenumbers are however dominated by spatial truncation errors and the filter employed (if any), and we do not show them here. Top: spectrum $S_c(k_x)$ of the vertically averaged concentration. Bottom: spectrum $S_h(k_x)$ of the position of the vertical “center of mass” of concentration.

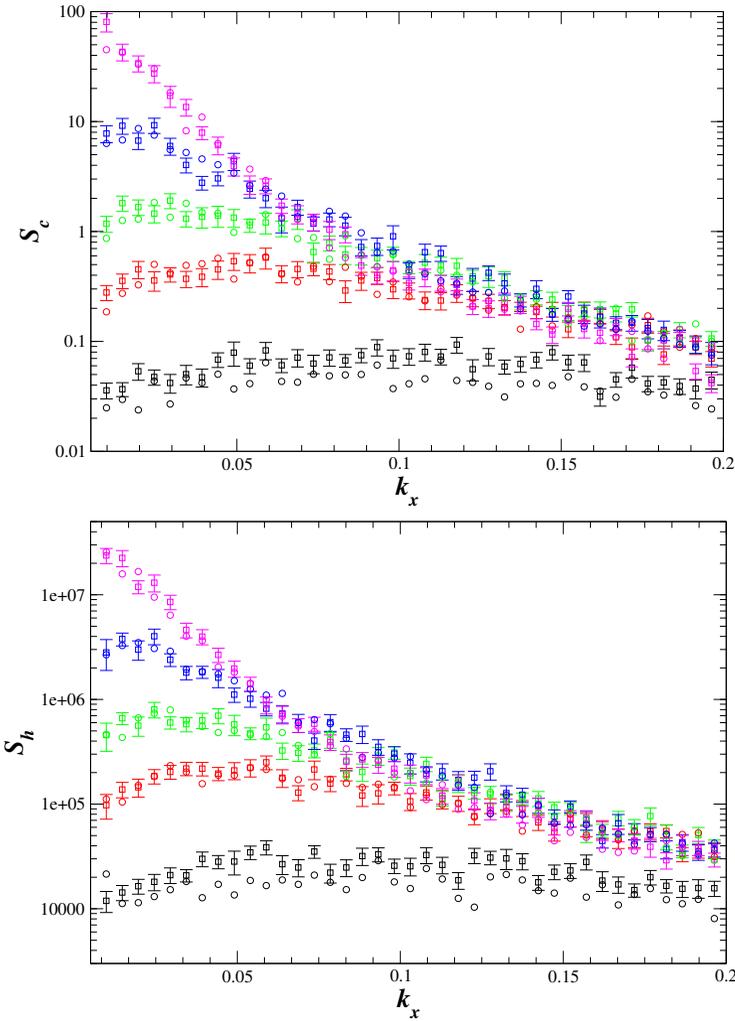


Figure 5. Same as [Figure 4](#) but for density ratio $R = 4$.

Note that, for a finite system, eventually complete mixing will take place and the concentration fluctuations will have to revert to their equilibrium spectrum, which is flat in Fourier space instead of the power-law behavior seen out of equilibrium. In [Figure 6](#), we show results for mixing up to a time $t = 7.42 \cdot 10^5$ (this is 128 times longer than those described above). These long simulations are only feasible for the fluctuating hydrodynamics code and employ a somewhat larger time step $\Delta t = 3.625$. The results clearly show that at late times the spectrum of the fluctuations reverts to the equilibrium one; however, this takes some time even after the mixing is essentially complete. Linearized incompressible fluctuating hydrodynamics [[77](#); [73](#)] predicts that at steady state the spectrum of nonequilibrium concentration

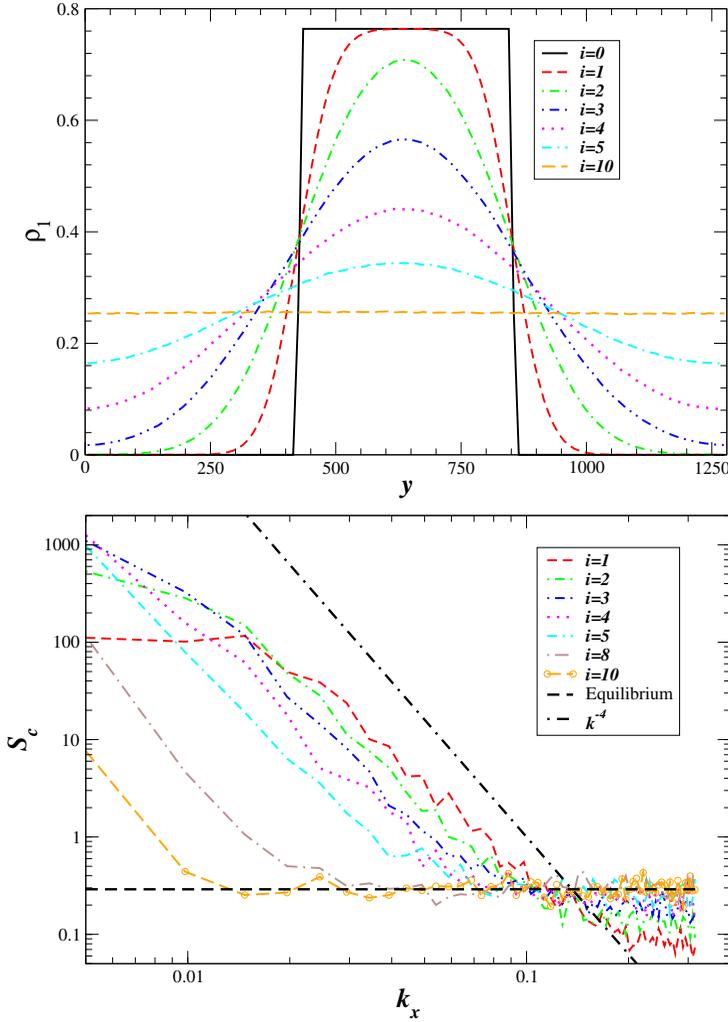


Figure 6. Mixing to a time 128 times longer than previous results with results reported at time intervals $t = 7424 i^2$ for $i = 1, \dots, 10$. These long simulations are only feasible for the fluctuating hydrodynamics code and employ a somewhat larger time step $\Delta t = 3.625$. Top: horizontally averaged ρ_1 as shown for the shorter runs in the top panel of Figure 9. Bottom: the spectrum of interface fluctuations $S_c(k_x)$ as shown in the top panels of Figures 4 and 5 for the shorter runs. The theoretical estimates for the spectrum of equilibrium fluctuations, which is independent of wavenumber, is also shown. We also indicate the theoretical prediction for the power law of the spectrum of steady-state nonequilibrium fluctuations under an applied concentration gradient, $S_c \sim k^{-4}$.

fluctuations is a power law with exponent -4 , $S_c \sim (\nabla c)^2 k^{-4}$. The dynamically evolving spectra in the bottom panel of Figure 6 show approximately such power-law behavior for intermediate times and wavenumbers.

VI-D. Hard-sphere fluctuating hydrodynamics simulations. In order to illustrate the appearance of giant fluctuations in three dimensions, we performed simulations of mixing in a mixture of hard spheres with equal diameters, $\sigma = 1$, and mass ratio $R = 4$. The packing density was chosen to be $\phi = 0.45$, which corresponds to a very dense gas but is still well below the freezing point $\phi_f = 0.49$. For the hydrodynamic simulations, we used cubic cells of dimension $\Delta x = 5$, which corresponds to about 107 particles per hydrodynamic cell on average. In [Figure 7](#), we show results from a single simulation with a grid of size $128 \times 64 \times 128$ cells, which would correspond to about 10^8 particles. This makes molecular-dynamics simulations infeasible and makes hydrodynamic calculations an invaluable tool in studying the mixing process at these mesoscopic scales.

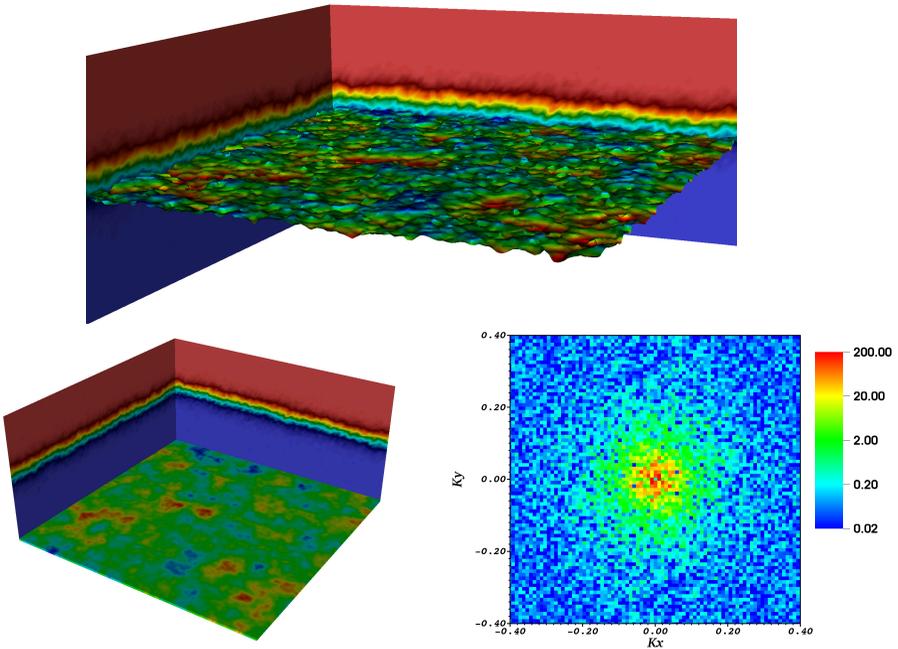


Figure 7. Diffusive mixing in three dimensions similar to that illustrated in [Figure 3](#) for two dimensions. Parameters are based on Enskog kinetic theory for a hard-sphere fluid at packing fraction $\phi = 0.45$, and there is no gravity. The mixing starts with the top half being one species and the bottom half another species with density ratio $R = 4$, and concentration is kept fixed at the top and bottom boundaries while the side boundaries are periodic. A snapshot taken at time $t = 5000$ is shown. Top: the side panes show two-dimensional slices for the concentration c . The approximated contour surface $c = 0.2$ is shown with color based on surface height to illustrate the rough diffusive interface. Bottom left: similar to top panel, but bottom pane shows vertically averaged concentration $c_v(x, z)$, illustrating the giant concentration fluctuations. Bottom right: the Fourier spectrum $S_c(k_x, k_y)$ of c_v . The color axis is logarithmic and clearly shows the appearance of large-scale (small-wavenumber) fluctuations as also seen in [Figure 5](#) in two dimensions.

In the hydrodynamic simulations, we used bare transport coefficient values based on Enskog kinetic theory for the hard-sphere fluid [69]. For the single-component fluid with molecular mass $m = 1$, this theory gives $\eta_0 \approx 2.32$ and $\chi_0 \approx 0.053$, which corresponds to a bare Schmidt number $S_c = \nu_0/\chi_0 \approx 51$. We used the same model dependence of bare transport coefficients on concentration as for hard disks; see Equations (C-1) and (C-2). The time step was set at $\Delta t = 1$ (corresponding to viscous CFL number $\beta = \nu_0 \Delta t / \Delta x^2 \approx 0.1$). In three dimensions, the cell Peclet number is reduced with decreasing Δx , and we did not find it necessary to use any filtering.

Instead of the fully periodic domain used in the two-dimensional hard-disk simulations, here we employ the fixed-concentration boundary conditions (15) and set $c(y = 0; t) = 0$ at the bottom and $c(y = L_y; t) = 1$ at the top boundary. This emulates the sort of “open” or “reservoir” boundaries [17] that mimic conditions in experimental studies of diffusive mixing [12]. The initial condition is a fully phase-separated mixture with $c = 1$ for $y \geq L/2$ and $c = 0$ otherwise. As the mixing process continues, the diffusive interface roughens and giant concentrations appear as illustrated in Figure 7 and also observed experimentally in water-glycerol mixtures in [12]. In three dimensions, however, the diffusive interface roughness is much smaller than in two dimensions, being on the order of only 20 molecular diameters for the snapshot shown in the figure. This illustrates the importance of dimensionality when including thermal fluctuations. In particular, unlike in deterministic fluid dynamics, in fluctuating hydrodynamics, one cannot simply eliminate dimensions from consideration even in simple geometries.

Approximate theory based on the Boussinesq approximation and linearization of the equations of fluctuating hydrodynamics has been developed in [77] and applied in the analysis of experimental results on mixing in a water-glycerol mixture in the presence of gravity [12]. The simulations reported here do not make the sort of approximations necessary in analytical theories and can in principle be used to study the mixing process quantitatively. However, it is important to emphasize that in realistic liquids, such as a water-glycerol mixture, the Schmidt number is on the order of 1000. This makes explicit time-stepping schemes that fully resolve the dynamics of the velocity fluctuations infeasible. In future work, we will consider semi-implicit type-stepping methods that relax the severe time-stepping restrictions present in the explicit schemes considered here.

VII. Conclusions

The behavior of fluids is strongly affected by thermal fluctuations at scales from the microscopic to the macroscopic. Fluctuating hydrodynamics is a powerful coarse-grained model for fluid dynamics at mesoscopic and macroscopic scales at

both a theoretical and a computational level. Theoretical calculations are rather complicated in the presence of realistic spatial inhomogeneities and nontrivial boundary conditions. In numerical simulations, those effects can readily be handled; however, the large separation of time scales between different physical processes poses a fundamental difficulty. Compressible fluctuating hydrodynamics bridges the gap between molecular and hydrodynamic scales. At spatial scales not much larger than molecular, sound and momentum and heat diffusion occur at comparable time scales in both gases and liquids. At mesoscopic and larger length scales, fast pressure fluctuations due to thermally actuated sound waves are much faster than diffusive processes. It is therefore necessary to eliminate sound modes from the compressible equations. In the deterministic context, this is accomplished using low Mach number asymptotic expansion.

For homogeneous simple fluids or mixtures of dynamically identical fluids, the zeroth-order low Mach equations are the well-known incompressible Navier–Stokes equations, in which pressure is a Lagrange multiplier enforcing a divergence-free velocity field. In mixtures of dissimilar fluids, local changes in composition and temperature cause local expansion and contraction of the fluid and thus a nonzero velocity divergence. In this paper, we proposed low Mach number fluctuating equations for isothermal binary mixtures of incompressible fluids with different density or a mixture of low-density gases with different molecular masses. These equations are a straightforward generalization of the widely used incompressible fluctuating Navier–Stokes equations. In the low Mach number equations, the incompressibility constraint $\nabla \cdot \mathbf{v} = 0$ is replaced by $\nabla \cdot \mathbf{v} = -\beta(Dc/Dt)$, which ensures that compositional changes are accompanied by density changes in agreement with the fluid equation of state (EOS) at constant pressure and temperature. This seemingly simple generalization poses many nontrivial analytical and numerical challenges, some of which we addressed in this paper.

At the analytical level, the low Mach number fluctuating equations are different from the incompressible equations because the velocity divergence is directly coupled to the time derivative of the concentration fluctuations. This means that at thermodynamic equilibrium the velocity is not only white in space, a well-known difficulty with the standard equations of fluctuating hydrodynamics, but is also white in time, adding a novel type of difficulty that has not heretofore been recognized. The unphysically fast fluctuations in velocity are caused by the unphysical assumption of infinite separation of time scales between the sound and the diffusive modes. This unphysical assumption also underlies the incompressible fluctuating Navier–Stokes equations; however, in the incompressible limit $\beta \rightarrow 0$, the problem is not apparent because the component of velocity that is white in time disappears. Here we analyzed the low Mach equations at the linearized level and showed that they reproduce the slow diffusive fluctuations in the full compressible equations while eliminating the

fast pressure fluctuations. At the formal level, we suggest that a generalized Hodge decomposition can be used to separate the vortical (solenoidal) modes of velocity as the independently fluctuating variable, coupled with a gauge formulation used to treat the divergence constraint. Such nonlinear analysis is deferred for future research, and here we relied on the fact that the temporal discretization regularizes the short-time dynamics at time scales faster than the time-step size Δt .

At the numerical level, the low Mach number equations pose several distinct challenges. The first challenge is to construct conservative spatial discretizations in which density is advected in a locally conservative manner while still maintaining the equation-of-state constraint relating the local densities and composition. We accomplish this here by using a specially chosen model EOS that is linear yet still rather versatile in practice, and by advecting densities using a velocity that obeys a discrete divergence constraint. We note that, for this simplified case, the system can be modeled using only the concentration to describe the thermodynamic state. However, for more general low Mach number models, maintaining a full thermodynamic representation of the state independent of the constraint leads to more robust numerics. As in incompressible hydrodynamics, enforcing this constraint requires a Poisson pressure solver that dominates the computational cost of the algorithm. A second challenge is to construct temporal integrators that are at least second-order in time. We accomplish this here by formally introducing an unconstrained gauge formulation of the equations while at the same time taking advantage of the gauge degree of freedom to avoid ever explicitly dealing with the gauge variable. The present temporal discretizations are purely explicit and are similar in spirit to an explicit projection method. A third and remaining challenge is to design efficient temporal integrators that handle momentum diffusion, the second-fastest physical process, semi-implicitly. This poses well-known challenges even in the incompressible setting. These challenges were bypassed in recently developed temporal integrators for the incompressible fluctuating Navier–Stokes equations [73] by avoiding the splitting inherent in projection methods. Extending this type of Stokes-system approach to the low Mach equations will be the subject of future research.

One of the principal motivations for developing the low Mach number equations and our numerical implementation was to model recent experiments on the development of giant concentration fluctuations in the presence of sharp concentration gradients. We first studied giant fluctuations in a time-independent or static setting as observed experimentally by inducing a constant concentration gradient via a constant applied temperature gradient. Our simulations show that, under conditions employed in experimental studies of the diffusive mixing of water and glycerol, it is reasonable to employ the Boussinesq approximation. The results also indicate that the constant-transport-coefficient approximation that is commonly used in

theoretical calculations is appropriate if the diffusion coefficient follows a Stokes–Einstein relation, but should be used with caution in general.

We continued our study of giant concentration fluctuations by simulating the temporal evolution of a rough diffusive interface during the diffusive mixing of hard-disk fluids. Comparison between computationally intensive event-driven molecular-dynamics simulations and our hydrodynamic calculations demonstrated that the low Mach number equations of fluctuating hydrodynamics provide an accurate coarse-grained model of fluid mixing. Special care must be exercised, however, in choosing the bare transport coefficients, especially the concentration-diffusion coefficient, as these are renormalized by the fluctuations and can be strongly grid-dependent [23; 24; 26]. Some questions remain about how to define and measure the bare transport coefficients from microscopic simulations, but we show that simply comparing particle and hydrodynamic calculations at large scales is a robust technique.

The strong coupling between velocity fluctuations and diffusive transport means that deterministic models have limited utility at mesoscopic scales and even macroscopic scales in two dimensions. This implies that standard fluorescent techniques for measuring diffusion coefficients, such as fluorescence correlation spectroscopy (FCS) and fluorescence recovery after photobleaching (FRAP) [8], may not in fact be measuring material constants but rather geometry-dependent values [26]. Fluctuating hydrodynamic simulations of typical experimental simulations, however, are still out of reach due to the very large separation of time scales between mass and momentum diffusion. Surpassing this limitation requires the development of a semi-implicit temporal discretization that is stable for large time steps. Furthermore, it is also necessary to develop novel mathematical models and algorithms that are not only stable but also accurate in the presence of such large separation of scales. This is a nontrivial challenge if thermal fluctuations are to be included consistently and will be the subject of future research.

Appendix A: Linearized analysis

As discussed in more depth in [73], there are fundamental mathematical difficulties with the interpretation of the nonlinear equations of fluctuating hydrodynamics due to the roughness of the fluctuating fields. It should be remembered, however, that these equations are coarse-grained models with the coarse-graining length scale set by the size of the hydrodynamic cells used in discretizing the equations [34]. The spatial discretization removes the small length scales from the stochastic forcing and regularizes the equations. It is important to point out, however, that imposing such a small-scale regularization (smoothing) of the stochastic forcing also requires a suitable renormalization of the transport coefficients [4; 23; 26] as we discuss in more detail in Section VI.

As long as there are sufficiently many molecules per hydrodynamic cell, the fluctuations in the spatially discrete hydrodynamic variables will be small and the behavior of the nonlinear equations will closely follow that of the *linearized* equations of fluctuating hydrodynamics [73], which can be given a precise meaning [13]. It is therefore crucial to understand the linearized equations from a theoretical perspective and to analyze the behavior of the numerical schemes in the linearized setting [29].

A1. Compressible equations. Some of the most important quantities predicted by the fluctuating hydrodynamics equations are the equilibrium structure factors (static covariances) of the fluctuating fields. These can be obtained by linearizing the compressible equations (1) around a uniform reference state $\rho = \rho_0 + \delta\rho$, $c = c_0 + \delta c$, $\mathbf{v} = \delta\mathbf{v}$, $P = P_0 + \delta P$, where

$$\delta P = c_T^2 [(\delta\rho) - \beta\rho(\delta c)],$$

and then applying a spatial Fourier transform [61; 29]. Owing to fluctuation-dissipation balance, the static structure factors are independent of the wavevector \mathbf{k} at thermodynamic equilibrium,

$$\begin{aligned} S_{\rho,\rho}(\mathbf{k}) &= \langle (\widehat{\delta\rho})(\widehat{\delta\rho})^* \rangle = \frac{\rho_0 k_B T_0}{c_T^2} + \beta^2 \frac{\rho_0 k_B T_0}{\mu_c}, \\ S_{\mathbf{v},\mathbf{v}}(\mathbf{k}) &= \langle (\widehat{\delta\mathbf{v}})(\widehat{\delta\mathbf{v}})^* \rangle = \rho_0^{-1} k_B T_0 \mathbf{I}, \\ S_{c,c}(\mathbf{k}) &= \langle (\widehat{\delta c})(\widehat{\delta c})^* \rangle = \frac{k_B T_0}{\rho_0 \mu_c}. \end{aligned} \tag{A-1}$$

Note that density fluctuations do not vanish even in the incompressible limit $c_T \rightarrow \infty$ unless $\beta = 0$. While fluctuations in ρ_1 and ρ_2 are uncorrelated, the fluctuations in concentration and density are *correlated* even at equilibrium,

$$S_{c,\rho} = \langle (\widehat{\delta\rho})(\widehat{\delta c})^* \rangle = \beta \frac{k_B T_0}{\mu_c} = \rho_0 \beta S_{c,c}.$$

We will see below that the low Mach equations correctly reproduce the static covariances of density and concentration in the limit $c_T \rightarrow \infty$.

The dynamics of the equilibrium fluctuations can also be studied by applying a Fourier–Laplace transform in time in order to obtain the dynamic structure factors (equilibrium correlation functions) as a function of wavenumber \mathbf{k} and wave frequency ω [61; 29]. It is well-known that the dynamic spectrum of density fluctuations $S_{\rho,\rho}(\mathbf{k}, \omega)$ exhibits three peaks for a given \mathbf{k} : one central Rayleigh peak at small frequencies (slow concentration fluctuations) and two symmetric Brillouin peaks centered around $\omega \approx \pm c_T k$. As the fluid becomes less compressible (i.e., the speed of sound increases), there is an increasing separation of time scales between

the side and central spectral peaks. As we will see below, the low Mach equations reproduce the central peaks in the dynamic structure factors only, eliminating the side peaks and the associated stiff dynamics.

A2. Low Mach equations. We now examine the spatiotemporal correlations of the steady-state fluctuations in the low Mach number equations (8), (9), (11), and (12). In order to model the nonequilibrium setting in which giant concentration fluctuations are observed, we include a constant background concentration gradient in the equations. Note that a density gradient will accompany a concentration gradient, and this can introduce some additional terms in \mathbf{F} depending on how $\rho\chi$ depends on concentration. For simplicity, we assume $\rho\chi$ is a constant so that the diffusive term $\nabla \cdot \mathbf{F}$ in (9) is simply $\rho\chi\nabla^2 c$. We also assume the viscosity η is spatially constant to get the simplified coupled velocity-concentration equations

$$\begin{aligned} D_t \mathbf{v} &= -\rho^{-1} \nabla \pi + \nu \nabla^2 \mathbf{v} + \rho^{-1} (\nabla \cdot \boldsymbol{\Sigma}) + \mathbf{g}, \\ D_t c &= \chi \nabla^2 c + \rho^{-1} (\nabla \cdot \boldsymbol{\Psi}), \\ \nabla \cdot \mathbf{v} &= -\beta D_t c, \end{aligned} \tag{A-2}$$

where $\nu = \eta/\rho$ and $\rho = \rho(c)$ is given by (13).

We linearize the equations (A-2) around a steady state, $c = \bar{c} + \delta c$, $\mathbf{v} = \bar{\mathbf{v}} + \delta \mathbf{v} = \delta \mathbf{v}$, and $\pi = \bar{\pi} + \delta \pi$, where the reference state is in mechanical equilibrium, $\bar{\rho}^{-1} \nabla \bar{\pi} = \mathbf{g}$. We denote the background concentration gradient with $\mathbf{h} = \nabla \bar{c}$. We additionally assume that the reference state varies very weakly on length scales of order of the wavelength and, in particular, that $\bar{\rho}$ and \bar{c} are essentially constant. This allows us to drop the bars from the notation and employ a *quasiperiodic* or weak-gradient approximation [77; 24]. In the linear approximation, the EOS constraint relates density and concentration fluctuations, $\delta \rho = \rho \beta (\delta c)$. The term $\mathbf{v} \cdot \nabla \mathbf{v}$ is second-order in the fluctuations and drops out, but the advective term $\mathbf{v} \cdot \nabla c$ leads to a term $(\delta \mathbf{v}) \cdot \mathbf{h}$ in the concentration equation. The forcing term due to gravity becomes $\rho^{-1} (\delta \rho) \mathbf{g} = \beta (\delta c) \mathbf{g}$. After a spatial Fourier transform, the linearized form of (A-2) becomes a collection of stochastic differential equations, one system of linear additive-noise equations per wavenumber,

$$\partial_t (\widehat{\delta \mathbf{v}}) = -i \rho^{-1} \mathbf{k} (\widehat{\delta \pi}) - \nu k^2 (\widehat{\delta \mathbf{v}}) + i \rho^{-1} \mathbf{k} \cdot \widehat{\boldsymbol{\Sigma}} + \beta \mathbf{g} (\widehat{\delta c}), \tag{A-3}$$

$$\partial_t (\widehat{\delta c}) = -\mathbf{h} \cdot (\widehat{\delta \mathbf{v}}) - \chi k^2 (\widehat{\delta c}) + i \rho^{-1} (\mathbf{k} \cdot \widehat{\boldsymbol{\Psi}}), \tag{A-4}$$

$$\hat{\mathbf{k}} \cdot (\widehat{\delta \mathbf{v}}) = -\beta [i \chi k (\widehat{\delta c}) + \rho^{-1} (\hat{\mathbf{k}} \cdot \widehat{\boldsymbol{\Psi}})]. \tag{A-5}$$

Replacing the right side of (A-5) with zero leads to the incompressible approximation used in [77], corresponding to the Boussinesq approximation of taking the limit $\beta \rightarrow 0$ while keeping the product $\beta \mathbf{g}$ constant.

Equilibrium fluctuations. Let us first compare the dynamics of the equilibrium fluctuations ($\mathbf{h} = \mathbf{0}$) in the low Mach equations with those in the complete compressible equations. For simplicity of notation, we will continue to use the hat symbol to denote the space-time Fourier transform.

In the wavenumber-frequency (\mathbf{k}, ω) Fourier domain, the concentration fluctuations in the absence of a gradient are obtained from (A-4),

$$\widehat{\delta c}(\mathbf{k}, \omega) = \frac{i\rho^{-1}k}{i\omega + \chi k^2}(\hat{\mathbf{k}} \cdot \widehat{\Psi}),$$

which is the same as the compressible equations. The density fluctuations follow the concentration fluctuations, $\widehat{\delta\rho} = \rho\beta\widehat{\delta c}$, and the dynamic structure factor for density shows the same central Rayleigh peak as obtained from the isothermal compressible equations [61],

$$S_{\rho,\rho}(\mathbf{k}, \omega) = \frac{\beta^2 k^2}{\omega^2 + \chi^2 k^4} \langle \widehat{\Psi} \widehat{\Psi}^* \rangle = \beta^2 (\rho \mu_c^{-1} k_B T) \frac{2\chi k^2}{\omega^2 + \chi^2 k^4},$$

where we used (3) for the covariance of $\widehat{\Psi}$. This shows that the low Mach number equations correctly reproduce the slow fluctuations (small ω) in density and concentration while eliminating the side Brillouin peaks associated with the fast isentropic pressure fluctuations.

The fluctuations in velocity, however, are different between the compressible and low Mach number equations. Let us first examine the transverse (solenoidal) component of velocity $\widehat{\delta v}_s = \widehat{\mathcal{P}}\widehat{\delta v}$, where \mathcal{P} is the constant-density orthogonal projection onto the space of divergence-free velocity fields ($\widehat{\mathcal{P}} = \mathbf{I} - k^{-2}(\mathbf{k}\mathbf{k}^*)$ in Fourier space). Applying the projection operator to the velocity equation (A-3) shows that the fluctuations of the solenoidal modes are the same as in the incompressible approximation,

$$\partial_t(\widehat{\delta v}_s) = -vk^2(\widehat{\delta v}_s) + i\rho^{-1}\mathbf{k} \cdot \widehat{\mathcal{P}}\widehat{\Sigma} + \beta\widehat{\mathcal{P}}\mathbf{g}(\widehat{\delta c}).$$

The fluctuations of the compressive velocity component $\widehat{\delta v}_l = \hat{\mathbf{k}} \cdot (\widehat{\delta v})$, on the other hand, are driven by the stochastic mass flux $\widehat{\Psi}$ as seen from (A-5) at thermodynamic equilibrium,

$$\widehat{\delta v}_l = \frac{i\omega\beta\rho^{-1}}{i\omega + \chi k^2}(\hat{\mathbf{k}} \cdot \widehat{\Psi}).$$

The dynamic structure factor (space-time Fourier spectrum) of the longitudinal component

$$S_{v,v}^{(l)} = \langle (\widehat{\delta v}_l)(\widehat{\delta v}_l)^* \rangle \sim \frac{\beta^2 \omega^2}{(\omega^2 + \chi^2 k^4)}$$

does not decay to zero as $\omega \rightarrow \infty$. This indicates that the fluctuations of velocity are not only white in space but also white in time. In the incompressible approximation, $\beta \rightarrow 0$ so that the longitudinal velocity fluctuations vanish and the static spectrum of the velocity fluctuations is equal to the projection operator, $S_{v,v} = \widehat{\mathcal{P}}$ [73]. In the compressible equations, the dynamic structure factor for the longitudinal component of velocity decays to zero as $\omega \rightarrow \infty$ because it has two sound (Brillouin) peaks centered around $\omega \approx c_T k$ in addition to the central diffusive (Rayleigh) peak. The low Mach number equations reproduce the central peak (slow fluctuations) correctly, replacing the side peaks with a flat spectrum for large ω . The origin of this unphysical behavior is the unjustified assumption of infinite separation of time scales between the propagation of sound and the diffusion of mass, momentum, and energy. In reality, the same molecular motion underlies all of these processes and the incompressible or the low Mach number equations cannot be expected to reproduce the correct physical behavior at very short time scales ($\omega \gtrsim c_T k$).

Nonequilibrium fluctuations. If we neglect the term involving $\widehat{\Psi}$ in (A-5) and eliminate the Lagrange multiplier (nonthermodynamic pressure) π using (A-5), we obtain the linearized velocity equation in Fourier space

$$\begin{aligned} \partial_t(\widehat{\delta v}) &= -\nu k^2(\widehat{\delta v}) + i\rho^{-1} \mathbf{k} \cdot \widehat{\mathcal{P}} \widehat{\Sigma} + \beta(\widehat{\delta c}) \widehat{\mathcal{P}} \mathbf{g} \\ &\quad - i\beta\chi[\mathbf{h} \cdot (\widehat{\delta v})] \mathbf{k} + i\beta\chi(\nu - \chi)k^2(\widehat{\delta c}) \mathbf{k}. \end{aligned} \quad (\text{A-6})$$

It is straightforward to obtain the steady-state covariances (static structure factors) in the presence of a concentration gradient from the linearized system of velocity-concentration equations (A-4) and (A-6) [29]. The procedure amounts to solving a linear system for three covariances (velocity-velocity, concentration-concentration, and velocity-concentration). These types of calculations are particularly well-suited for modern computer algebra systems like Maple and can be carried out for arbitrary wavenumber and background concentration gradient. We omit the full solution for brevity.

Experiments measure the steady-state spectrum of concentration fluctuations averaged along the gradient [12; 75], and we will therefore focus on wavenumbers perpendicular to the gradient, $\mathbf{k} \cdot \mathbf{h} = 0$. A straightforward calculation shows that the concentration fluctuations are enhanced as the square of the applied gradient,

$$\begin{aligned} S_{c,c}(\mathbf{k}) &= \langle (\widehat{\delta c})(\widehat{\delta c})^* \rangle \\ &= \frac{k_B T_0}{\rho_0 \mu_c} + \frac{\nu k_B T}{\rho(\nu + \chi) \left[(\nu \chi k_{\perp}^4 + h_{\parallel} g \beta) + \beta^2 (\chi^3 \nu / (\nu + \chi)^2) k_{\perp}^2 h_{\perp}^2 \right]} h_{\parallel}^2, \end{aligned} \quad (\text{A-7})$$

where \perp and \parallel denote the perpendicular and parallel components relative to gravity, respectively. The term in the denominator involving h_{\perp} comes from the low Mach number constraint (11) and is usually negligible since the concentration gradient is

parallel to gravity or $\chi/\nu \ll 1$. Without this term, the result (A-7) is the same result as obtained in [77] and shows that fluctuations at wavenumbers below $k_{\perp}^4 = h_{\parallel} g \beta / (\nu \chi)$ are suppressed by gravity as we study numerically in Section V.

Appendix B: Spatial filtering

In our spatial discretization, we use centered differencing for the advective terms because this leads to a skew-adjoint discretization of advection [54] that maintains discrete fluctuation-dissipation balance in the spatially discretized stochastic equations [29; 18]. It is well-known that centered discretizations of advection do not preserve monotonicity properties of the underlying PDEs in the deterministic setting unlike one-sided (upwind) discretizations. Therefore, our spatiotemporal discretization can lead to unphysical oscillations of the concentration and density in cases where the cell Peclet number $\text{Pe} = \Delta x \|\mathbf{v}\| / \chi$ is large.

In the deterministic setting, Pe can always be decreased by reducing Δx and resolving the fine-scale dissipative features of the flow. However, in the stochastic setting, the magnitude of the fluctuating velocities at equilibrium is

$$\langle (\delta v)^2 \rangle \sim \frac{k_B T}{\rho \Delta V},$$

where ΔV is the volume of the hydrodynamic cell. Therefore, in two dimensions, the characteristic advection velocity magnitude is $\|\mathbf{v}\| \sim \Delta x^{-1}$. This means that in two dimensions Pe is independent of the grid size and reducing Δx cannot fix problems that may arise due to a large cell Peclet number. For some of the simulations reported in Section VI, we have found it necessary to implement a spatial filtering procedure to reduce the magnitude of the fluctuating velocities while preserving their spectrum as well as possible at small wavenumbers.

The filtering procedure consists of applying a local averaging operation to the spatially discretized random fields \mathcal{W} and $\widetilde{\mathcal{W}}$ independently along each Cartesian direction. This local averaging smooths the random forcing and thus reduces the spectrum of the random forcing at larger wavenumbers. The specific filters we use are taken from [46]. For stencil width $w_F = 2$, filtering a discrete field W in one dimension takes the form

$$W_i \leftarrow \frac{5}{8} W_i + \frac{1}{4} (W_{i-1} + W_{i+1}) - \frac{1}{16} (W_{i-2} + W_{i+2}).$$

In Fourier space, for discrete wavenumber $\Delta k = k \Delta x$, this local averaging multiplies the spectrum of W by $\mathcal{F}(\Delta k) = 1 + O(\Delta k^4)$ and therefore maintains the second-order accuracy of the spatial discretization. At the same time, the filtering reduces the variance of the fluctuating fields by about a factor of two in one dimension (a larger factor in two dimensions). The spectrum of the fluctuations can be preserved

even more accurately if a stencil of width $w_F = 4$ is used for the local averaging,

$$W_i \leftarrow \frac{93}{128} W_i + \frac{7}{32} (W_{i-1} + W_{i+1}) - \frac{7}{64} (W_{i-2} + W_{i+2}) + \frac{1}{32} (W_{i-3} + W_{i+3}) - \frac{1}{256} (W_{i-4} + W_{i+4}),$$

giving a sixth-order-accurate filter $\mathcal{F}(\Delta k) = 1 + O(\Delta k^8)$ and a reduction of the variance by about a third in one dimension. In two and three dimensions, the filtering operators are simple tensor products of one-dimensional filtering operators. Note that we only use these filters with periodic boundary conditions. One can, of course, also use Fourier-transform techniques to filter out high-frequency components from the stochastic mass and momentum fluxes.

Appendix C: Extracting transport properties from molecular dynamics

The hydrodynamic simulations described in [Section VI](#) require as input transport coefficients, notably, the shear viscosity η and diffusion coefficient χ , which need to be extracted from the underlying microscopic (molecular) dynamics. This is a very delicate and important step that has not, to our knowledge, been carefully performed in previous studies. In this appendix, we give details about the procedure we developed for this purpose.

C1. Viscosity ν . As discussed in more detail in [\[24; 26\]](#), the transport coefficients in fluctuating hydrodynamics are not universal material constants but rather depend on the spatial scale (degree of coarse-graining) under question. We emphasize that this scale-dependent renormalization is not a molecular scale effect but rather an effect arising out of hydrodynamic fluctuations and persists even at the hydrodynamic scales we are examining here. The best way to define and measure transport coefficients is by examining the dynamics of *equilibrium* fluctuations, specifically, by examining the *dynamic structure factors* of the hydrodynamic fields [\[61\]](#), i.e., the equilibrium averages of the spatiotemporal Fourier spectra of the fluctuating hydrodynamic fields. For a hydrodynamic variable ξ that is transported by a purely diffusive process, the spectrum of the fluctuations at a given wavenumber k and wave frequency ω is expected to be a Lorentzian peak of the form

$$S_x(k, \omega) = \langle \hat{x}(k, \omega) \hat{x}^*(k, \omega) \rangle \sim [\omega^2 + \zeta^2 k^4]^{-1},$$

where in general the diffusion constant $\zeta(k)$ depends on the wavenumber k (wavelength $\lambda = 2\pi/k$). We can therefore estimate the diffusion coefficient χ by fitting a Lorentzian peak to $S_c(k, \omega)$ for different k 's (i.e., $\xi \equiv c$). Similarly, we can estimate the kinematic viscosity $\nu = \eta/\rho$ by fitting a Lorentzian curve to dynamic structure factors for the scaled vorticity, $\xi \equiv k^{-1}(\nabla \times \mathbf{v})_z$.

We performed long equilibrium molecular-dynamics simulations of systems corresponding to a grid of $N_c = 32$ hydrodynamic cells and then calculated the

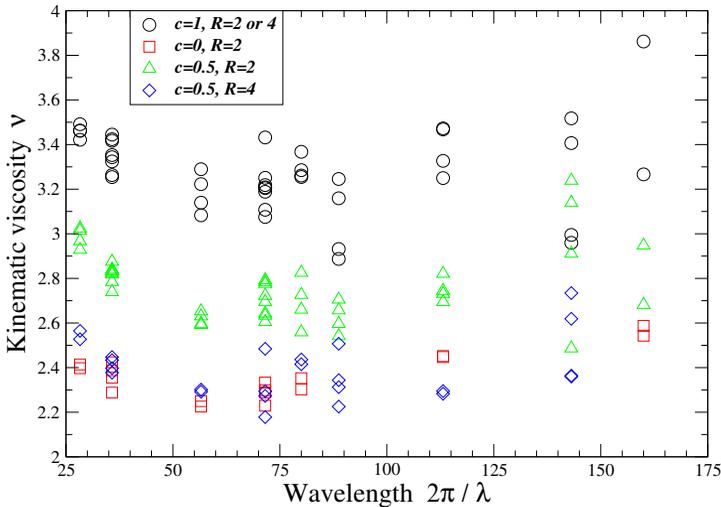


Figure 8. Estimates of the momentum diffusion coefficient (viscosity) $\nu = \eta/\rho$ obtained from the width of the central peak in the dynamic structure factor of vorticity. A collection of 24 distinct discrete wavenumbers k were used and the width of the peaks estimated using a nonlinear least-squares Lorentzian fit.

discrete spatiotemporal Fourier spectrum of the hydrodynamic fields at a collection of discrete wavenumbers k . Since these simulations are at equilibrium, the systems are well-mixed; specifically, the initial configurations were generated by randomly assigning a species label to each particle. We then performed a nonlinear least-squares Lorentzian fit in ω for each k and estimated the width of the Lorentzian peak. The results for the dynamics of the equilibrium vorticity fluctuations are shown in [Figure 8](#). We see that kinematic viscosity is relatively constant for a broad range of wavelengths, consistent with fluctuating hydrodynamics calculations [\[51\]](#) and previous molecular-dynamics simulations [\[38\]](#). For the pure component-one fluid, $c = 1$, with density $\rho \approx 0.764$, the figure shows $\nu \approx 3.3$. We therefore used $\eta_1 \approx 0.764 \cdot 3.3 \approx 2.5$ in all of the hydrodynamic runs reported in [Section VI](#). This is about 20% higher than the prediction of the simple Enskog kinetic theory [\[42\]](#), $\eta \approx 2.06$, and is consistent with the estimates reported in [\[38\]](#). Because the diffusion coefficient is small at the densities we study, more specifically because the Schmidt number $S_c = \nu/\chi$ is larger than 10, we were unable to obtain reliable estimates for $\chi(k)$ from the dynamic structure factor for concentration.

Simple dimensional analysis or kinetic theory shows that $\eta \sim \sqrt{m}$. Since the disks of the two species have equal diameters, the viscosity of the pure second fluid component is

$$\eta_2 = \eta_1 \sqrt{\frac{m_2}{m_1}} = \eta_1 \sqrt{R}. \quad (\text{C-1})$$

There is no simple theory that accurately predicts the concentration dependence of the viscosity of a hard-disk mixture at higher densities [69]. To our knowledge, there is no published Enskog kinetic-theory calculations for hard-disk mixtures in two dimensions even for the simpler case of equal diameters. As an approximation to the true dependence, we employed a simple linear interpolation of the *kinematic* viscosity $\nu(c) = \eta(c)/\rho$ as a function of the mass concentration c between the two known values $\nu_1 = \nu(c = 1) \approx 3.3$ and $\nu_2 = \nu(c = 0) = \nu_1/\sqrt{R}$. The numerical results for mixtures with mass ratios $R = 2$ and $R = 4$ in Figure 8 are consistent with this approximation to within the large error bars. For example, for $c = 1/2$ and $R = 4$, the interpolation gives $\nu = 3 \cdot 3.3/4 \approx 2.5$, which is in reasonable agreement with the numerical estimate.

C2. Diffusion coefficient χ . For the interspecies diffusion coefficient χ , which we emphasize is distinct from the self-diffusion coefficients for particles of either species, Enskog kinetic theory predicts no concentration dependence and a simple scaling with the mass ratio [69],

$$\chi(R) = \chi(R = 1) \sqrt{\frac{1 + R}{2R}}. \quad (\text{C-2})$$

This particular dependence on mass ratio R comes from the fact that the average relative speed between particles of different species is $\sim \sqrt{k_B T/m_R}$, where $m_R = 2m_1 m_2 / (m_1 + m_2)$ is the reduced molecular mass. We have assumed in our hydrodynamic calculations that the diffusion coefficient is independent of concentration and follows (C-2). The only input to the hydrodynamic calculation is the bare self-diffusion coefficient for the pure-component fluid, $\chi_0(R = 1)$. Diffusion is strongly renormalized by thermal fluctuations, and fluctuating hydrodynamics theory and simulations predict a strong dependence of the diffusion coefficient χ on the wavelength [24], consistent with molecular-dynamics results [38].

In order to estimate the appropriate value of the bare diffusion coefficient χ_0 , we numerically solved an inverse problem. Using simple bisection, we looked for the value of χ_0 that leads to best agreement for the average or “macroscopic” diffusion (mixing) between the particle and continuum simulations. Specifically, we calculated the density of the first species $\rho_1^{(h)}(y)$ along the y -direction by averaging ρ_1 in each horizontal row of hydrodynamic cells; see (50). The results for $\rho_1^{(h)}$ for mass ratios $R = 1$ and $R = 4$ are shown in Figure 9 at different points in time for systems of size $N_c = 64$ cells. The figures show the expected sort of diffusive-mixing profile, which is exactly what would be used in experiments to measure diffusion coefficients using fluorescent techniques such as fluorescence recovery after photobleaching (FRAP) [8]. This macroscopic measurement smooths over the fluctuations (roughness) of the diffusive interface and only measures an effective diffusion coefficient at the scale of the domain length L . If deterministic

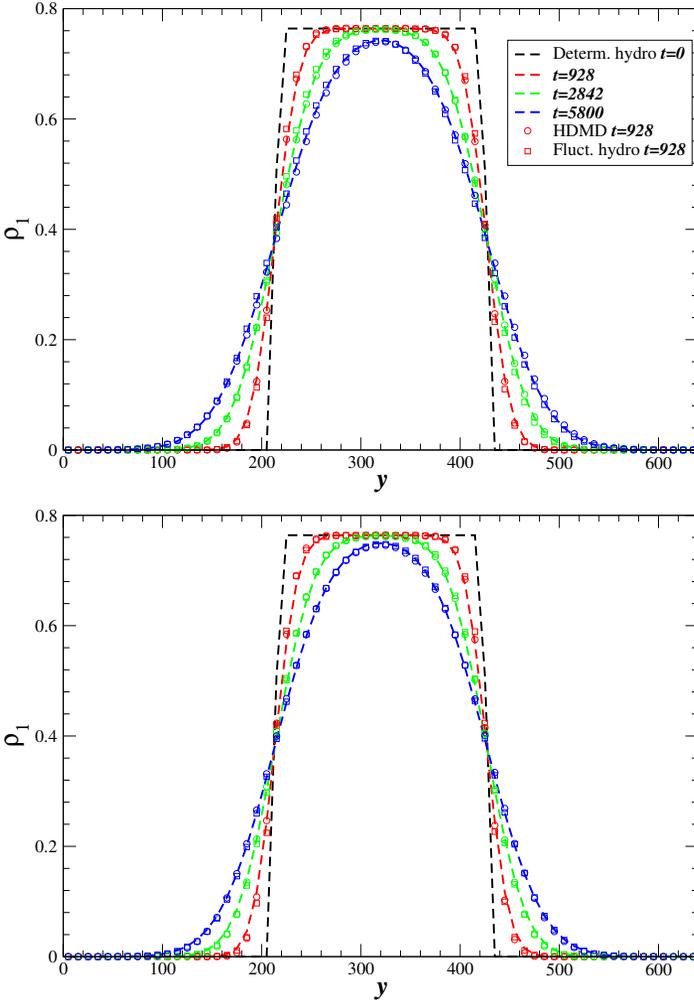


Figure 9. Top: diffusive evolution of the horizontally averaged density $\rho_1^{(h)}(y)$ for a system of size $N_c = 64$ hydrodynamic cells and density ratio $R = 1$ as obtained from HDMD simulations (circles, averaged over 64 runs), deterministic hydrodynamics with $\chi_{\text{eff}} = 0.2$ (dashed lines), and fluctuating hydrodynamics with $\chi_0 = 0.09$ (squares, averaged over 64 runs). Error bars are comparable to the symbol size and not shown for clarity. Bottom: same as the top panel except the density ratio is $R = 2$ and the transport coefficients are adjusted according to (C-1)–(C-2).

hydrodynamics is employed, $\rho_1^{(h)}(y)$ is the solution of a one-dimensional system of equations obtained by simply deleting the stochastic forcing and the x -dependence in the low Mach equations. Instead of solving this system analytically, we employed our spatiotemporal discretization with fluctuations turned off and with an effective diffusion coefficient $\chi = \chi_{\text{eff}}$ that accounts for the renormalization of the diffusion coefficient by the thermal fluctuations.

By matching the profile $\rho_1^{(h)}(y)$ between the HDMD and the fluctuating and deterministic hydrodynamic simulations at mass ratio $R = 1$ and system size $N_c = 64$ cells, we obtained estimates for the bare χ_0 and the renormalized χ_{eff} coefficients (see Figure 9 on the previous page). The best estimate for the bare diffusion coefficient based on this matching in the absence of filtering is $\chi_0 = 0.09 \pm 0.01$. This compares reasonably well to the prediction of Enskog theory [42] of $\chi \approx 0.08$ as well as to the measurement of the self-diffusion coefficient for a periodic system with 169 disks reported in [38], $\chi \approx 0.14$ (recall that a single hydrodynamic cell in our case contains about 76 particles). When a five-point filter is employed, the estimate is $\chi_0(w_F = 2) \approx 0.12$, and when a nine-point filter is employed, $\chi_0(w_F = 4) \approx 0.11$. The estimated renormalized diffusion coefficient is much larger, $\chi_{\text{eff}} \approx 0.20 \pm 0.01$, consistent with a rough estimate based on the simple theory presented in [24],

$$\chi_{\text{eff}} \approx \chi_0 + \frac{k_B T}{4\pi\rho(v + \chi_0)} \ln\left(\frac{N_c}{3}\right) \approx \begin{cases} 0.18 & \text{for } N_c = 64, \\ 0.20 & \text{for } N_c = 128. \end{cases}$$

To within statistical accuracy, we were not able to detect the increase in the estimated diffusion coefficients when using the larger systems of size $N_c = 128$ cells; however, for $N_c = 32$, it was clear that χ_{eff} is reduced.

It is important to emphasize that χ_{eff} is not a material constant but rather depends on the details of the problem in question, in particular, the system geometry and size and boundary conditions [26]. By contrast, χ_0 is a constant for a given spatial discretization, and one can use the same number for different scenarios so long as the hydrodynamic cell size and the filter are kept fixed. Unlike deterministic hydrodynamics, which presents an incomplete picture of diffusion, fluctuating hydrodynamics correctly accounts for the important contribution of the thermal velocity fluctuations and the roughness of the diffusive interface seen in Figure 3.

Acknowledgments

We would like to thank Boyce Griffith and Mingchao Cai for helpful comments. J. Bell, A. Nonaka, and A. Garcia were supported by the DOE Applied Mathematics Program of the DOE Office of Advanced Scientific Computing Research under the U.S. Department of Energy under contract number DE-AC02-05CH11231. A. Donev was supported in part by the National Science Foundation under grant DMS-1115341 and the Office of Science of the U.S. Department of Energy through an Early Career Award (number DE-SC0008271). T. Fai wishes to acknowledge the support of the DOE Computational Science Graduate Fellowship under grant number DE-FG02-97ER25308. Y. Sun was supported by the National Science Foundation under award OCI 1047734.

References

- [1] A. S. Almgren, A. J. Aspden, J. B. Bell, and M. L. Minion, *On the use of higher-order projection methods for incompressible turbulent flow*, SIAM J. Sci. Comput. **35** (2013), no. 1, B25–B42. [MR 3033070](#) [Zbl 1264.76032](#)
- [2] A. S. Almgren, J. B. Bell, P. Colella, L. H. Howell, and M. 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](#)
- [3] P. J. Atzberger, *Stochastic Eulerian Lagrangian methods for fluid-structure interactions with thermal fluctuations*, J. Comput. Phys. **230** (2011), no. 8, 2821–2837. [MR 2012c:74040](#) [Zbl 05909504](#)
- [4] D. Bedeaux and P. Mazur, *Renormalization of the diffusion coefficient in a fluctuating fluid*, I, Physica **73** (1974), no. 3, 431–458. [MR 53 #10059](#)
- [5] 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](#)
- [6] J. B. Bell, A. L. Garcia, and S. A. Williams, *Computational fluctuating fluid dynamics*, ESAIM Math. Model. Numer. Anal. **44** (2010), no. 5, 1085–1105. [MR 2011h:76030](#) [Zbl 05798944](#)
- [7] L. Bocquet and E. Charlaix, *Nanofluidics, from bulk to interfaces*, Chem. Soc. Rev. **39** (2010), no. 3, 1073–1095.
- [8] K. Braeckmans, L. Peeters, N. N. Sanders, S. C. De Smedt, and J. Demeester, *Three-dimensional fluorescence recovery after photobleaching with the confocal scanning laser microscope*, Biophys. J. **85** (2003), no. 4, 2240–2252.
- [9] D. Brogioli, *Giant fluctuations in diffusion in freely-suspended liquid films*, preprint, 2011. [arXiv 1103.4763v2](#)
- [10] M. Cai, A. J. Nonaka, J. B. Bell, B. E. Griffith, and A. Donev, *Efficient variable-coefficient finite-volume Stokes solvers*, preprint, 2013. [arXiv 1308.4605v1](#)
- [11] F. Croccolo, H. Bataller, and F. Scheffold, *A light scattering study of non equilibrium fluctuations in liquid mixtures to measure the soret and mass diffusion coefficient*, J. Chem. Phys. **137** (2012), #234202.
- [12] F. Croccolo, D. Brogioli, A. Vailati, M. Giglio, and D. S. Cannell, *Nondiffusive decay of gradient-driven fluctuations in a free-diffusion process*, Phys. Rev. E **76** (2007), no. 4, 041112.
- [13] G. Da Prato, *Kolmogorov equations for stochastic PDEs*, Birkhäuser, 2004. [MR 2005m:60002](#) [Zbl 1066.60061](#)
- [14] B. Davidovitch, E. Moro, and H. A. Stone, *Spreading of viscous fluid drops on a solid substrate assisted by thermal fluctuations*, Phys. Rev. Lett. **95** (2005), no. 24, 244505.
- [15] M. S. Day and J. B. Bell, *Numerical simulation of laminar reacting flows with complex chemistry*, Combust. Theory Model. **4** (2000), no. 4, 535–556. [Zbl 0970.76065](#)
- [16] R. Delgado-Buscalioni, E. Chacon, and P. Tarazona, *Hydrodynamics of nanoscopic capillary waves*, Phys. Rev. Lett. **101** (2008), no. 10, 106102.
- [17] R. Delgado-Buscalioni, *Tools for multiscale simulation of liquids using open molecular dynamics*, Numerical analysis of multiscale computations (B. Engquist, O. Runborg, and Y.-H. R. Tsai, eds.), Lect. Notes Comput. Sci. Eng., no. 82, Springer, Heidelberg, 2012, pp. 145–166. [MR 3075795](#) [Zbl 1243.82005](#)
- [18] S. Delong, B. E. Griffith, E. Vanden-Eijnden, and A. Donev, *Temporal integrators for fluctuating hydrodynamics*, Phys. Rev. E **87** (2013), no. 3, 033302.

- [19] G. D’Errico, O. Ortona, F. Capuano, and V. Vitagliano, *Diffusion coefficients for the binary system glycerol + water at 25°C: a velocity correlation study*, J. Chem. Engin. Data **49** (2004), no. 6, 1665–1670.
- [20] F. Detcheverry and L. Bocquet, *Thermal fluctuations in nanofluidic transport*, Phys. Rev. Lett. **109** (2012), 024501.
- [21] F. Detcheverry and L. Bocquet, *Thermal fluctuations of hydrodynamic flows in nanochannels*, Phys. Rev. E **88** (2013), no. 1, 012106.
- [22] A. Donev, B. J. Alder, and A. L. Garcia, *Stochastic hard-sphere dynamics for hydrodynamics of nonideal fluids*, Phys. Rev. Lett. **101** (2008), 075902.
- [23] A. Donev, J. B. Bell, A. de la Fuente, and A. L. Garcia, *Diffusive transport by thermal velocity fluctuations*, Phys. Rev. Lett. **106** (2011), no. 20, 204501.
- [24] ———, *Enhancement of diffusive transport by non-equilibrium thermal fluctuations*, J. Stat. Mech. Theor. Exp. (2011), P06014.
- [25] A. Donev, T. G. Fai, and E. Vanden-Eijnden, *Reversible diffusion by thermal fluctuations*, preprint, 2013. [arXiv 1306.3158v3](https://arxiv.org/abs/1306.3158v3)
- [26] ———, *A reversible mesoscopic model of diffusion in liquids: from giant fluctuations to Fick’s law*, J. Stat. Mech. Theor. Exp. (2014), P04004.
- [27] A. Donev, S. Torquato, and F. H. Stillinger, *Neighbor list collision-driven molecular dynamics simulation for nonspherical hard particles, I: Algorithmic details*, J. Comput. Phys. **202** (2005), no. 2, 737–764. [MR 2006e:82093](https://doi.org/10.1016/j.jcp.2006.02.003) [Zbl 1067.82061](https://zbmath.org/journal/1067.82061)
- [28] ———, *Neighbor list collision-driven molecular dynamics simulation for nonspherical hard particles, II: Applications to ellipses and ellipsoids*, J. Comput. Phys. **202** (2005), no. 2, 765–793. [MR 2006e:82094](https://doi.org/10.1016/j.jcp.2006.02.004) [Zbl 1067.82062](https://zbmath.org/journal/1067.82062)
- [29] A. Donev, E. Vanden-Eijnden, A. L. Garcia, and J. B. Bell, *On the accuracy of finite-volume schemes for fluctuating hydrodynamics*, Commun. Appl. Math. Comput. Sci. **5** (2010), no. 2, 149–197. [MR 2012d:65017](https://doi.org/10.1007/s12220-012-9501-7) [Zbl 1277.76089](https://zbmath.org/journal/1277.76089)
- [30] J. R. Dorfman, T. R. Kirkpatrick, and J. V. Sengers, *Generic long-range correlations in molecular fluids*, Annu. Rev. Phys. Chem. **45** (1994), 213–239.
- [31] B. Dünweg and A. J. C. Ladd, *Lattice Boltzmann simulations of soft matter systems*, Advanced computer simulation approaches for soft matter sciences, III (C. Holm and K. Kremer, eds.), Advances in Polymer Science, no. 221, Springer, Berlin, 2009, pp. 89–166.
- [32] D. R. Durran, *Improving the anelastic approximation*, J. Atmos. Sci. **46** (1989), no. 11, 1453–1461.
- [33] W. E and J.-G. Liu, *Gauge method for viscous incompressible flows*, Commun. Math. Sci. **1** (2003), no. 2, 317–332. [MR 2004c:76039](https://doi.org/10.1142/S1746973903000129) [Zbl 1160.76329](https://zbmath.org/journal/1160.76329)
- [34] P. Español, J. G. Anero, and I. Zúñiga, *Microscopic derivation of discrete hydrodynamics*, J. Chem. Phys. **131** (2009), 244117.
- [35] P. Español and I. Zúñiga, *On the definition of discrete hydrodynamic variables*, J. Chem. Phys. **131** (2009), 164106.
- [36] A. L. Garcia, M. Malek Mansour, G. C. Lie, M. Mareschal, and E. Clementi, *Hydrodynamic fluctuations in a dilute gas under shear*, Phys. Rev. A **36** (1987), no. 9, 4348–4355.
- [37] A. L. Garcia, *Estimating hydrodynamic quantities in the presence of microscopic fluctuations*, Commun. Appl. Math. Comput. Sci. **1** (2006), 53–78. [MR 2007c:82078](https://doi.org/10.1007/s12220-006-9001-1) [Zbl 1111.82051](https://zbmath.org/journal/1111.82051)
- [38] R. García-Rojo, S. Luding, and J. J. Brey, *Transport coefficients for dense hard-disk systems*, Phys. Rev. E **74** (2006), no. 6, 061305.

- [39] C. W. Gardiner and M. L. Steyn-Ross, *Adiabatic elimination in stochastic systems, I*, Phys. Rev. A **29** (1984), 2814–2822.
- [40] ———, *Adiabatic elimination in stochastic systems, II*, Phys. Rev. A **29** (1984), 2823–2833.
- [41] ———, *Adiabatic elimination in stochastic systems, III*, Phys. Rev. A **29** (1984), 2834–2844.
- [42] D. M. Gass, *Enskog theory for a rigid disk fluid*, J. Chem. Phys. **54** (1971), no. 5, 1898–1902.
- [43] F. H. Harlow and J. E. Welch, *Numerical calculation of time-dependent viscous incompressible flow of fluids with free surfaces*, Phys. Fluids **8** (1965), 2182–2189.
- [44] Y. Hennequin, D. G. A. L. Aarts, J. H. van der Wiel, G. Wegdam, J. Eggers, H. N. W. Lekkerkerker, and D. Bonn, *Drop formation by thermal fluctuations at an ultralow surface tension*, Phys. Rev. Lett. **97** (2006), no. 24, 244502.
- [45] C. Hijón, P. Español, E. Vanden-Eijnden, and R. Delgado-Buscalioni, *Mori–Zwanzig formalism as a practical computational tool*, Faraday Discuss. **144** (2010), 301–322.
- [46] C. A. Kennedy and M. H. Carpenter, *Several new numerical methods for compressible shear-layer simulations*, Appl. Num. Math. **14** (1994), no. 4, 397–433. MR 95d:76073 Zbl 0804.76062
- [47] S. Klainerman and A. Majda, *Compressible and incompressible fluids*, Comm. Pure Appl. Math. **35** (1982), no. 5, 629–651. MR 84a:35264 Zbl 0478.76091
- [48] L. D. Landau and E. M. Lifshitz, *Fluid mechanics*, Course of Theoretical Physics, no. 6, Pergamon Press, Oxford, 1959.
- [49] ———, *Statistical physics, I*, 3rd ed., Course of Theoretical Physics, no. 5, Butterworth–Heinemann, Oxford, 1980.
- [50] J. Lowengrub and L. Truskinovsky, *Quasi-incompressible Cahn–Hilliard fluids and topological transitions*, Proc. R. Soc. Lond. Ser. A **454** (1998), no. 1978, 2617–2654. MR 2000e:82022 Zbl 0927.76007
- [51] J. Lutsko and J. W. Dufty, *Mode-coupling contributions to the nonlinear shear viscosity*, Phys. Rev. A **32** (1985), 1229–1231.
- [52] A. Majda and J. Sethian, *The derivation and numerical solution of the equations for zero Mach number combustion*, Combust. Sci. Technol. **42** (1985), no. 3–4, 185–205.
- [53] M. Mareschal, M. Malek Mansour, G. Sonnino, and E. Kestemont, *Dynamic structure factor in a nonequilibrium fluid: a molecular-dynamics approach*, Phys. Rev. A **45** (1992), 7180–7183.
- [54] Y. Morinishi, T. S. Lund, O. V. Vasilyev, and P. Moin, *Fully conservative higher order finite difference schemes for incompressible flow*, J. Comput. Phys. **143** (1998), no. 1, 90–124. MR 99a:76100 Zbl 0932.76054
- [55] M. Moseler and U. Landman, *Formation, stability, and breakup of nanojets*, Science **289** (2000), no. 5482, 1165–1169.
- [56] B. Müller, *Low-Mach-number asymptotics of the Navier–Stokes equations*, J. Eng. Math. **34** (1998), no. 1–2, 97–109. MR 99f:76111 Zbl 0924.76095
- [57] A. Naji, P. J. Atzberger, and F. L. H. Brown, *Hybrid elastic and discrete-particle approach to biomembrane dynamics with application to the mobility of curved integral membrane proteins*, Phys. Rev. Lett. **102** (2009), no. 13, 138102.
- [58] F. Nicoud, *Conservative high-order finite-difference schemes for low-Mach number flows*, J. Comput. Phys. **158** (2000), no. 1, 71–97. MR 2000j:76112 Zbl 0973.76068
- [59] H. Noguchi, N. Kikuchi, and G. Gompper, *Particle-based mesoscale hydrodynamic techniques*, Europhys. Lett. **78** (2007), 10005.

- [60] J. M. Ortiz de Zárate and J. V. Sengers, *On the physical origin of long-ranged fluctuations in fluids in thermal nonequilibrium states*, J. Stat. Phys. **115** (2004), no. 5–6, 1341–1359. Zbl 1157.76308
- [61] ———, *Hydrodynamic fluctuations in fluids and fluid mixtures*, Elsevier, Amsterdam, 2006.
- [62] H. C. Öttinger, *Beyond equilibrium thermodynamics*, Wiley, Hoboken, NJ, 2005.
- [63] R. B. Pember, L. H. Howell, J. B. Bell, P. Colella, W. Y. Crutchfield, W. A. Fiveland, and J. P. Jessee, *An adaptive projection method for unsteady, low-Mach number combustion*, Combust. Sci. Technol. **140** (1998), no. 1–6, 123–168.
- [64] C. S. Peskin, G. M. Odell, and G. F. Oster, *Cellular motions and thermal fluctuations: the Brownian ratchet*, Biophys. J. **65** (1993), no. 1, 316–324.
- [65] P. Rauwoens, J. Vierendeels, E. Dick, and B. Merci, *A conservative discrete compatibility-constraint low-Mach pressure-correction algorithm for time-accurate simulations of variable density flows*, J. Comput. Phys. **228** (2009), no. 13, 4714–4744. MR 2010g:65123 Zbl 05581929
- [66] R. G. Rehm and H. R. Baum, *The equations of motion for thermally driven buoyant flows*, J. Res. Natl. Bur. Stand. **83** (1978), 297–308. Zbl 0433.76072
- [67] T. Schneider, N. Botta, K. J. Geratz, and R. Klein, *Extension of finite volume compressible flow solvers to multi-dimensional, variable density zero Mach number flows*, J. Comput. Phys. **155** (1999), no. 2, 248–286. MR 2000g:76081 Zbl 0968.76054
- [68] B. Z. Shang, N. K. Voulgarakis, and J.-W. Chu, *Fluctuating hydrodynamics for multiscale simulation of inhomogeneous fluids: mapping all-atom molecular dynamics to capillary waves*, J. Chem. Phys. **135** (2011), 044111.
- [69] C. M. Silva and H. Liu, *Modelling of transport properties of hard sphere fluids and related systems, and its applications*, Theory and simulation of hard-sphere fluids and related systems (A. Mulero, ed.), Lect. Notes Phys., no. 753, Springer, Berlin, 2008, pp. 383–492. MR 2503513
- [70] M. Skoge, A. Donev, F. H. Stillinger, and S. Torquato, *Packing hyperspheres in high-dimensional Euclidean spaces*, Phys. Rev. E **74** (2006), no. 4, 041127. MR 2007j:82048
- [71] P. T. Sumesh, I. Pagonabarraga, and R. Adhikari, *Lattice-Boltzmann–Langevin simulations of binary mixtures*, Phys. Rev. E **84** (2011), 046709.
- [72] C. J. Takacs, G. Nikolaenko, and D. S. Cannell, *Dynamics of long-wavelength fluctuations in a fluid layer heated from above*, Phys. Rev. Lett. **100** (2008), no. 23, 234502.
- [73] F. B. Usabiaga, J. B. Bell, R. Delgado-Buscalioni, A. Donev, T. G. Fai, B. E. Griffith, and C. S. Peskin, *Staggered schemes for fluctuating hydrodynamics*, Multiscale Model. Simul. **10** (2012), no. 4, 1369–1408. MR 3022043 Zbl 06160065
- [74] A. Vailati, R. Cerbino, S. Mazzoni, M. Giglio, C. J. Takacs, and D. S. Cannell, *Gradient-driven fluctuations in microgravity*, J. Phys. Condens. Matter **24** (2012), no. 28, 284134.
- [75] A. Vailati, R. Cerbino, S. Mazzoni, C. J. Takacs, D. S. Cannell, and M. Giglio, *Fractal fronts of diffusion in microgravity*, Nat. Commun. **2** (2011), 290.
- [76] A. Vailati and M. Giglio, *Giant fluctuations in a free diffusion process*, Nature **390** (1997), 262–265.
- [77] ———, *Nonequilibrium fluctuations in time-dependent diffusion processes*, Phys. Rev. E **58** (1998), no. 4, 4361–4371.
- [78] N. K. Voulgarakis and J.-W. Chu, *Bridging fluctuating hydrodynamics and molecular dynamics simulations of fluids*, J. Chem. Phys. **130** (2009), no. 13, 134111.
- [79] L. Wang and M. Quintard, *Nanofluids of the future*, Advances in transport phenomena 2009 (L. Wang, ed.), Adv. Trans. Phenom., no. 1, Springer, Berlin, 2009, pp. 179–243.

Received November 26, 2013. Revised January 14, 2014.

ALEKSANDAR DONEV: donev@courant.nyu.edu

*Courant Institute of Mathematical Sciences, New York University, 251 Mercer Street,
New York, NY 10012, United States*

ANDY NONAKA: ajnonaka@lbl.gov

*Center for Computational Sciences and Engineering, Lawrence Berkeley National Laboratory,
1 Cyclotron Road, Berkeley, CA 94720, United States*

YIFEI SUN: yifei@cims.nyu.edu

*Courant Institute of Mathematical Sciences, New York University, 251 Mercer Street,
New York, NY 10012, United States*

and

*Leon H. Charney Division of Cardiology, Department of Medicine, New York University School of
Medicine, New York, NY 10016, United States*

THOMAS G. FAI: tfai@cims.nyu.edu

*Courant Institute of Mathematical Sciences, New York University, 251 Mercer Street,
New York, NY 10012, United States*

ALEJANDRO L. GARCIA: alejandro.garcia@sjsu.edu

*Department of Physics and Astronomy, San Jose State University, 1 Washington Square,
San Jose, CA 95192, United States*

JOHN B. BELL: jbbell@lbl.gov

*Center for Computational Sciences and Engineering, Lawrence Berkeley National Laboratory,
1 Cyclotron Road, Berkeley, CA 94720, United States*

HIGH-ORDER METHODS FOR COMPUTING DISTANCES TO IMPLICITLY DEFINED SURFACES

ROBERT I. SAYE

Implicitly embedding a surface as a level set of a scalar function $\phi : \mathbb{R}^d \rightarrow \mathbb{R}$ is a powerful technique for computing and manipulating surface geometry. A variety of applications, e.g., level set methods for tracking evolving interfaces, require accurate approximations of minimum distances to or closest points on implicitly defined surfaces. In this paper, we present an efficient method for calculating high-order approximations of closest points on implicit surfaces, applicable to both structured and unstructured meshes in any number of spatial dimensions. In combination with a high-order approximation of ϕ , the algorithm uses a rapidly converging Newton's method initialised with a guess of the closest point determined by an automatically generated point cloud approximating the surface. In general, the order of accuracy of the algorithm increases with the approximation order of ϕ . We demonstrate orders of accuracy up to six for smooth problems, while nonsmooth problems reliably reduce to their expected order of accuracy. Accompanying this paper is C++ code that can be used to implement the algorithms in a variety of settings.

1. Introduction

A powerful technique for representing curves in two dimensions and surfaces in three dimensions is to define them implicitly, via a fixed level set of a continuous function $\phi : \mathbb{R}^d \rightarrow \mathbb{R}$. Implicit representations of surfaces lead to mathematical and computational advantages in a wide array of problems, such as in methods for computing diffusion and advection processes on embedded surfaces [2; 5; 15], in level set methods [20; 28; 19] for propagating interfaces coupled to computational fluid dynamics [34; 30], and in mesh generation for implicitly defined geometry [21; 24].

Given an implicitly defined surface embedded in \mathbb{R}^d , a common task is to calculate the closest point on the surface to a given query point $x \in \mathbb{R}^d$. For example, level set methods may require the construction of extension velocities [16; 3] or signed distance functions corresponding to a moving interface. In this context, the query

MSC2010: primary 65D99, 68U05, 35R37; secondary 65D17, 65D18, 35R01.

Keywords: implicit surfaces, closest point, level set methods, reinitialisation, redistancing, high-order.

points are the set of grid points of the computational domain, possibly in a narrow band [1]. As another example, embedding techniques for solving partial differential equations (PDEs) defined on curved surfaces work by: (i) embedding the (unknown) solution u in a higher-dimensional function u_{ext} defined on \mathbb{R}^n ; and (ii) deriving a PDE for u_{ext} in such a way that the restriction of u_{ext} to the surface is the solution of the original surface PDE. Techniques using this idea generally use closest points on the surface to define the extension function u_{ext} and its corresponding PDE [2; 29; 5; 15].

In many of these applications, a high-order approximation of the closest point on the surface is required. This is because the resulting distance function or closest point function is used to infer the geometry of the surface, such as when calculating normal vector fields or curvature quantities like the mean curvature or Gaussian curvature. It is often the case that the level set function ϕ is known only at the grid points of a background grid/mesh. It follows that some form of interpolation must be used to define the interface throughout the domain. Even though the values of ϕ at those grid points may have an associated error (e.g., those arising from finite difference approximations or temporal errors in an evolving simulation), it remains necessary to accurately resolve the geometry of the interpolated interface.

In this paper, we develop a general purpose method for computing high-order approximations of closest points on implicitly defined surfaces. The algorithm is largely based on geometry alone and consists of two main stages. First, in an initialisation stage, a level set function ϕ defined on a grid is piecewise approximated by high-order polynomials. Assuming it is the zero level set which defines the surface/interface, these polynomials are then “sampled” by seeding points on their zero level set with sufficient density to form a scattered cloud of points approximating the interface of ϕ . In the second stage of the algorithm, given a query point x , the closest point in the cloud to x is found. This closest point forms an approximation of the actual closest point to x , and this approximation is then improved by using the original polynomial from which it was created together with Newton’s method for solving the minimum-distance optimisation problem. As shown below, this combination of first finding the closest point in the cloud, and then “polishing” it with Newton’s method, leads to highly accurate and robust closest point calculations. By making use of a k-d tree optimised for surfaces, the method is also inexpensive, as finding the closest point in the cloud is relatively cheap, and not many iterations of Newton’s method are required for convergence. Except for the initial stage of forming high-order approximations of ϕ , the method does not rely on any computational grid and can be used to compute closest points at arbitrary locations.

The outline of the paper is as follows. In [Section 2](#), we briefly review previous work on computing distance approximations. In [Section 3](#), the high-order method

is presented, starting with a discussion of high-order polynomial approximations, followed by the sampling procedure and Newton's method. We then discuss some implementation choices, before presenting convergence results and test cases in [Section 4](#). In [Section 5](#) some final remarks are given, including a short description of the C++ code that accompanies this paper. Lastly, in the [Appendix](#), a k-d tree optimised for codimension-one surfaces is presented.

2. Motivation and previous work

Our interest in the minimum-distance/closest-point problem stems from work on level set methods for tracking the interface between two evolving regions, and on Voronoi Implicit Interface Methods [25; 26] for tracking interconnected interfaces with junctions in multiphase physics. Two common tasks in these applications are: (i) calculating extensions of some quantity F defined on the interface, e.g., extension velocities, such that $F_{\text{ext}}(x) = F(\text{cp}(x))$, where $\text{cp}(x)$ is the closest point on the interface to x , and (ii) replacing a function that implicitly defines the interface by the distance function to the interface. In the level set method literature, the latter procedure is a well-known task commonly referred to as *reinitialisation* or *redistancing* and is often performed frequently over the course of a simulation. For example, one reason for maintaining a distance function is related to the treatment of jumps in density and viscosity of a multiphase fluid, or singular forces such as surface tension on a liquid-gas interface, which may require smoothing of Heaviside and Dirac delta functions [8; 34; 30].

Methods for computing distances to implicitly defined surfaces differ in how the geometry of the surface is determined. Approaches include geometrically approximating the surface by explicitly reconstructing it, using root-finding to locate specific points on the surface, employing Eulerian grid-based techniques as in the level set method, or a combination of these methods.

Explicit approaches often use piecewise linear interpolation to find a faceted mesh representation of the interface, from which closest points can be computed by simple geometry [9; 16]. Strain [31] extended this idea to a fast quadtree-based reinitialisation algorithm which enables distances to be efficiently computed on the entire domain based on a mesh that locally adapts to the shape of the interface. Explicit representations also play an important role in the fields of computational geometry and graphics, in which different approximations are applicable; see for instance the review [14].

In the context of level set methods, a common technique for computing the distance function to the interface is to solve a PDE, which is typically done through one of two methods:

- Solve a static boundary-value problem: find ψ such that $\|\nabla\psi\| = 1$, with the requirement that the zero level set of ψ coincides with the zero level set of ϕ .

The solution to this equation is a signed distance function to the interface, i.e., $\psi(x) = \pm \min_{y, \phi(y)=0} \|x - y\|$. A common method for solving this special instance of an Eikonal equation is to apply the Fast Marching Method [27], which solves the general Eikonal equation $\|\nabla\psi\| = F$ where $F = F(x)$ is a general speed function, but apply it to the simpler equation with $F \equiv 1$.

- The second PDE-based method converts the static equation $\|\nabla\psi\| = 1$ into a time-dependent auxiliary PDE whose steady-state solution returns the signed distance function. Here it is generally assumed that ϕ is already close to a distance function, making this an iterative type method. First used in [34], this PDE takes the form

$$\frac{\partial\psi}{\partial\tau} + \text{sign}(\phi)(\|\nabla\psi\| - 1) = 0, \quad \psi(\tau = 0) = \phi.$$

In theory, the zero level set of ψ remains fixed by the process of evolving ψ as $\tau \rightarrow \infty$; in practice, the $\text{sign}(\phi)$ function must be suitably smoothed for the discretised version. These methods rely on high-order ENO and WENO methods to approximate spatial derivatives and high-order Runge–Kutta methods in time. A variety of methods have been developed to improve the accuracy of this approach, see, e.g., [23; 32; 12; 17].

High-order approaches typically compute accurate distances nearby the interface and then employ a PDE-based method to compute distances elsewhere. A notable example is in Chopp’s method [10], which uses a piecewise bicubic (in 2D) or piecewise tricubic (in 3D) interpolant of the level set function that is globally C^1 smooth. For grid points adjacent to the interface, a quasi-Newton method is used to compute closest points on the zero level set of the bicubic/tricubic polynomials, which are then input to a second or third order fast marching method to build the distance function away from the interface. The resulting method is approximately third order in the distance function for smooth interfaces [10]. A similar approach can be used in gradient augmented level set methods [18], where both ϕ and its gradient are defined at each grid point, in which case a type of Hermite interpolation defines a high-order approximation of the interface. This again requires a nonlinear minimisation method to find closest points for query points adjacent to the interface — reinitialisation methods for gradient augmented level set methods include that of [4], which is based in part on Chopp’s quasi-Newton method, and the method of [7], which follows the principles of the fast marching method by using Huygens’ principle and Newton’s method restricted to individual tetrahedrons. Other high order methods include the discontinuous spectral element method of [33], in which a root finding procedure is used to convert the zero level set of a polynomial into a height function, followed by Newton’s method to find closest points on this height function. In this last work, distance

functions were computed with up to sixth order accuracy. Rigorous analyses of errors in reinitialisation methods for finite-element based level set methods have also been performed [22; 13].

In comparison, the high-order method presented in this paper is essentially entirely geometric. The approach extends the ideas of Chopp’s method to arbitrary-order polynomials and replaces the quasi-Newton method with a full Newton’s method that converges much more rapidly when the query point is far away from a curved interface. The method does not rely on any PDE technique, and as such can be used to calculate closest points from arbitrary query points making it suitable for, for example, highly unstructured grids.

3. High-order calculation of closest points

Given a level set function ϕ defined on a computational grid, the high-order closest point algorithm essentially consists of two parts: initialisation, and closest point computation. In the initialisation, a high-order approximation of ϕ is defined on each mesh element containing the interface, followed by a “sampling” procedure which creates a cloud of points approximating the interface. Given a query point $x_q \in \mathbb{R}^d$, a closest point calculation proceeds by finding the closest point in the cloud to x_q , which is then improved by using Newton’s method on the minimum-distance optimisation problem applied to the high-order approximation of the interface.

In order to present the essential ideas of the algorithm, motivated in part by common finite difference-based implementations of the level set method, we mainly consider the case that ϕ is defined on a regular Cartesian grid. The presented techniques can be adapted in a natural way to other cases, such as gradient-augmented level set methods, continuous and discontinuous finite element methods on unstructured grids, etc.; guidelines for doing so are also discussed.

3.1. Piecewise polynomial approximation. Given a level set function defined on a Cartesian grid, many possibilities exist for finding high-order approximations of ϕ between grid points. A natural choice is to find a piecewise polynomial interpolant, such as that used in Chopp [10], in which each grid cell is represented by a bicubic (in 2D) or tricubic (in 3D) polynomial in such a way that the global interpolant is C^1 . However, finding high-order interpolants that are continuous with continuous derivatives can be expensive, since enforcing the continuity requirements requires many degrees of freedom that do not necessarily contribute to the approximation accuracy of the interpolant. For example, a C^1 piecewise tricubic interpolant requires 64 polynomial coefficients per grid cell, but is only third-order accurate; many of the degrees of freedom in the polynomial $\sum_{i,j,k=0}^3 c_{ijk} x^i y^j z^k$ are lost through the enforcing of the C^1 continuity requirement. Compare this to the polynomial corresponding to a third-order accurate Taylor series in three dimensions, which

has only 10 coefficients. For even higher order interpolants that are required to be continuous, possibly with continuous derivatives as well, the situation considerably worsens. Since these polynomials need to be constructed and evaluated many times, it is thus worthwhile to consider an alternative method of approximation.

In regards to the reinitialisation/closest point problem, it is not actually necessary to find a continuous interpolant. All we need is a high-order approximation of the zero level set of ϕ in each grid cell containing the interface. We can achieve this by using polynomials on each grid cell with the minimum number of degrees of freedom necessary for a certain accuracy (as determined by the canonical Taylor series expansion). Note, however, that in doing so, continuity of the zero level set between grid cells may be lost. When the interface is sufficiently smooth, the amount of discontinuity is of the same order as the truncation error of the approximating polynomial and thus will not affect the global approximation order. When the interface is not smooth, such as at the corner of a square, the amount of discontinuity is in general first order in the grid cell size; this cannot be avoided unless specific knowledge of nonsmooth features is incorporated. In either case, provided the polynomials on each grid cell are suitably defined, and the discontinuities of the interface are robustly handled by associated algorithms, the location of the interface defined by the set of polynomials carries the expected order of accuracy.

A straightforward technique for determining these polynomials is to employ a simple least squares method: given a space of polynomials and a stencil of grid points, find the best polynomial in that space which minimises the pointwise interpolation errors in an L^2 norm. Provided the stencil has enough points, this polynomial is uniquely determined. To illustrate, consider a two-dimensional case in which we seek a degree 2 polynomial of the form

$$p(x, y) = c_0 + c_1x + c_2y + c_3x^2 + c_4xy + c_5y^2$$

for determining ϕ in the grid cell $(x_i, x_{i+1}) \times (y_i, y_{i+1})$. We would like it to interpolate the values ϕ_{ij} , $\phi_{i+1,j}$, $\phi_{i,j+1}$ and $\phi_{i+1,j+1}$. However, these 4 conditions are not enough to uniquely determine the 6 coefficients of p , so more grid points are required. We could add exactly two more grid points and this would uniquely determine p , but the resulting stencil would be asymmetric. This may not be a problem when ϕ is smooth, but generally speaking, such asymmetries can lead to stability problems in an evolving interface. Instead, we opt for a symmetric 12-point stencil, as shown in [Figure 1](#). Enumerating the points of the stencil as $\{(x_k, y_k)\}_{k=1}^{12}$, the least squares problem amounts to finding

$$\arg \min_p \sum_{k=1}^{12} |p(x_k, y_k) - \phi(x_k, y_k)|^2,$$

and this can be solved in the usual fashion: form the 12×6 Vandermonde matrix

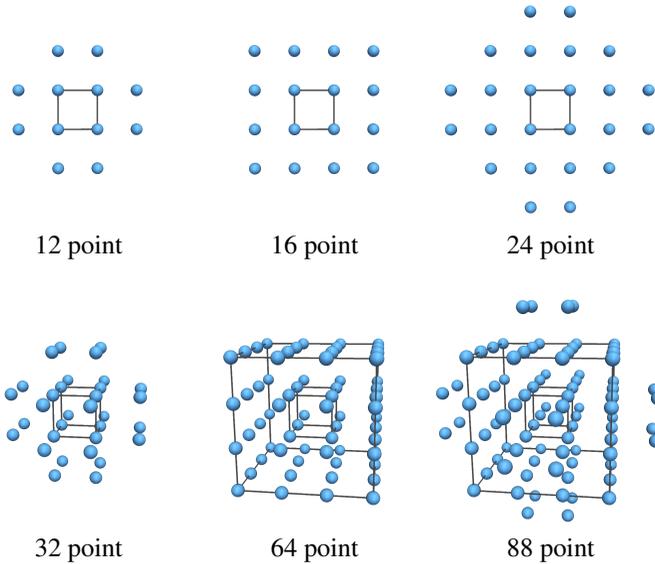


Figure 1. Stencils used to find the polynomials in Table 1. Top: two-dimensional stencils. Bottom: three-dimensional stencils.

A with i -th row $[1, x_i, y_i, x_i^2, x_i y_i, y_i^2]$ and calculate

$$[c_0, \dots, c_5]^T = (A^T A)^{-1} A^T [\phi(x_1, y_1), \dots, \phi(x_{12}, y_{12})]^T.$$

Provided A has full rank, the pseudoinverse $(A^T A)^{-1} A^T$ exists. In this particular example involving a 12-point stencil, this is indeed the case.

This technique easily generalises to other spaces of polynomials and in different dimensions. In each case, a stencil of grid points is designed to be as small as possible such that the corresponding Vandermonde matrix A has full rank. Table 1 and Figure 1 summarise the stencils and polynomials used in this work. Note that the 12-point stencil in the previous example can be used to find both degree 2 polynomials and degree 3 polynomials in 2D. Geometrically this states that the 12-point stencil contains enough information to determine a fourth-order accurate Taylor series approximation. Another point of interest is that the Taylor polynomial of degree 4 in two dimensions, having 15 coefficients, requires a wider stencil of extent 6, despite there being 16 degrees of information in a square 4×4 stencil.¹ This is necessary because the 4×4 stencil does not carry enough information to uniquely determine all of the higher-order terms such as x^4 .

Note that by using a standard reference cell, e.g., $[0, 1]^d$, the pseudoinverses of the Vandermonde matrices can be precomputed. Thus, a polynomial with n

¹The Vandermonde matrix corresponding to the Taylor polynomial of degree 4 in 2D and a square 4×4 stencil has rank 13.

d	Polynomial type	n_c	stencil	p
2	Bicubic, $\sum_{i,j=0}^3 c_{ij}x^i y^j$	16	16	3
2	Taylor degree 2, $c_0 + c_1x + c_2y + c_3x^2 + c_4xy + c_5y^2$	6	12	3
2	Taylor degree 3, $\sum_{ \alpha \leq 3} c_\alpha(x, y)^\alpha$	10	12	4
2	Taylor degree 4, $\sum_{ \alpha \leq 4} c_\alpha(x, y)^\alpha$	15	24	5
2	Taylor degree 5, $\sum_{ \alpha \leq 5} c_\alpha(x, y)^\alpha$	21	24	6
3	Tricubic, $\sum_{i,j,k=0}^3 c_{ijk}x^i y^j z^k$	64	64	3
3	Taylor degree 2, $c_0 + c_1x + c_2y + c_3z + \dots + c_9z^2$	10	32	3
3	Taylor degree 3, $\sum_{ \alpha \leq 3} c_\alpha(x, y, z)^\alpha$	20	32	4
3	Taylor degree 4, $\sum_{ \alpha \leq 4} c_\alpha(x, y, z)^\alpha$	35	88	5
3	Taylor degree 5, $\sum_{ \alpha \leq 5} c_\alpha(x, y, z)^\alpha$	56	88	6

Table 1. d -dimensional polynomials used in this work, indicating the form of the polynomial, number of coefficients n_c , number of points in the stencil (see Figure 1), and the expected order of accuracy p for sufficiently smooth problems.

coefficients can be determined by a stencil of m points by a single matrix-vector multiplication of size $n \times m$. We also note that while the stencils often involve more points than there are coefficients in the polynomials, ultimately it is only the polynomial and its derivatives that need to be evaluated many times.

3.2. Sampling the interface. Using the above piecewise polynomial approximation, we can find a high-order approximation of ϕ in each grid cell. For those grid cells containing the interface², we would like to sample the cell’s polynomial by placing points on its zero level set. It will not be necessary to do this with a high degree of resolution, in fact only a few seed points per grid cell are required.³ Thus, a very simple strategy can be adopted: subdivide each grid cell containing the interface into a 2×2 subgrid (in 2D) or $2 \times 2 \times 2$ subgrid (in 3D), and in each subcell, place a point in the centre. Then, “project” these points onto the zero level set of the polynomial p with a simple Newton-style procedure: given a point $x_0 \in \mathbb{R}^d$, we iterate

$$x_{i+1} = x_i - \frac{p(x_i)\nabla p(x_i)}{\|\nabla p(x_i)\|^2},$$

until a suitable convergence criterion is met. This iterative procedure can be viewed⁴ as moving x_i to its closest point on the zero level set of the linear approximation of p at x_i , given by $p(x_i + \delta) \approx p(x_i) + \delta \cdot \nabla p(x_i)$. Generally, as in Newton’s

²Methods to determine whether a grid cell contains the interface are discussed shortly.

³Generally speaking, the 2×2 subcell division described here is sufficient for most level set applications. If a grid cell contains a polynomial with very high curvature, more points may be required, depending on the application; adaptive approaches are discussed shortly.

⁴It is also the “ δ_1 ” direction used in Chopp’s method [10], as discussed later.

method, this iterative method exhibits second order convergence; in practice it is very quick and reliable. Some remarks:

- A point starts in the centre of its subcell and is projected onto the zero level set of p . As a result, it may move outside its subcell or indeed the parent cell. If the gap between the point and the parent cell is small (i.e., a fraction of Δx), we keep the point — this fits in with the strategy employed later for allowing polynomials from adjacent grid cells to slightly “overlap.” If the point is far away from its subcell, it is discarded.
- It is not necessary for the point to lie exactly on the zero level set of p , as these points only form an initial guess to a full Newton’s method (see [Section 3.3](#)). In practice, a simple convergence criterion suffices, which is to stop iterating when $\|x_{i+1} - x_i\|$ is a small fraction of the subcell size, e.g., 1%; typically only one or two iterations of the above scheme are then necessary.

By doing this for each cell containing the interface, a collection of points is generated. The points are in no particular order and form a cloud of scattered points approximating the interface on the entire domain.

3.3. High-order closest point calculations via Newton’s method. The output of the above sampling stage is a set of points $C = \{x_1, \dots, x_N\} \subset \mathbb{R}^d$ approximating the interface of ϕ . To each point we associate the polynomial p_i from which it was generated, coming from the high-order approximation of ϕ in each grid cell. For the general closest point problem, we are given an arbitrary query point $x_q \in \mathbb{R}^d$ and need to approximate the closest point on the zero level set of ϕ . This is accomplished in two steps:

- Find the closest point in C to x_q . Denote it by x_0 , with associated polynomial p .
- Return the closest point on the zero level set of p restricted to a small domain.

Step (i) is a well-known scattered-data closest-point query problem for which various efficient methods exist, including the use of k-d trees, quadtrees, octrees, etc. In this application, the points lie on a codimension-one surface, and this extra information can be exploited to gain greater efficiency. The [Appendix](#) presents a k-d tree optimised for surfaces that was developed as part of this work. Independent of the implementation details, however, step (i) can be considered to be a black box.

To solve step (ii), a simple Newton’s method for the minimum distance optimisation problem works well. Consider the functional $f : \mathbb{R}^d \times \mathbb{R} \rightarrow \mathbb{R}$ given by

$$f(x, \lambda) = \frac{1}{2} \|x - x_q\|^2 + \lambda p(x),$$

whose gradient and Hessian are

$$\nabla f = \begin{pmatrix} \nabla_x f \\ \partial_\lambda f \end{pmatrix} = \begin{pmatrix} x - x_q + \lambda \nabla p(x) \\ p(x) \end{pmatrix} \quad \text{and} \quad D^2 f = \begin{pmatrix} I + \lambda D^2 p(x) & \nabla p(x) \\ \nabla p(x)^T & 0 \end{pmatrix}.$$

Minimising f amounts to minimising the squared distance from x_q to a point x , with the constraint that x be on the zero level set of p , implemented via a Lagrange multiplier. Generally speaking, this optimisation problem is well-conditioned provided that (a) the gradient of p_i does not vanish near x , and (b) the closest point is unique, i.e., x_q is not located at a shock of the distance function

$$d(x) = \min_{y, p(y)=0} \|x - y\|.$$

Part (a) is a natural regularity assumption in the context of level set methods, while (b) is guaranteed near smooth parts of the distance function d . Naturally, either one of these conditions might fail in practice, but with appropriate safeguards the optimisation problem can be made to be highly robust and efficient. Newton's method for minimising f is as follows: we start at the closest point in C (i.e., x_0) and initialise the Lagrange multiplier at step 0 to be⁵ $\lambda_0 := (x_q - x_0) \cdot \nabla p(x_0) / \|\nabla p(x_0)\|^2$. Let $y = (x, \lambda) \in \mathbb{R}^d \times \mathbb{R}$, with initial value $y_0 := (x_0, \lambda_0)$. We thus iterate

$$y_{k+1} = y_k - (D^2 f(y_k))^{-1} \nabla f(y_k) \quad (1)$$

until convergence to within a suitable tolerance, or else halt the iterations if x travels "too far" from the initial point x_0 . Several remarks are in order:

- To evaluate the Hessian and gradient of f , the Hessian and gradient of the polynomial p are needed. These are straightforward to evaluate for any particular class of polynomial.
- The polynomials were generated from high-order approximations to ϕ on each grid cell. Thus, each polynomial is only valid in a small region surrounding its cell. We therefore only allow the iterates to travel a maximum distance (proportional to Δx) away from the initial starting point x_0 . In addition to preventing iterates from travelling too far (which may occur when the interface is not smooth, e.g., at the corner of a square, as discussed later), this "bounding ball" also provides a straightforward mechanism to allow polynomials from adjacent grid cells to slightly overlap. This is mainly relevant to the case when the closest seed point (i.e., x_0) is close to the boundary of a grid cell, but the true closest point lies slightly in the neighbouring grid cell. It is important to note, however, that the order of accuracy is unaffected by using slightly overlapping polynomials (whether or not the interface is smooth).
- In (1), the Hessian of f , a small $(d+1) \times (d+1)$ square matrix, must be inverted. We can use a simple Gaussian elimination method with partial pivoting, which also indicates when the matrix is nearly singular.⁶ Singularity indicates that

⁵The initial value for the Lagrange multiplier comes from the approximation that $\nabla f \approx 0$ at x_0 .

⁶In this work, the criterion for determining singularity is whether any pivot is less than 10^{-12} in magnitude. This tolerance is based on double precision arithmetic and the property that the length

there are many solutions to the closest point problem for the given x_q . In theory this may certainly occur, such as when finding the closest point to the centre of a circular or spherical interface. However, in practice, noninvertibility of the Hessian almost never arises. Nevertheless, if the Hessian is detected as singular, we employ a different method. This backup mechanism follows that of Chopp's method [10] — the update is replaced by $x_{k+1} = x_k + \delta_1 + \delta_2$, where δ_1 moves x_k to the zero level set of p , and δ_2 moves x_k tangentially to the level set to enforce the orthogonality condition of the closest point. These directions are given by

$$\delta_1 = -\frac{p(x_k)\nabla p(x_k)}{\|\nabla p(x_k)\|^2} \quad \text{and} \quad \delta_2 = \left(I - \frac{\nabla p(x_k)\nabla p(x_k)^T}{\|\nabla p(x_k)\|^2} \right)(x_q - x_k).$$

In fact, Chopp's method can be viewed as a type of gradient descent on f , i.e., $y_{k+1} = y_k - \alpha \nabla f(y_k)$, with a Lagrange multiplier that is suitably reevaluated at the beginning of each iteration.

- As an additional safeguard, no update in the iterative procedure is allowed to move x_k by a large amount (i.e., 50% of the bounding ball radius). This is effectively a simple type of line search common to many optimisation methods, and is generally only relevant when x_q is extremely close to a centre of curvature of p . Once again, this safeguard is rarely invoked in practice.
- To decide when the iterations have converged, we test if $\|x_{k+1} - x_k\| < \epsilon$, where ϵ is a small threshold relating to the accuracy of the polynomial. It is not necessary to converge to machine precision when the polynomial itself is only an approximation of ϕ . We take ϵ to be Δx^p where p is the order of accuracy of the class of polynomials being used — see Table 1.

Newton's method for finding the closest point is summarised in Algorithm 1. Since the initial guess x_0 for the closest point is almost always near the actual closest point, in practice the algorithm converges very quickly; in almost all cases it converges within 2–4 iterations to a sufficient accuracy. Nonconvergence occurs when either (a) the method failed to converge within a fixed number of iterations (20, say), or (b) the iterate left the bounding ball. Although both situations are rare, (a) typically occurs when x_q is very near a shock of the distance function generated by the zero level set of p , while (b) occurs when the closest point is near a nonsmooth part of the interface, e.g., the corner of a square, where the polynomial approximations of the level set functions lead to bumps in the interface (see, e.g., Figures 4 and 5). In the rare case that Newton's method does not successfully converge, but an approximate closest point is nevertheless required as the output of a black-box type algorithm,

scales considered in the test problems are $\mathcal{O}(1)$. Experiments indicated that the overall algorithm is not particularly sensitive to this choice.

```

1:  $\lambda_0 := (x_q - x_0) \cdot \nabla p(x_0) / \|\nabla p(x_0)\|$ .
2: for  $k = 1$  to maximum number of iterations do
3:    $g := (x_k - x_q + \lambda_k \nabla p(x_k), p(x_k))$ .
4:    $H := \begin{pmatrix} I + \lambda_k D^2 p(x_k) & \nabla p(x_k) \\ \nabla p(x_k)^T & 0 \end{pmatrix}$ .
5:   Solve for  $\delta = (\delta_x, \delta_\lambda)$  such that  $H\delta = g$  via Gaussian elimination with partial pivoting.
6:   if succeeded then
7:     if  $\|\delta_x\| > \frac{1}{2}r$  then  $\delta \leftarrow (\frac{1}{2}r / \|\delta_x\|)\delta$ .
8:      $(x_{k+1}, \lambda_{k+1}) := (x_k, \lambda_k) - \delta$ .
9:   else
10:     $\delta_1 := -(p(x_k) / \|\nabla p(x_k)\|^2) \nabla p(x_k)$ .
11:     $\lambda_{k+1} := (x_q - x_k) \cdot \nabla p(x_k) / \|\nabla p(x_k)\|^2$ .
12:     $\delta_2 := x_q - x_k - \lambda_{k+1} \nabla p(x_k)$ .
13:    if  $\|\delta_2\| > \frac{1}{10}r$  then  $\delta_2 \leftarrow (\frac{1}{10}r / \|\delta_2\|)\delta_2$ .
14:     $x_{k+1} := x_k + \delta_1 + \delta_2$ .
15:    if  $\|x_k - x_0\| > r$  then
16:      return did not converge within ball  $B(x_0, r)$ .
17:    else if  $\|x_{k+1} - x_k\| < \epsilon$  then
18:      return converged with solution  $x_{k+1}$ .
19: return did not converge within maximum number of iterations.

```

Algorithm 1. Newton's method for finding the closest point on the zero level set of p given an initial guess x_0 and a bounding ball of radius r .

an approach which suffices in most practical situations is to return the last iterate inside the bounding ball. This approximation carries the same order of accuracy that one may expect when either of the cases (a) or (b) occur, as demonstrated in our convergence tests.

3.4. General algorithm. Combining the above steps, we arrive at the following general approach for computing high-order approximations of closest points on implicit surfaces:

- *Initialisation.*

- (1) For each grid cell/element detected to contain the interface, define or construct a high-order approximation of ϕ on that grid cell/element. For example, when ϕ is defined on a Cartesian grid we can use the least-squares determined polynomials outlined in [Section 3.1](#). Other possibilities may naturally arise given the specific application, for example in a gradient augmented level set method, one can use the associated Hermite interpolants; in a discontinuous or continuous finite element method, ϕ is already naturally defined as a polynomial on the elements of an unstructured mesh.

- (2) For each of these constructed polynomials, sample the zero level set of the polynomial on its domain. In the case that the domain is a rectangular element, e.g., a cell of a Cartesian grid, [Section 3.2](#) described a method based on using a subcell decomposition together with a projection procedure. This approach can naturally be extended to other cases, such as polynomials defined on triangular or tetrahedral elements, by using a similar decomposition method to generate and project points. Guidelines regarding the sampling resolution are provided shortly.
- (3) After sampling the interface on the whole domain, one obtains a cloud of points $C = \{x_1, \dots, x_N\} \subset \mathbb{R}^d$. In the final step of initialisation, a data structure for efficient closest point queries is then created; an example of a k-d tree optimised for surfaces is discussed in the [Appendix](#).

- *Closest point evaluation.* Given an arbitrary point $x_q \in \mathbb{R}^d$, first find the closest point in C to x_q and use this as the initial guess to Newton’s method for determining a high-order closest point, $\text{cp}(x_q)$; see [Section 3.3](#).

[Section 4.4](#) discusses the computational efficiency of the method. As an example application, for the reinitialisation problem in level set methods, we simply replace ϕ with the new signed distance function given by $x \mapsto \text{sign}(\phi(x))\|x - \text{cp}(x)\|$ evaluated at each grid node. We now consider some practical details:

- *Determining which cells contain the interface:* One of the simplest strategies for predicting when a grid cell contains the interface is to examine the sign of ϕ on the vertices of the grid cell — a grid cell is then declared to contain the interface if and only if the signs are not all the same. Clearly, this is not completely reliable. Two typical possibilities include: (a) a closed interface completely contained within a single cell; and (b) an interface which enters and exits the cell on one side/face without crossing any other side/face. In the case that the interface is well-resolved, (a) should not occur (unless subgrid details are to be expected as discussed shortly), but (b) may still occur. Nonetheless, the simple check of examining the signs on grid vertices can still be used to resolve situation (b), essentially because the polynomials from adjacent grid cells are allowed to overlap (as in [Section 3.2](#) and [Section 3.3](#)). For example, a spherical interface defined on a high-resolution grid may partially cross the face of a grid cell without crossing any of its edges. On such a grid cell, the sign check of its vertices will not detect the interface, but the sign check on the neighbouring grid cell will identify the presence of the interface; the polynomial on this grid cell sufficiently approximates the interface in the original cell, due to the overlap allowed in sampling and in Newton’s method.

Depending on the application, such as the subgrid capturing example in [Section 4.5](#), it may be necessary to employ a more sophisticated strategy than to simply check signs of grid vertices. A simple approach is to suppose *every* grid cell contains the

interface; in this case the sampling procedure, while being forced to sample more polynomials, would automatically avoid generating points for cells not containing the interface. Another possibility is to use properties of the polynomial p itself to evaluate bounds of the form

$$\max_{x \in B(x_c, r)} |p(x) - p(x_c)| < C,$$

where C provides a uniform bound on the values of $p(x)$ for x in a ball centred at x_c with a certain grid-dependent radius. If $|p(x_c)| > C$ we can thus prove the polynomial has no zero level set in the corresponding ball.

- *Sampling resolution.* Since the closest point in C to the query point x_q forms an initial approximation to the true closest point, it follows that the sampling resolution of the seed points in C should locally depend on the amount of curvature exhibited by the interface. In other words, on each individual grid cell/mesh element, the length scale characterising the typical separation distance between seed points should be on the same order as the smaller of Δx or the smallest radius of curvature of the interface on that mesh element. In almost all practical applications of the level set method, the interface (once approximated by polynomials) rarely exhibits curvature higher than $\mathcal{O}(1/\Delta x)$. Thus, in [Section 3.2](#), the simple strategy of using a $m \times m$ ($\times m$) subgrid to generate points with $m = 2$ or $m = 3$ typically suffices. In other cases, e.g., a triangular or tetrahedral mesh element, a similar decomposition can be used to sample with similar resolution. For very high-order level set methods, it is possible to capture subgrid effects, in which a single grid cell may contain, for example, an isolated spherical droplet. In these applications, it may be necessary to make m larger. On the other hand, for reasons of efficiency we do not want an excessive number of points in the cloud C since this affects the performance of closest point queries. Ideally, we would like a sampling algorithm that automatically adapts to the curvature exhibited by each polynomial on each mesh element. One possibility for achieving this is to analyse the polynomial and its coefficients to calculate bounds on second derivative information across the entire cell — using these bounds, m could be made automatically adaptive such that $m = 1$ or 2 in smooth parts of the domain, with m larger in regions of high curvature. Though feasible, we will not pursue this idea here or in the accompanying C++ code for the sake of overall simplicity.

- *Overlapping threshold.* The approximate distance between points in the cloud C is also a good measure of how much to allow adjacent grid cell polynomials to overlap. Generally, the polynomials overlap by about $\frac{1}{2}\Delta x$ or less; this is also used as the radius of the bounding ball in Newton’s method.

- *Treatment of boundary conditions.* In the case of a Cartesian grid, we assumed that we could apply stencils at each relevant grid cell to obtain high-order polynomial

approximations. However, this cannot be immediately applied at the boundary of the computational domain, since grid points outside the domain do not exist. Here it is necessary to either (i) find polynomials based on interior data only, and/or (ii) enforce a given boundary condition on ϕ or its interface, such as a zero Neumann boundary condition. One could use, for instance, “ghost layers” in which grid points are defined outside of the domain whose values are based on extrapolation. Since accurate treatment of boundary conditions is highly application-dependent, we will not consider this further here.

- *Narrow banding.* A common implementation of level set methods is to only define ϕ in a small narrow band surrounding the zero level set of ϕ , given by those grid points x for which $\|x - \text{cp}(x)\| < r$, where r is a band radius equal to a fixed number of grid cells [1]. It is straightforward to modify a k-d tree search to consider only points for which the distance to the query point is less than r , returning null if no such points exist; search queries can use this extra information to very efficiently determine the location of the narrow band.
- *Parallelisation.* Parallelising closest point/distance algorithms such as this depends crucially on the intended application. In a level set method, it is often the case that the global domain is subdivided into subdomains, with individual subdomains assigned to individual processors. In this case, and when narrow banding, it is straightforward to parallelise the closest point algorithm: (i) each processor would examine the grid cells in its subdomain and sample the interface; (ii) points that are within a distance r from its subdomain boundary are communicated, together with their associated polynomials, to adjacent processors; (iii) each processor can then proceed completely independently from the rest by building a k-d tree for a slightly larger-sized subdomain. If the application cannot narrow-band, or if a different type of processor decomposition is used, another strategy is likely necessary, such as communicating between processors coarse-grained information about the geometry of the interface.

4. Results

4.1. Convergence tests. For a sufficiently smooth level set function ϕ , each of the methods in Table 1 for approximating ϕ in each grid cell is p -th order accurate. Combined with the closest point algorithm, this leads to an approximation of the closest point function $\text{cp}_h(x)$ and distance function $d_h(x) := \|x - \text{cp}_h(x)\|$. In general, we can expect the distance function approximation to be p -th order accurate, both near and far away from the interface. On the other hand, the closest point approximation may lose up to two orders of accuracy if x is near a “curvature singularity,” e.g., near the centre of a circle. One way to see this is to note that the exact closest point and distance functions satisfy the relation $\text{cp}(x) = x - d(x)\nabla d(x)$

almost everywhere.⁷ Therefore, to recover cp_h from d_h we need to differentiate d_h , thereby incurring an error proportional to $D^2 d(x)h^{p-1}$. The Hessian of a distance function is related to the curvature of its level sets — $D^2 d$ has a singularity behaving like $\|x - x_c\|^{-1}$ near a centre of curvature x_c . It follows that $\text{cp}_h(x)$ may be $(p-2)$ -th order accurate near such a singularity, and some of our results confirm this.

To assess the approximation errors in d_h and cp_h , we consider a variety of smooth and nonsmooth test problems and measure the error locally and globally, in both the $\|\cdot\|_1$ and $\|\cdot\|_\infty$ norms. These tests are performed on a uniform grid such that $\Delta x = \Delta y = \Delta z = h$. More precisely:

- Let S be the set of points in the domain Ω for which the exact closest point function is multivalued, e.g., the centre of a sphere or the inside diagonals of a square. In a numerical setting, it would be overly complicated to request the closest point algorithm to return all possible solutions when the query point is in S . Hence, to simplify the convergence analysis, we will ignore grid points that are in S or situated very close to S , as follows. Let S_h be the set of grid points whose minimum distance to S is less than δ ; in our results, the threshold has been set to half a grid cell, $\delta = \frac{1}{2}h$. Grid points in S_h are ignored *only* when measuring errors in the closest point function; they are still considered in the case of the distance function.
- Local errors are measured in a narrow band of radius 8 grid cells: let N_h be the set of grid points x for which $d_h(x) < 8h$. Define

$$\|d - d_h\|_{1, N_h} = \frac{1}{|N_h|} \sum_{x \in N_h} |d(x) - d_h(x)|,$$

$$\|\text{cp} - \text{cp}_h\|_{1, N_h \setminus S_h} = \frac{1}{|N_h \setminus S_h|} \sum_{x \in N_h \setminus S_h} \|\text{cp}(x) - \text{cp}_h(x)\|,$$

and

$$\|d - d_h\|_{\infty, N_h} = \max_{x \in N_h} |d(x) - d_h(x)|,$$

$$\|\text{cp} - \text{cp}_h\|_{\infty, N_h \setminus S_h} = \max_{x \in N_h \setminus S_h} \|\text{cp}(x) - \text{cp}_h(x)\|.$$

- Global errors are measured across the entire computational domain Ω , with the same definitions of the norm, except that N_h is replaced with the set of all grid points; grid points in S_h are ignored only when measuring errors in the closest point function.

A variety of test problems have been analysed, including a circle, sphere, ellipse, ellipsoid, square, cube, rectangle with rounded ends and a cylinder with rounded

⁷The relation is not defined at shocks where d is not differentiable — equivalently, where there is more than one closest point to x .

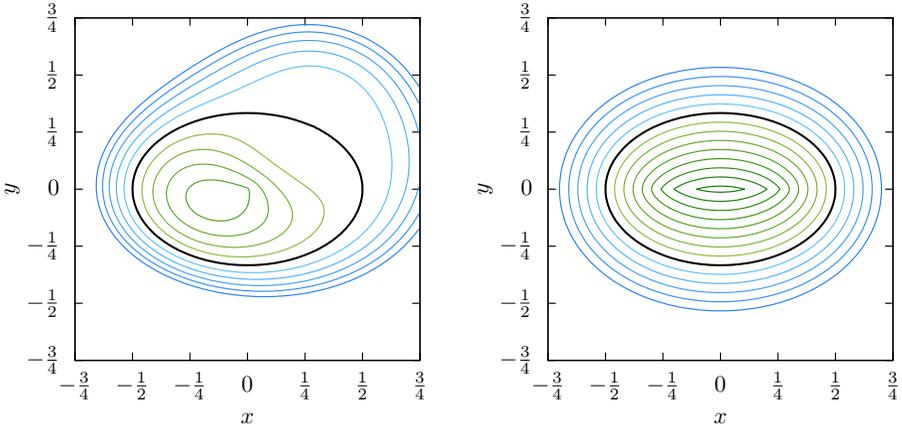


Figure 2. Two-dimensional test case corresponding to an ellipse with semimajor axis $\frac{1}{2}$ and semiminor axis $\frac{1}{3}$. Left: contours of the initial level set function ϕ given by (2), with the zero level set indicated by a thick line. Right: reinitialised signed distance function.

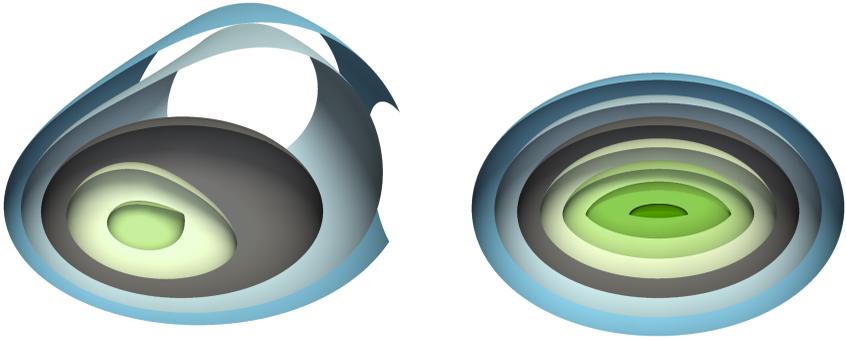


Figure 3. Three-dimensional test case corresponding to an ellipsoid with semiprincipal axes $\frac{1}{2}$, $\frac{1}{3}$ and $\frac{1}{2}$. Left: contours of the initial level set function ϕ given by (3), with the zero level set shown in dark grey. Right: reinitialised signed distance function. In both figures, the contours have been sliced by a plane in order to reveal the inner structure.

ends. Here we show results for the most instructive case, that of an ellipse in 2D and ellipsoid in 3D, followed by a summary of the results of the other tests. In all cases, the domain is $\Omega = [-\frac{3}{4}, \frac{3}{4}]^d$ and the level set function ϕ is defined on a uniform Cartesian $n \times n$ grid (in 2D) or $n \times n \times n$ grid (in 3D). For the ellipse, ϕ is evaluated at grid points via

$$\phi(x, y) = (1 - \exp(-(x - 0.3)^2 - (y - 0.3)^2))(\sqrt{4x^2 + 9y^2} - 1), \quad (2)$$

and for the ellipsoid,

$$\phi(x, y, z) = (1 - \exp(-(x - 0.3)^2 - (y - 0.3)^2))(\sqrt{4x^2 + 9y^2 + 4z^2} - 1). \quad (3)$$

These functions were designed to exhibit large changes in the norm of the gradient near the zero level set. Figures 2 and 3 show contours of ϕ as well as their reinitialised counterparts. We can see that the resulting distance function is not smooth: in the case of the ellipse, $d(x)$ is not smooth on the segment

$$\{(x, y) : |x| \leq \frac{1}{2} - \frac{2}{9}, y = 0\},$$

and has curvature singularities at

$$(x, y) = (\pm(\frac{1}{2} - \frac{2}{9}), 0).$$

A similar disc of nonsmoothness exists for the ellipsoid. Thus we expect to see differing rates of convergence depending on the local and global metrics. Tables 2 and 3 presents the convergence results⁸ for all the polynomials of Table 1 in both 2D and 3D. For each type of polynomial, the convergence rate is estimated by taking ratios of errors between different grid sizes and are indicated by bold numbers in the two tables. The results can be summarised as follows:

- The bicubic and tricubic polynomials (which recall are designed to find a C^1 interpolant of the level set function) are locally third order accurate, for both the distance function and closest point function. Globally, the distance function is third order; however, the closest point function is approximately first order.
- For each of the Taylor polynomials of degree d_T , letting $p = d_T + 1$ (as in Table 1), both the distance and closest point functions are locally p -th order accurate. Globally, the distance function is also p -th order accurate. The closest point function is globally $(p-1)$ -th order accurate in the L^1 norm, and is between $(p-2)$ -th and $(p-1)$ -th order accurate in the maximum norm.

Thus, we obtain the optimal convergence rate in both the distance function and closest point function, depending on proximity to the interface or curvature singularities. To be more precise, for a sufficiently smooth interface there are three zones of convergence: (i) if x is such that $d(x) = \mathcal{O}(h)$ (as in a narrow band), then the closest point approximation is p -th order accurate; (ii) if x is a fixed distance away from the interface (i.e., independent from h) and is not located at a curvature singularity, then the closest point approximation is $(p-1)$ -th order accurate; and (iii) if x has distance $\mathcal{O}(h)$ from a curvature singularity, then the closest point approximation is $(p-2)$ -th order accurate. The distance function approximation d_h is p -th order accurate in all three zones (for a sufficiently smooth interface).

⁸In order to measure the error, we need the exact closest point function for an ellipse and ellipsoid. This was implemented by using Newton's method, similar to that developed in Section 3.3, applied to the polynomials $4x^2 + 9y^2 - 1$ in 2D and $4x^2 + 9y^2 + 4z^2 - 1$ in 3D, with enough iterations to compute the closest point to machine precision accuracy.

	d	n	Distance error		Closest point error		
			$\ \cdot\ _1$	$\ \cdot\ _\infty$	$\ \cdot\ _1$	$\ \cdot\ _\infty$	
Bicubic	2	64	3.66×10^{-5}	1.14×10^{-3}	6.21×10^{-4}	2.81×10^{-2}	
		128	6.90×10^{-6}	2.4 8.95×10^{-4}	0.3 2.07×10^{-4}	1.6 2.10×10^{-2}	0.4
		256	5.75×10^{-7}	3.6 1.23×10^{-4}	2.9 5.47×10^{-5}	1.9 1.04×10^{-2}	1.0
		512	6.95×10^{-8}	3.0 2.25×10^{-5}	2.5 1.50×10^{-5}	1.9 4.04×10^{-3}	1.4
Tricubic	3	64	2.85×10^{-5}	7.85×10^{-3}	5.47×10^{-4}	4.38×10^{-2}	
		128	3.77×10^{-6}	2.9 1.56×10^{-3}	2.3 1.58×10^{-4}	1.8 2.58×10^{-2}	0.8
		256	3.54×10^{-7}	3.4 2.59×10^{-4}	2.6 4.10×10^{-5}	1.9 1.18×10^{-2}	1.1
		512	3.75×10^{-8}	3.2 3.55×10^{-5}	2.9 9.89×10^{-6}	2.1 5.37×10^{-3}	1.1
Taylor degree 2	2	64	5.03×10^{-4}	1.20×10^{-2}	2.11×10^{-3}	6.88×10^{-2}	
		128	5.05×10^{-5}	3.3 1.32×10^{-3}	3.2 3.65×10^{-4}	2.5 2.09×10^{-2}	1.7
		256	5.95×10^{-6}	3.1 2.14×10^{-4}	2.6 8.86×10^{-5}	2.0 6.74×10^{-3}	1.6
		512	6.74×10^{-7}	3.1 3.11×10^{-5}	2.8 2.19×10^{-5}	2.0 2.64×10^{-3}	1.3
	3	64	2.20×10^{-4}	1.23×10^{-2}	1.32×10^{-3}	6.77×10^{-2}	
		128	2.25×10^{-5}	3.3 1.55×10^{-3}	3.0 2.84×10^{-4}	2.2 2.25×10^{-2}	1.6
		256	2.56×10^{-6}	3.1 2.24×10^{-4}	2.8 7.14×10^{-5}	2.0 7.10×10^{-3}	1.7
		512	2.91×10^{-7}	3.1 2.99×10^{-5}	2.9 1.80×10^{-5}	2.0 3.05×10^{-3}	1.2
Taylor degree 3	2	64	7.24×10^{-6}	4.31×10^{-4}	1.05×10^{-4}	1.84×10^{-2}	
		128	4.19×10^{-7}	4.1 1.79×10^{-5}	4.6 1.46×10^{-5}	2.8 2.00×10^{-3}	3.2
		256	2.52×10^{-8}	4.1 9.30×10^{-7}	4.3 1.93×10^{-6}	2.9 2.96×10^{-4}	2.8
		512	1.61×10^{-9}	4.0 5.94×10^{-8}	4.0 1.34×10^{-7}	3.0 3.88×10^{-5}	2.9
	3	64	4.48×10^{-6}	2.54×10^{-4}	7.28×10^{-5}	1.23×10^{-2}	
		128	2.69×10^{-7}	4.1 1.82×10^{-5}	3.8 9.58×10^{-6}	2.9 2.08×10^{-3}	2.6
		256	1.66×10^{-8}	4.0 1.05×10^{-6}	4.1 1.22×10^{-6}	3.0 3.08×10^{-4}	2.8
		512	1.03×10^{-9}	4.0 6.73×10^{-8}	4.0 1.54×10^{-7}	3.0 4.23×10^{-5}	2.9
Taylor degree 4	2	64	1.93×10^{-6}	8.14×10^{-5}	1.75×10^{-5}	1.12×10^{-3}	
		128	5.68×10^{-8}	5.1 2.53×10^{-6}	5.0 1.03×10^{-6}	4.1 9.43×10^{-5}	3.6
		256	1.80×10^{-9}	5.0 8.64×10^{-8}	4.9 6.32×10^{-8}	4.0 6.57×10^{-6}	3.8
		512	5.65×10^{-11}	5.0 2.89×10^{-9}	4.9 3.94×10^{-9}	4.0 4.92×10^{-7}	3.7
	3	64	9.97×10^{-7}	1.23×10^{-4}	1.10×10^{-5}	1.32×10^{-3}	
		128	3.14×10^{-8}	5.0 2.79×10^{-6}	5.5 6.69×10^{-7}	4.0 1.00×10^{-4}	3.7
		256	9.88×10^{-10}	5.0 1.01×10^{-7}	4.8 4.15×10^{-8}	4.0 6.34×10^{-6}	4.0
		512	3.09×10^{-11}	5.0 3.32×10^{-9}	4.9 2.59×10^{-9}	4.0 4.72×10^{-7}	3.7
Taylor degree 5	2	64	5.22×10^{-8}	1.52×10^{-6}	6.41×10^{-7}	4.84×10^{-5}	
		128	7.39×10^{-10}	6.1 3.01×10^{-8}	5.7 1.79×10^{-8}	5.2 1.08×10^{-6}	5.5
		256	1.18×10^{-11}	6.0 4.67×10^{-10}	6.0 5.28×10^{-10}	5.1 5.64×10^{-8}	4.3
		512	1.95×10^{-13}	5.9 7.31×10^{-12}	6.0 1.67×10^{-11}	5.0 2.31×10^{-9}	4.6
	3	64	4.64×10^{-8}	2.78×10^{-6}	6.25×10^{-7}	4.15×10^{-5}	
		128	7.22×10^{-10}	6.0 5.33×10^{-8}	5.7 1.86×10^{-8}	5.1 1.65×10^{-6}	4.6
		256	1.12×10^{-11}	6.0 8.21×10^{-10}	6.0 5.75×10^{-10}	5.0 4.86×10^{-8}	5.1
		512	1.75×10^{-13}	6.0 1.27×10^{-11}	6.0 1.79×10^{-11}	5.0 1.94×10^{-9}	4.6

Table 2. Convergence results (*global error*) for the ellipse (dimension $d = 2$) and ellipsoid ($d = 3$) for several polynomial classes: the bicubic (in 2D), tricubic (in 3D) and the Taylor polynomials in Table 1. The left pair measures the error in the distance function and the second pair the error in the closest point function. For each polynomial type, the error is indicated for a grid of size $n \times n$ in 2D and $n \times n \times n$ in 3D. Ratios between errors on successive grids yield the convergence rates in bold.

	d	n	Distance error		Closest point error	
			$\ \cdot\ _1$	$\ \cdot\ _\infty$	$\ \cdot\ _1$	$\ \cdot\ _\infty$
Bicubic	2	64	2.98×10^{-5}	1.14×10^{-3}	3.47×10^{-4}	1.76×10^{-2}
		128	4.31×10^{-6}	2.8 6.81×10^{-4}	0.7 5.95×10^{-5}	2.5 8.25×10^{-3}
		256	4.97×10^{-7}	3.1 7.22×10^{-5}	3.2 7.47×10^{-6}	3.0 1.69×10^{-3}
		512	5.89×10^{-8}	3.1 7.62×10^{-6}	3.2 8.09×10^{-7}	3.2 1.54×10^{-4}
Tricubic	3	64	2.25×10^{-5}	7.85×10^{-3}	2.67×10^{-4}	3.88×10^{-2}
		128	2.26×10^{-6}	3.3 1.50×10^{-3}	2.4 3.62×10^{-5}	2.9 1.30×10^{-2}
		256	2.42×10^{-7}	3.2 1.35×10^{-4}	3.5 4.29×10^{-6}	3.1 4.27×10^{-3}
		512	2.97×10^{-8}	3.0 1.37×10^{-5}	3.3 5.27×10^{-7}	3.0 2.73×10^{-4}
Taylor degree 2	2	64	3.24×10^{-4}	1.19×10^{-2}	9.08×10^{-4}	4.04×10^{-2}
		128	3.64×10^{-5}	3.2 1.31×10^{-3}	3.2 1.00×10^{-4}	3.2 5.56×10^{-3}
		256	4.29×10^{-6}	3.1 2.13×10^{-4}	2.6 1.15×10^{-5}	3.1 1.08×10^{-3}
		512	5.18×10^{-7}	3.0 3.09×10^{-5}	2.8 1.30×10^{-6}	3.1 1.09×10^{-4}
Taylor degree 3	3	64	1.58×10^{-4}	1.23×10^{-2}	6.41×10^{-4}	3.97×10^{-2}
		128	1.71×10^{-5}	3.2 1.53×10^{-3}	3.0 7.91×10^{-5}	3.0 6.96×10^{-3}
		256	1.96×10^{-6}	3.1 2.18×10^{-4}	2.8 9.67×10^{-6}	3.0 1.08×10^{-3}
		512	2.37×10^{-7}	3.0 2.84×10^{-5}	2.9 1.18×10^{-6}	3.0 1.30×10^{-4}
Taylor degree 3	2	64	5.84×10^{-6}	2.04×10^{-4}	5.56×10^{-5}	9.14×10^{-3}
		128	3.48×10^{-7}	4.1 1.42×10^{-5}	3.8 3.68×10^{-6}	3.9 3.62×10^{-4}
		256	2.20×10^{-8}	4.0 9.00×10^{-7}	4.0 2.24×10^{-7}	4.0 2.58×10^{-5}
		512	1.40×10^{-9}	4.0 5.94×10^{-8}	3.9 1.34×10^{-8}	4.1 1.34×10^{-6}
Taylor degree 3	3	64	3.83×10^{-6}	2.25×10^{-4}	3.55×10^{-5}	9.04×10^{-3}
		128	2.36×10^{-7}	4.0 1.54×10^{-5}	3.9 2.28×10^{-6}	4.0 3.62×10^{-4}
		256	1.48×10^{-8}	4.0 9.59×10^{-7}	4.0 1.44×10^{-7}	4.0 2.57×10^{-5}
		512	9.25×10^{-10}	4.0 6.33×10^{-8}	3.9 8.99×10^{-9}	4.0 1.48×10^{-6}
Taylor degree 4	2	64	1.67×10^{-6}	8.14×10^{-5}	8.95×10^{-6}	6.96×10^{-4}
		128	4.95×10^{-8}	5.1 2.48×10^{-6}	5.0 2.61×10^{-7}	5.1 1.75×10^{-5}
		256	1.58×10^{-9}	5.0 8.61×10^{-8}	4.9 8.00×10^{-9}	5.0 4.54×10^{-7}
		512	4.94×10^{-11}	5.0 2.80×10^{-9}	4.9 2.47×10^{-10}	5.0 1.77×10^{-8}
Taylor degree 4	3	64	9.32×10^{-7}	1.21×10^{-4}	5.56×10^{-6}	1.32×10^{-3}
		128	2.92×10^{-8}	5.0 2.68×10^{-6}	5.5 1.72×10^{-7}	5.0 1.53×10^{-5}
		256	9.20×10^{-10}	5.0 1.01×10^{-7}	4.7 5.40×10^{-9}	5.0 5.07×10^{-7}
		512	2.88×10^{-11}	5.0 3.31×10^{-9}	4.9 1.69×10^{-10}	5.0 1.71×10^{-8}
Taylor degree 5	2	64	4.50×10^{-8}	1.35×10^{-6}	3.24×10^{-7}	2.71×10^{-5}
		128	6.57×10^{-10}	6.1 3.01×10^{-8}	5.5 4.84×10^{-9}	6.1 2.11×10^{-7}
		256	1.03×10^{-11}	6.0 4.61×10^{-10}	6.0 7.06×10^{-11}	6.1 3.39×10^{-9}
		512	1.73×10^{-13}	5.9 7.15×10^{-12}	6.0 1.13×10^{-12}	6.0 5.65×10^{-11}
Taylor degree 5	3	64	4.08×10^{-8}	2.48×10^{-6}	2.87×10^{-7}	2.27×10^{-5}
		128	6.44×10^{-10}	6.0 5.05×10^{-8}	5.6 4.50×10^{-9}	6.0 2.65×10^{-7}
		256	1.00×10^{-11}	6.0 7.70×10^{-10}	6.0 7.04×10^{-11}	6.0 4.27×10^{-9}
		512	1.57×10^{-13}	6.0 1.21×10^{-11}	6.0 1.10×10^{-12}	6.0 7.22×10^{-11}

Table 3. Convergence results (*local error in a narrow band of radius 8 grid cells*) for the ellipse (dimension $d = 2$) and ellipsoid ($d = 3$) for several polynomial classes: the bicubic (in 2D), tricubic (in 3D) and the Taylor polynomials in Table 1. The left pair measures the error in the distance function and the second pair the error in the closest point function. For each polynomial type, the error is indicated for a grid of size $n \times n$ in 2D and $n \times n \times n$ in 3D. Ratios between errors on successive grids yield the convergence rates in bold.

Three more test problems were considered, each with a different degree of smoothness. Here we summarise the convergence results⁹:

- *Circle in 2D and sphere in 3D.* In this case, the distance function is infinitely smooth except for a single isolated point at the origin of the circle/sphere. The results confirm that the distance and closest point functions are p -th order accurate everywhere, except near the singularity where only the closest point function loses two orders of accuracy.
- *Square in 2D and cube in 3D.* This test problem is more subtle. Locally, both the distance and closest point function are second order accurate in the L^1 norm, and first order accurate in the maximum norm, which is to be expected. Globally, the distance function is first order accurate in both norms. However, the closest point function is approximately half-order accurate globally in the maximum norm. The reason for achieving only half-order accuracy is as follows: the interpolation/approximation of the corner of a square inevitably leads to small bumps (see, e.g., [Figure 5](#) on page 129). These small bumps are $\mathcal{O}(h)$ perturbations of the flat edge of the square, and are “seen” far away from the interface. The locus of points for which the distance to the bump equals the distance to the edge of the square approximately forms a parabola; see [Figure 4](#). For a fixed distance away from the edge, the parabola’s width is $\mathcal{O}(\sqrt{h})$. All grid points within the parabola see the bump, leading to a half-order error in the maximum norm of the closest point function. This loss of accuracy applies only to the outside of the square/cube. On the inside, the closest point function is multivalued along the diagonals (either two values in 2D or up to three in 3D). Along these shock lines of the distance function, the algorithm returns the exact distance to the interface (since the flat sides of the square/cube are exactly recovered).
- *Rounded rectangle and cylinder with rounded ends.* This example serves as a somewhat smooth but not infinitely smooth surface. In two dimensions, the interface is a square of width $\frac{1}{2}$ with two semicircles on the left and right sides; see [Figure 4](#). In three dimensions, the interface is a cylinder with two hemispheres on either side. In both cases, the interface has a continuous normal vector field, but its curvature is discontinuous. Results show that both the distance function and closest point function are locally second order accurate, the distance function is globally second order accurate, and the closest point function is approximately second order accurate except near curvature singularities (all in the maximum norm).

⁹These convergence results may be reproduced by the reader with the C++ code accompanying this paper.

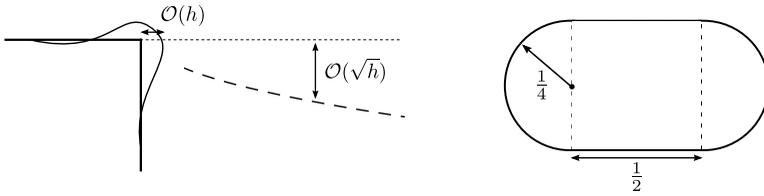


Figure 4. Left: polynomial approximation of a square leads to bumps on the corners which are $\mathcal{O}(h)$ in width and are seen far away from the interface by the closest point function. The set of points for which the distance to the bump equals the distance to the original square approximately forms a parabola (indicated by the dashed line). Right: a rounded rectangle used in one of the convergence tests, consisting of a square of width $\frac{1}{2}$ with two semicircles of diameter $\frac{1}{4}$ on either end.

4.2. Convergence of Newton’s method. Recall that the threshold for deciding convergence in Newton’s method was whether $\|x_{k+1} - x_k\| < \epsilon$. In our test cases, we set¹⁰ $\epsilon = h^p$, where p is the expected order of accuracy of the class of polynomials being used. Across all test problems it was found that in the vast majority of cases, Newton’s method converged within 2–4 iterations. Table 4 illustrates the typical convergence behaviour with a histogram counting the number of steps taken by Newton’s method accumulated across the entire computational grid. Generally

Test case	Polynomial	Number of iterations in Newton’s method								
		1	2	3	4	5	6	7–20	F	E
2D ellipse	Bicubic	0.02	33.6	64.6	1.2	0.2	0.1	0.01	0.2	
2D ellipse	Taylor degree 2	0.02	34.4	65.3	0.2					
2D ellipse	Taylor degree 4		0.3	78.9	20.8					
2D square	Bicubic	0.3	87.8	5.4	4.1	1.1	0.2	0.7	0.01	0.4
2D square	Taylor degree 2	0.3	87.1	8.6	4.0	<0.01				
2D square	Taylor degree 4	0.3	87.2	0.4	7.5	4.6				
3D ellipsoid	Tricubic	<0.01	9.7	89.4	0.6	0.2	0.04	<0.01	0.07	<0.01
3D ellipsoid	Taylor degree 2	<0.01	9.3	90.7	0.05	<0.01				<0.01
3D ellipsoid	Taylor degree 4		<0.01	62.2	37.8					
3D cube	Tricubic	<0.01	72.4	10.9	10.3	3.7	0.5	1.5	0.1	0.6
3D cube	Taylor degree 2	<0.01	71.2	19.1	9.7	<0.01				
3D cube	Taylor degree 4	<0.01	70.7	0.08	16.2	13.0	<0.01	<0.01		

Table 4. Convergence of Newton’s method. In each case, executed on either a 256×256 grid (in 2D) or $256 \times 256 \times 256$ grid (in 3D), the percentage of grid points which needed the indicated number of steps for convergence is shown; entries greater than 10% are in boldface. A blank cell indicates exactly 0%, “F” means Newton’s method did not converge within 20 iterations, and “E” means the iterate left the bounding ball that determines the amount of overlap between adjacent grid cells.

¹⁰In fact, we set $\epsilon = \max(10^{-14}, h^p)$ to ensure that convergence is declared when using a highly resolved grid for which errors are limited to double precision arithmetic.

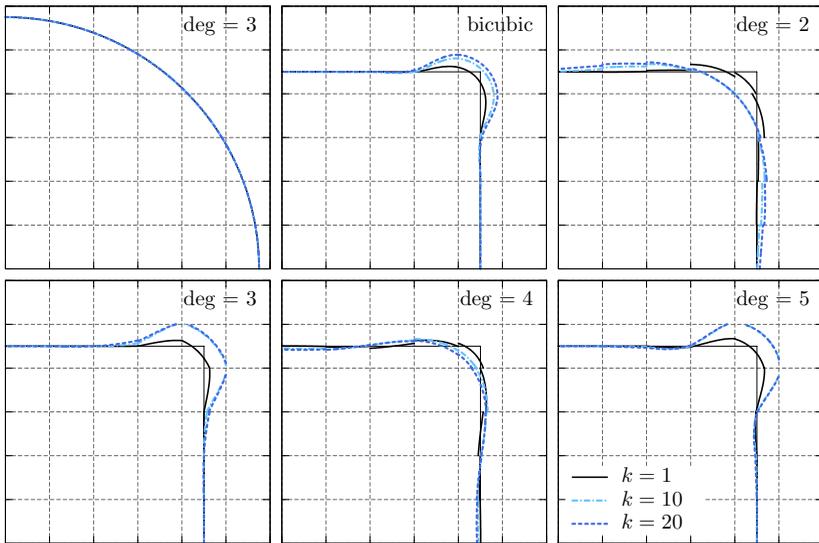


Figure 5. Effect of multiple reinitialisations on an interface, zooming in on a 6×6 patch of grid cells to observe cell-sized effects. Each patch shows the interface after one reinitialisation (solid line), after 10 reinitialisations (dashed-dot line), and after 20 reinitialisations (dashed). Top left: reinitialisation applied to a circle of radius approximately 5.5 grid cells, computed using a Taylor polynomial of degree 3. Remaining grids: reinitialisation applied to the corner of a square where, except for the bicubic, a Taylor polynomial of the indicated degree is used.

speaking, it is very rare that more than 6 iterations are needed. Note also that in the case of the square and cube, the majority of grid points requires just two iterations: when the closest point is on a flat face of the square/cube, only one iteration is needed for convergence, but two iterations are necessary to detect this.

4.3. Repeated reinitialisation. In level set methods, it is often necessary to periodically reinitialise the level set function as a distance function and a common practice for doing this is to reinitialise ϕ every fixed number of steps. Reinitialising as frequently as this may even be necessary to converge to the correct solution, such as in the Voronoi Implicit Interface Method [25; 26] which evolves an unsigned distance function. It follows that an important requirement of a reinitialisation method is that any perturbation in the location of the interface should be made as small as possible. As an example, if the time step for an evolving simulation is $\Delta t = \mathcal{O}(h^2)$, then the level set function will be reinitialised approximately $\mathcal{O}(h^{-2})$ many times over the course of the simulation. The accuracy of the reinitialisation procedure must then necessarily be at least second order accurate — in fact, it often needs to be much higher to ensure that accumulated errors from reinitialisation do not dominate the overall error.

In addition to requirements on the order of accuracy, it should also be confirmed that the reinitialisation method is stable when invoked on the same problem multiple times [10]. Let $R(\phi)$ be the operator which takes a grid-defined level set function ϕ and returns an approximation to the signed distance function evaluated on the same grid. Of interest is the error in the interface of $R \circ \dots \circ R(\phi) = R^k(\phi)$ after k reinitialisations, compared to the original interface of ϕ . Figure 5 illustrates the behaviour for a smooth interface (a circle) and nonsmooth interface (the corner of a square), zooming in on a 6×6 patch of grid cells. In Figure 5, the interface, defined by the zero level set of the relevant polynomial of each grid cell, is shown for $k = 1, 10,$ and 20 . For smooth interfaces, and provided the reinitialisation method is at least third-order accurate, the effect of reinitialisation is essentially unobservable, except on extremely coarse grids. For nonsmooth interfaces, we expect to see $\mathcal{O}(h)$ perturbations; however, we do not wish the amount of perturbation to rapidly grow as k is increased.

To analyse this more carefully, a metric measuring the amount of perturbation is required. Here we use a metric that measures the maximum deviation in the interface, defined by the Hausdorff distance d_H : given two interfaces Γ_1 and Γ_2 (each a surface of codimension-one), define

$$d_H(\Gamma_1, \Gamma_2) = \max(\sup_{x \in \Gamma_1} d(x, \Gamma_2), \sup_{x \in \Gamma_2} d(x, \Gamma_1)), \quad (4)$$

where $d(x, \Gamma_i)$ is the minimum distance from x to interface Γ_i . Figure 6 plots the error¹¹ for a circle in 2D, a sphere in 3D, a square in 2D, and a cube in 3D, for k between 1 and 20 iterations, for all polynomial types considered in this paper. We observe that in all cases, the error after multiple reinitialisation steps is stable and remains on the same order as the original approximation error. Another metric for measuring accuracy of reinitialisation methods is the ability to conserve area/volume — since the Hausdorff metric bounds the error in area/volume conservation, it follows that the closest point algorithm also preserves volume (both locally and globally) with at least the same order of accuracy.¹²

4.4. Computational efficiency. Let N be the number of grid cells containing the interface. Then the basic computational complexity of the algorithm is: (i) $\mathcal{O}(N)$ to

¹¹To actually compute the Hausdorff distance, we supersample each interface by using a subgrid of 10^d subcells per grid cell, such that the exact solution has a cloud of points $\{x_{i,\text{exact}}\} \subset \Gamma_{\text{exact}}$, and the approximate interface has a cloud of points $\{x_{j,h}\} \subset \Gamma_h$. With these source points, we compute $d_H \approx \max(\max_i d(x_{i,\text{exact}}, \Gamma_h), \max_j d(x_{j,h}, \Gamma_{\text{exact}}))$, where $d(\cdot, \Gamma_{\text{exact}})$ is evaluated using knowledge of the exact solution, while $d(\cdot, \Gamma_h)$ is evaluated using Newton's method, similar to that described in Section 3.3, but with convergence to machine precision. Altogether, this is a sufficiently accurate approximation of the true Hausdorff distance.

¹²In the case of the square, which has first order approximation errors at the corners of the square, the area of the reinitialised square is in fact second order accurate.

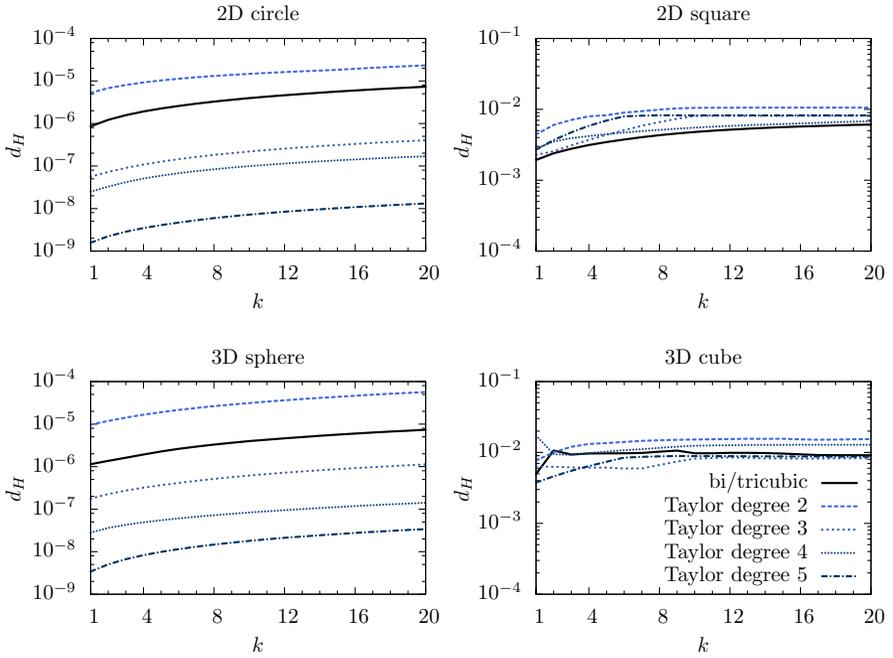


Figure 6. Error in the position of the interface after k repeated reinitialisations, where the error is measured by the Hausdorff distance (4) between the exact interface and the interface defined by the relevant polynomials of each grid cell. The legend in the bottom-right figure applies to all figures. In each case, a domain $[-1, 1]^d$ is subdivided into a grid of 128×128 cells (in 2D) or $128 \times 128 \times 128$ cells (in 3D). Top left: error for a circle of radius $\frac{1}{2}$. Top right: error for a square of width 1. Bottom left: error for a sphere of radius $\frac{1}{2}$. Bottom right: error for a cube of width 1.

construct grid cell polynomials and sample their zero level set; (ii) $\mathcal{O}(N \log N)$, on average, to construct the k -d tree; (iii) $\mathcal{O}(\log N)$, on average, per closest point query in searching the k -d tree for the closest point in C (in the best case it is $\mathcal{O}(1)$; in the worst case it can be $\mathcal{O}(N)$, as for example when x_q is at the centre of a sphere); and (iv) $\mathcal{O}(1)$ cost per query point in applying Newton’s method in all cases. Roughly speaking, timing of individual components of the algorithm shows that:

- When computing the closest point function in a narrow band with radius a fixed number of grid cells (e.g., 5–15): constructing the polynomials takes 15–20% of the time, sampling up to 5%, constructing the k -d tree up to 5%, searching the tree between 40–70%, and running Newton’s method between 15–30%.
- When computing the closest point in the entire domain, the size of N relative to the total number of grid points is more relevant. For medium to highly resolved grids, the majority of the overall computation time is spent solely in searching the k -d tree and running Newton’s method. Depending on the dimension of the problem

Test problem	FMM	High-order
2048 × 2048 grid, narrow band	0.08 s	0.1 s
2048 × 2048 grid, entire domain	1.6 s	1.9 s
256 × 256 × 256 grid, narrow band	1.6 s	1.8 s
256 × 256 × 256 grid, entire domain	20 s	36 s

Table 5. Timing tests for a circular/spherical interface, performed on an Intel i7 3.1 GHz desktop machine (single core), comparing a second-order fast marching method [27] to the high-order closest point algorithm.

and accuracy of the polynomials, this ranges from 20% of the time in Newton’s method and 80% of the time in searching to an even split between the two.

It follows that no single component of the algorithm clearly dominates the overall cost. To provide a general idea of the practical performance of the method, Table 5 compares its speed to a fast marching method which has been optimised for computing distances. (It is important to note that the two methods are intended for different classes of problems, so the comparison in speed should only be used as a guideline.) Further improvements in efficiency could be made by taking into account specific computing architecture, e.g., using more advanced optimisation techniques such as SSE instructions in k-d tree searches, but in the interest of simplicity and code portability these were not considered here.

4.5. Subgrid features. As our final example, we demonstrate that the algorithm for finding closest points on implicitly defined surfaces can accurately capture subgrid features in the interface, such as “droplets” completely contained within one grid cell. The problem setup is as follows. We begin with a level set function ϕ that is defined only at grid points. In the context of subgrid resolution, we then make the natural assumption that subgrid details are successfully captured by high-order approximations of ϕ on each grid cell.

Given this assumption, the pertinent issue in the closest point algorithm is whether the sampling procedure described in Section 3.2 can successfully detect and sample these subgrid details. In Section 3.4 it was discussed how this could be achieved by using polynomial bounds to automatically and adaptively determine where to place points on the zero level set of each grid cell’s polynomial. Once sampled sufficiently well, Newton’s method will successfully find the closest point on the interface. Figure 7 illustrates a pair of two-dimensional examples in which the values of ϕ were defined on the grid points via a scaled version of the function

$$f(x, y) = x^2 - \frac{1}{27}y^3 + \frac{2}{3}y - \alpha.$$

Here α is a parameter determining which level set is considered the interface. Two cases with different values of α are shown in Figure 7, resulting in two topologically

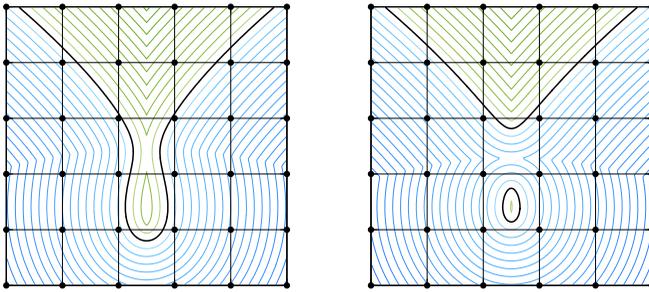


Figure 7. Capturing subgrid details of an interface defined by a high-order approximation on a 5×5 patch of grid cells. Here the level set function values are defined only at grid points. Shown are the contours of the computed signed distance function, evaluated throughout the 5×5 patch of grid cells. Left: a situation where three cells containing the interface have the same sign of the level set function at all their vertices. Right: an interface with two connected components with a droplet completely contained within one grid cell.

different interfaces. In either case, the cells in the middle cannot detect the presence of the interface by examining only the signs of ϕ at their vertices. High-order polynomial approximations can nevertheless recover these subgrid details, and these are accounted for in the closest point algorithm¹³ throughout the 5×5 patch. An analogous problem in three dimensions is shown in [Figure 8](#) using the same-sized grid cells, where again isolated droplets and long thin interfaces are correctly accounted for.

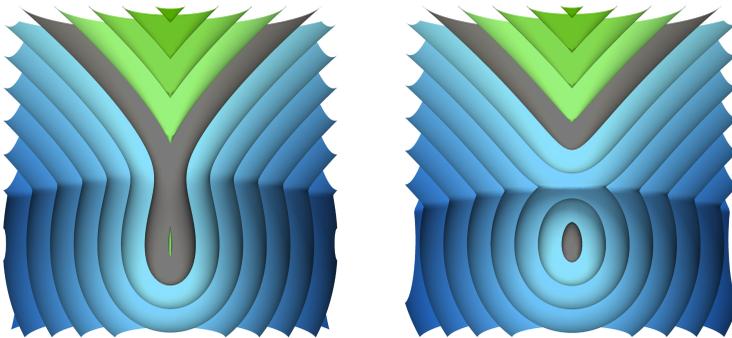


Figure 8. Analogy of [Figure 7](#) in three dimensions corresponding to a $5 \times 5 \times 5$ patch of grid cells for which the interface (shown in dark grey) exhibits subgrid details. Though not shown in the figure, the cell sizes are identical to those in [Figure 7](#). Here contours of the computed signed distance function have been cut by a plane in order to see inner details. Left: a situation in which the interface passes through several grid cells that have the same sign of the level set function on all their vertices. Right: a droplet completely contained within one grid cell.

¹³While automatic sampling is possible, in this particular example we used a simple method that used a subgrid of 10×10 to sample with (see [Section 3.2](#)) based on a Taylor polynomial of degree 3.

5. Concluding remarks

The presented method for computing high-order approximations of closest points on implicitly defined surfaces is straightforward — given a level set function defined by a high-order polynomial on each element of the computational domain, the zero level set is sampled to produce a sufficiently dense cloud of points approximating the interface. A closest point calculation proceeds by first finding the closest point in the cloud, and then improving this guess by using Newton’s method. The results show that the algorithm is both robust and efficient — typically only 2–3 iterations of Newton’s method are required to achieve convergence. In comparison to marching-based or PDE-based methods, for which implementation on unstructured meshes can be subtle, the presented approach can be used on highly unstructured meshes, or indeed at arbitrary query points. In the case of level set functions that are defined on a rectangular Cartesian grid, high-order polynomial approximations based on least-squares interpolation were presented. In other applications, such as gradient augmented level set methods or high-order discontinuous Galerkin finite element methods, the polynomials defining the interface are naturally specified. Convergence tests were performed and showed orders of accuracy of up to six in both the computed distance function and closest point function. For smooth problems, one obtains the optimal order of accuracy in both the computed distance and closest point functions. Near curvature singularities, the distance function remains high-order accurate, but the closest point function may lose up to two orders of accuracy.

The algorithm can be used to accurately reinitialise level set functions in level set methods. Though no time evolving simulations were presented in this paper, it has been successfully applied to a variety of moving interface problems which require very frequent reinitialisation, including in the Voronoi Implicit Interface Method [25; 26] for tracking interconnected interfaces with junctions. Some additional applications include:

- *Nonconstant extensions.* A common method for extending a function f defined on the interface is to make it constant along characteristics of the distance function, i.e., $f_{\text{ext}}(x) = f(\text{cp}(x))$. Some applications require this process to be bootstrapped in such a way that the extension is a linear or quadratic polynomial along characteristics; see, e.g., [29]. This can be achieved by a one-pass algorithm that calculates

$$f_{\text{ext}}(x) = f_0(\text{cp}(x)) + \|x - \text{cp}(x)\| f_1(\text{cp}(x)) + \frac{1}{2} \|x - \text{cp}(x)\|^2 f_2(\text{cp}(x)) + \dots ,$$

where the f_i are the coefficients of the polynomial restricted to a particular characteristic.

- *High-order evaluation of curvature of the signed distance function.* Many applications of the level set method require accurate calculation of the mean curvature κ of the interface or other level sets of the signed distance function. Let \mathbf{n} be the normal vector field of the signed distance function determined by the interface of a given level set function ϕ (which is not necessarily itself a distance function). Then it can be shown that derivatives of \mathbf{n} at a point x are related to the derivatives of \mathbf{n} evaluated at the closest point $\text{cp}(x)$ via

$$\nabla \mathbf{n}(x) = \left(I + \|x - \text{cp}(x)\| \nabla \mathbf{n}(\text{cp}(x)) \right)^{-1} \nabla \mathbf{n}(\text{cp}(x)),$$

while derivatives of \mathbf{n} at a point y on the interface can be evaluated with

$$\nabla \mathbf{n}(y) = \frac{1}{\|\nabla \phi\|} (I - \mathbf{n}\mathbf{n}^T) D^2 \phi (I - \mathbf{n}\mathbf{n}^T) \Big|_y, \quad \mathbf{n}(y) = \frac{\nabla \phi(y)}{\|\nabla \phi(y)\|}, \quad y \in \{\phi = 0\}.$$

These relations can be used to calculate curvature information of the signed distance function, such as the mean curvature $\kappa = \text{tr}(\nabla \mathbf{n})$, via derivatives of ϕ evaluated only at the interface. This fits into the presented framework as we can then use the high-order polynomials approximating the interface itself, rather than relying on a finite difference scheme (say) applied to a precomputed grid-defined signed distance function.

We conclude by briefly describing the C++ code that accompanies this article (available on the author's web site). The code implements all the methods presented in this paper and can be used to verify the convergence results. In particular:

- Much of the code is templated on both the dimension d and the class of polynomials being used. To assist with part of this functionality, the code makes use of *blitz++* [6], an open-source implementation of d -dimensional arrays and fixed-length vectors in C++ with convenient expression template techniques.
- Ten different types of polynomials are provided: bicubic, tricubic, and each of the Taylor polynomials, with their corresponding pseudoinverses precomputed, as well as routines to evaluate the polynomial, its gradient and Hessian, using Horner's method.
- A k-d tree optimised for codimension-one surfaces (as described in the [Appendix](#)) is also supplied.
- A basic method for reinitialising a level set function as a signed distance function is also provided — it can be used to adapt the methods to different polynomial types and other applications.

Appendix: A k-d tree optimised for codimension-one manifolds

Given a fixed query point $x_q \in \mathbb{R}^d$, we would like to determine which point in $C = \{x_1, \dots, x_N\} \subset \mathbb{R}^d$ is closest to x_q . One of the most efficient data structures for closest point queries such as this is a *k-d tree*. A k-d tree organises the set of points into a hierarchy based on geometric considerations and allows for closest point queries in time approximately $\mathcal{O}(\log N)$. Each nonleaf node of the tree has two children: one child contains all the points on the “left” and the other child contains all the points on the “right.” When searching a node in the tree for the closest point, the child which is more likely to contain the closest point is searched first; the other child is searched only if it could potentially contain a closer point than the current candidate. In a conventional k-d tree, “left” and “right” are determined by a hyperplane dividing the node’s set of points into two, with normal direction equal to the x -axis, y -axis, z -axis, etc., cycled down the tree — the k in k-d tree refers to there being k dimensions to cycle through.

In the case that the points come from smooth surfaces, we can use the geometry of the surface itself to improve the efficiency of a k-d tree. The main idea for the tree developed in this work is to apply coordinate transformations in order to create “tight” bounding boxes. By using tighter bounding boxes, larger portions of the tree can be avoided when searching the tree. The essential ideas are as follows; for further details the reader is referred to the C++ code.

A node of the tree is either a leaf node, or else has exactly two children. A leaf node contains `leafsize` many points together with a bounding box of those points; typically `leafsize` is between 10–50 points, tunable according to computer hardware characteristics.¹⁴ Each nonleaf node has four parameters: a pointer to the “left” and “right” children, a bounding box, and a pointer to a rotation matrix. The bounding box is of all the points represented by the node, i.e., the union of the bounding boxes of all its leaf nodes. The rotation matrix pointer, if not null, determines the coordinate transform which has been applied to all points represented by the node.

Delaying the description of constructing the tree for a moment, consider searching the tree to find the closest point to x_q . The basic routine for searching a node recursively is shown in [Algorithm 2](#) and is initiated by calling `search` on the root node with $x = x_q$. The output is the index i of the closest point in C , where d^2 is the squared distance¹⁵ from x_q to point i . Except for [line 6](#), the search procedure is essentially identical to a normal k-d tree. The difference is that some nonleaf nodes may have a rotation matrix R , whose purpose is to apply a coordinate transform to

¹⁴It is often much more efficient to perform a linear search on a handful of points in the leaf nodes, compared to searching a tree whose leaf nodes contain a single point.

¹⁵It is much faster to compute and store squared distances, rather than the distance itself, as the former avoids expensive `sqrt` calls.

```

1: if node is the root then initialise  $i := -1$  and  $d^2 := \infty$ .
2: if node is a leaf then
3:   for each of the points  $x_j$  in the leaf do
4:     if  $\|x_j - x\|^2 < d^2$  then update  $i := j$  and  $d^2 := \|x_j - x\|^2$ .
5: else
6:   if node has a rotation matrix  $R$  then set  $x \leftarrow Rx$ .
7:   Calculate the squared distances  $d_L^2$  and  $d_R^2$  from  $x$  to the bounding boxes of the left
8:     and right child nodes. (If inside a bounding box, the distance is zero.)
9:   if  $d_L^2 < d_R^2$  and  $d_L^2 < d^2$  then
10:     search(left child,  $x$ ).
11:     if  $d_R^2 < d^2$  then search(right child,  $x$ ).
12:   else if  $d_R^2 < d_L^2$  and  $d_R^2 < d^2$  then
13:     search(right child,  $x$ ).
14:     if  $d_L^2 < d^2$  then search(left child,  $x$ ).

```

Algorithm 2. search(node, x).

the query point x (the points contained in the children of such a node have already had their coordinates transformed). Ignoring the role of R for the moment, consider what makes k-d trees efficient: the ability to avoid searching entire parts of the tree. In particular, if the minimum distance from the query point to a child node's bounding box is larger than the distance to the current candidate closest point, then there is no point in searching that child node. It follows that a k-d tree can be made more efficient by attempting to make these bounding boxes as tight as possible. This is where we can utilise the fact that the cloud of points originate from a surface: after enough subdivision, groups of points are situated close to the same tangent plane of the surface, and so a bounding box rotated to align with the plane will be “thin.” One possibility therefore is to store at each node of the tree a rotated bounding box. Instead, a more efficient approach is to transform the points themselves (once only, upon construction of the tree) by applying rotation matrices.

To explain the calculation of these rotation matrices, we turn now to the construction of the tree. Like search, this is also done recursively, and the basic method is shown in Algorithm 3. The tree construction is initiated by calling buildtree on the root node with the entire range of points ($j_\ell = 1$, $j_u = N$). Except for lines 5 to 13, the algorithm is essentially the same as constructing a normal k-d tree. These lines are responsible for deciding whether to perform a coordinate transformation and the specifics of that transform. Generally speaking, it is more efficient to delay the transforms until the k-d tree has subdivided sufficiently many times — on line 6, we can thus use metrics based on the current depth of the tree and how many levels there are to a leaf node; in addition, once a set of points is

rotated, it is essentially unnecessary to consider them again for rotation. If the node is being considered for coordinate transformations, we must estimate the normal n of its set of points. One possibility for doing this is to use principal component analysis to estimate the principal direction for which the points change position the least, leading to an eigenvalue problem on the covariance matrix of the points' positions. A simpler and more efficient method for estimating n is as follows:

Compute the mean $\mu := (j_u - j_\ell + 1)^{-1} \sum_{j=j_\ell}^{j_u} x_j$.

Initialise $n := (1, 0, \dots, 0) \in \mathbb{R}^d$.

for $j = j_\ell$ **to** j_u **do**

$$n \leftarrow n - \frac{(x_j - \mu) \cdot n}{\|x_j - \mu\|^2} (x_j - \mu)$$

if $n \neq 0$ **then return** $n/\|n\|$ **else** do not consider transform.

Geometrically, this procedure removes components from n that are estimated to be in the tangent space. Although the calculation of n depends on the ordering of the points, it is unnecessary to estimate n with a high amount of accuracy. Returning to [Algorithm 3](#), on line 10 the degree to which the new bounding box is “thin” is measured by comparing the new coordinates (in the normal direction) with 10% of the longest length of the untransformed bounding box. The factor of 10% was determined empirically to lead to the best overall efficiency. The remaining details of implementing this k-d tree are left to the C++ code.

In short, the above k-d tree, which has been optimised for point clouds arising from smooth surfaces, is about 4 to 10 times faster than a conventional k-d tree.

The reader may wonder if other methods of characterising the bounding regions may lead to even more efficient tree traversal. For example, if the points are known to come from a sphere (say) or a surface that locally looks like a sphere, one might construct a bounding “shell” which is curved to match the curvature of the sphere. While such an approach is possible, it turns out that computing the distance to a bounding shell is so computationally expensive that the overall cost of traversing the tree is greater, despite there being fewer nodes to search.

One final possibility for very efficient closest point queries is to use a combination of data structures. It is possible to construct a data structure with search operation costing $\mathcal{O}(1)$ (instead of $\mathcal{O}(\log N)$ on average as it is with k-d trees), specifically for points arising from smooth surfaces, provided the surfaces are very finely sampled. The idea is to use a k-d tree (or possibly a quadtree/octree/etc.) for the initial hierarchical subdivision of points, but only use a fixed number of levels so that the depth of the tree is bounded. The leaf nodes of the tree would then represent many thousands of points, and supposing that they essentially form a flat surface, these points could be binned into a conventional $(d - 1)$ -dimensional array, rotated to be tangent with the surface. Finding closest points in an array

-
- 1: Compute the bounding box of the node.
 - 2: **if** $j_u - j_\ell + 1 < \text{leafsize}$ **then**
 - 3: Mark node as a leaf node and record range j_ℓ, j_u .
 - 4: **return**
 - 5: Initialise node's R matrix as null.
 - 6: **if** node is being considered for coordinate transformation **then**
 - 7: Estimate the normal n of the surface approximated by the points $\{x_{j_\ell}, \dots, x_{j_u}\}$.
 - 8: Compute the coordinates of the points as though n was an axis:

$$\alpha_{\min} = \min_{j_\ell \leq j \leq j_u} x_j \cdot n, \quad \alpha_{\max} = \max_{j_\ell \leq j \leq j_u} x_j \cdot n.$$

- 9: Determine the longest length L of the bounding box computed on line 1.
 - 10: **if** $\alpha_{\max} - \alpha_{\min} < 0.1L$ **then**
 - 11: Calculate an orthonormal basis $\{r_1, \dots, r_d\}$ using the normal n as the first axis.
 - 12: Set the node's R matrix as $R = [r_1, \dots, r_d]$.
 - 13: Transform all points: **for** $j_\ell \leq j \leq j_u$ **do** $x_j \leftarrow R x_j$.
 - 14: Determine the axis k along which the (possibly new) bounding box of this node has greatest extent.
 - 15: Calculate the median $m = \lfloor \frac{1}{2}(j_\ell + j_u) \rfloor$.
 - 16: Rearrange the points $\{x_j\}$ such that $x_{j,k} \leq x_{m,k}$ for all $j < m$ and $x_{j,k} \geq x_{m,k}$ for all $j > m$.
 - 17: Split the points into two halves and build the left and right child nodes:
 - node.left = *new* node; buildtree(node.left, j_ℓ, m)
 - node.right = *new* node; buildtree(node.right, $m + 1, j_u$)
 - 18: **return**
-

Algorithm 3. buildtree(node, j_ℓ, j_u).

such as this can be made to have $\mathcal{O}(1)$ cost, provided the points are essentially uniformly scattered throughout array. This idea was tested and compared with the performance of the above k-d tree. Despite being a $\mathcal{O}(1)$ search algorithm, the constant is sufficiently large that no benefits are obtained for standard-sized reinitialisation problems in level set methods—in other words, the interface is rarely “flat enough” compared to the resolution of the grid. The approach may be beneficial for very large 3D problems (such as those arising from $256 \times 256 \times 256$ grids or higher). For medium sized problems, the calculations required to search the k-d tree, and then transform the problem into searching a rotated array of binned particles, is too expensive compared to the above k-d tree with slightly larger depth. (On a related note, different techniques are possible for adapting k-d trees and other space partitioning algorithms to situations where the point data arises from a possibly unknown low-dimensional manifold embedded in a high number of dimensions; see, e.g., [11].)

Acknowledgements

The author thanks Ben Preskill and Ethan Van Andel at UC Berkeley for their helpful feedback in testing the C++ code as part of their work on advanced level set methods for inextensible and elastic surfaces. This research was supported in part by a Luis W. Alvarez Postdoctoral Fellowship at Lawrence Berkeley National Laboratory, the Laboratory Directed Research and Development Program of LBNL, and by the Applied Mathematics Program of the U.S. DOE Office of Advanced Scientific Computing Research under contract number DE-AC02-05CH11231.

References

- [1] D. Adalsteinsson and J. A. Sethian, *A fast level set method for propagating interfaces*, J. Comput. Phys. **118** (1995), no. 2, 269–277.
- [2] D. Adalsteinsson and J. A. Sethian, *Transport and diffusion of material quantities on propagating interfaces via level set methods*, J. Comput. Phys. **185** (2003), no. 1, 271–288.
- [3] D. Adalsteinsson and J. A. Sethian, *The fast construction of extension velocities in level set methods*, J. Comput. Phys. **148** (1999), no. 1, 2–22.
- [4] L. Anumolu and M. F. Trujillo, *Gradient augmented reinitialization scheme for the level set method*, Int. J. Numer. Methods Fluids **73** (2013), no. 12, 1011–1041.
- [5] M. Bertalmio, L.-T. Cheng, S. Osher, and G. Sapiro, *Variational problems and partial differential equations on implicit surfaces*, J. Comput. Phys. **174** (2001), no. 2, 759–780.
- [6] *Blitz++*, 2013, <http://sourceforge.net/projects/blitz/>.
- [7] A. Bøckmann and M. Vartdal, *A gradient augmented level set method for unstructured grids*, J. Comput. Phys. **258** (2014), 47–72.
- [8] J. U. Brackbill, D. B. Kothe, and C. Zemach, *A continuum method for modeling surface tension*, J. Comput. Phys. **100** (1992), no. 2, 335–354.
- [9] D. L. Chopp, *Computing minimal surfaces via level set curvature flow*, J. Comput. Phys. **106** (1993), 77–91.
- [10] ———, *Some improvements of the fast marching method*, SIAM J. Sci. Comput. **23** (2001), no. 1, 230–244.
- [11] S. Dasgupta and Y. Freund, *Random projection trees and low dimensional manifolds*, Proceedings of the 40th Annual ACM Symposium on Theory of Computing, STOC '08, 2008, pp. 537–546.
- [12] A. du Chéné, C. Min, and F. Gibou, *Second-order accurate computation of curvatures in a level set framework using novel high-order reinitialization schemes*, J. Sci. Comput. **35** (2008), 114–131.
- [13] P. Esser and J. Grande, *An accurate and robust finite element level set redistancing method*, Tech. Report 379, Institut für Geometrie und Praktische Mathematik, 2013.
- [14] M. W. Jones, J. A. Baerentzen, and M. Sramek, *3d distance fields: A survey of techniques and applications*, IEEE Trans. Vis. Comput. Graphics **12** (2006), no. 4, 581–599.
- [15] C. B. Macdonald and S. J. Ruuth, *Level set equations on surfaces via the closest point method*, J. of Sci. Comput. **35** (2008), 219–240.
- [16] R. Malladi, J. A. Sethian, and B. C. Vemuri, *Shape modeling with front propagation: A level set approach*, IEEE Trans. Pattern Analysis and Machine Intelligence **17** (1995), no. 2, 158–175.

- [17] C. Min, *On reinitializing level set functions*, J. Comput. Phys. **229** (2010), no. 8, 2764–2772.
- [18] J.-C. Nave, R. R. Rosales, and B. Seibold, *A gradient-augmented level set method with an optimally local, coherent advection scheme*, J. Comput. Phys. **229** (2010), no. 10, 3802–3827.
- [19] S. Osher and R. Fedkiw, *Level set methods and dynamic implicit surfaces*, Applied Mathematical Sciences, Springer, 2003.
- [20] S. Osher and J. A. Sethian, *Fronts propagating with curvature-dependent speed: Algorithms based on Hamilton-Jacobi formulations*, J. Comput. Phys. **79** (1988), no. 1, 12–49.
- [21] P.-O. Persson and G. Strang, *A simple mesh generator in Matlab*, SIAM Review **46** (2004), no. 2, 329–345.
- [22] A. Reusken, *A finite element level set redistancing method based on gradient recovery*, SIAM J. Numer. Anal. **51** (2013), no. 5, 2723–2745.
- [23] G. Russo and P. Smereka, *A remark on computing distance functions*, J. Comput. Phys. **163** (2000), no. 1, 51–67.
- [24] R. I. Saye, *An algorithm to mesh interconnected surfaces via the Voronoi interface*, Engin. Comput. (2013), 1–17.
- [25] R. I. Saye and J. A. Sethian, *The Voronoi Implicit Interface Method for computing multiphase physics*, Proc. Nat. Acad. Sci. **108** (2011), no. 49, 19498–19503.
- [26] ———, *Analysis and applications of the Voronoi Implicit Interface Method*, J. Comput. Phys. **231** (2012), no. 18, 6051–6085.
- [27] J. A. Sethian, *A fast marching level set method for monotonically advancing fronts*, Proc. Nat. Acad. Sci. **93** (1996), 1591–1595.
- [28] ———, *Level set methods and fast marching methods: Evolving interfaces in geometry, fluid mechanics, computer vision, and materials sciences*, Cambridge University Press, 1999.
- [29] J. A. Sethian and Y. Shan, *Solving partial differential equations on irregular domains with moving interfaces, with applications to superconformal electrodeposition in semiconductor manufacturing*, J. Comput. Phys. **227** (2008), no. 13, 6411–6447.
- [30] J. A. Sethian and P. Smereka, *Level set methods for fluid interfaces*, Annual Review of Fluid Mechanics **35** (2003), 341–372.
- [31] J. Strain, *Fast tree-based redistancing for level set computations*, J. Comput. Phys. **152** (1999), no. 2, 664–686.
- [32] M. Sussman and E. Fatemi, *An efficient, interface-preserving level set redistancing algorithm and its application to interfacial incompressible fluid flow*, SIAM J. Sci. Comput. **20** (1999), no. 4, 1165–1191.
- [33] M. Sussman and M. Y. Hussaini, *A discontinuous spectral element method for the level set equation*, J. Sci. Comput. **19** (2003), no. 1–3, 479–500.
- [34] M. Sussman, P. Smereka, and S. Osher, *A level set approach for computing solutions to incompressible two-phase flow*, J. Comput. Phys. **114** (1994), no. 1, 146–159.

Received January 23, 2014. Revised April 30, 2014.

ROBERT I. SAYE: rsaye@lbl.gov

Lawrence Berkeley National Laboratory and Department of Mathematics, One Cyclotron Road,
MS: 50A-1148, Berkeley, CA 94720, United States

<http://math.lbl.gov/~saye/>

ON INFERENCE OF STATISTICAL REGRESSION MODELS FOR EXTREME EVENTS BASED ON INCOMPLETE OBSERVATION DATA

OLGA KAISER AND ILLIA HORENKO

We present a computationally efficient, semiparametric, nonstationary framework for statistical regression analysis of extremes with systematically missing covariates based on the generalized extreme value (GEV) distribution. It is shown that the involved regression model becomes nonstationary if some of the relevant model covariates are systematically missing. The resulting nonstationarity and the ill-posedness of the inverse problem are resolved by deploying the recently introduced finite-element time-series analysis methodology with bounded variation of model parameters (FEM-BV). The proposed FEM-BV-GEV approach allows a well-posed problem formulation and goes beyond probabilistic a priori assumptions of methods for analysis of extremes based on, e.g., nonstationary Bayesian mixture models, smoothing kernel methods or neural networks. FEM-BV-GEV determines the significant resolved covariates, reveals directly their influence on the trend behavior in probabilities of extremes and reflects the implicit impact of missing covariates. We compare the FEM-BV-GEV approach to the state-of-the-art GEV-CDN methodology (based on artificial neural networks) on test cases and real data according to four criteria: (1) information content of the models, (2) robustness with respect to the systematically missing information, (3) computational complexity and (4) interpretability of the models.

1. Introduction

Modeling of extreme events plays a crucial role in different areas of science (e.g., in weather/climate research, economics, biology/medicine) Simulation and prediction of such events is challenging since by definition they are rare and occur irregularly. To approach these challenges, statistical modeling of extreme events is widely accepted (as an alternative to deterministic physical/mathematical modeling based on “first principles”). Extreme value analysis (EVA) is a standard tool in statistics for description of probability distributions of extremes; its foundations were laid by

MSC2010: primary 62G05, 62G32, 65R32; secondary 65C50, 62F03.

Keywords: generalized extreme-value distribution, systematically missing information, nonstationary time-series analysis, nonparametric statistics, finite-element method.

E. Gumbel, W. Weibull and M. Fréchet already in the first half of the 20th century; for more details, see [13; 17; 19]. EVA distinguishes between two types of extremes: (a) exceedances over a higher or lower threshold and (b) block maxima or minima, e.g., annual extremes. In this work, we focus on block maxima. Consider a sample of identical and independently distributed (i.i.d.) variables Y_1, \dots, Y_n with common distribution $F(\cdot)$. Analogously to the limit distribution of the partial sums of the sample, described by the central limit theorem, there exists a limit distribution for the sample extremes $X_n = \max\{Y_1, \dots, Y_n\}$ (or $X_n = \min\{Y_1, \dots, Y_n\}$) as $n \rightarrow \infty$: the limit distribution of $\mathbb{P}[X_n \leq x] = F^n(x)$, as $n \rightarrow \infty$, is the generalized extreme value (GEV) distribution introduced by Fisher and Tippet (1928) and Gnedenko (1943), which has the form (see [17, Theorem 1.1.13])

$$G(x; \mu, \sigma, \xi) = \exp\left(-\left[1 + \xi \frac{x - \mu}{\sigma}\right]^{-1/\xi}\right) \quad (1)$$

with location, scale and shape parameters $\mu, \sigma, \xi \in \mathbb{R}$, respectively, and subject to $[1 + \xi(x - \mu)/\sigma] > 0$ and $\sigma > 0$. By fitting model (1) to a series of block maxima, we imply an unchanging behavior of the underlying dynamics (stationarity). This is obviously not always the case; e.g., in the context of climatology/meteorology, the monthly maxima of temperature or precipitation should be affected by the periodic seasonal effects. The most general way to release the stationarity assumption is to include external influence (denoted as covariates, modes or factors) by constructing the GEV parameters as functions of covariates, i.e., as regression models. And thus, the aim of data-based analysis of block maxima will be to infer the values of the GEV regression parameters from observed data. Standard state-of-the-art methods applicable to this task can be roughly divided into two groups: parametric and nonparametric regression approaches. In parametric approaches, the GEV parameters can be expressed as linear combinations of some explicitly known and given functions (e.g., sine/cosine functions to model the seasonal trends in meteorology). The nonlinearity of explicit covariate combinations is achieved deploying the standard tools from machine learning, e.g., artificial neural networks (ANNs) [8] and support vector machines (SVMs) [35]. Combination of GEV with a special form of ANN called conditional density-estimation network (CDN) has recently led to a creation of the GEV-CDN [8], a robust and flexible approach to the nonstationary and nonlinear extension of (1). However, GEV-CDN as well as all other parametric nonstationary extensions of (1) rely on the explicit availability of all of the relevant covariates and some strong probabilistic assumptions about the systematically missing/unresolved covariates, e.g., i.i.d. assumption for unresolved covariates. As a result, these methods implicitly assume time independence of regression coefficients (e.g., of the hidden neurons weights and biases in the case of GEV-CDN). But due to the multiscale nature of most of the realistic applications

(e.g., in climate research, economics or biology/medicine), one would never be able to guarantee that the set of information collected about the analyzed system is complete. One would also not be able to guarantee that all of the necessary probabilistic assumptions are fulfilled a priori for the analyzed system. As demonstrated in this paper, the violation of the i.i.d. condition for the systematically missing covariates leads to the conceptual and practical problems of the standard parametric approaches and may result in the wrong/biased analysis of the statistics of extremes.

Nonparametric approaches for regression analysis of block maxima exploit local likelihood smoothing [16] or Bayesian techniques [15; 42]. The limitations of these methods are their locality (e.g., a local stationarity assumption) and a priori parametric assumptions about the distributions of the GEV parameters. Another strategy is to involve mixture models and hidden Markov models (HMM) [38; 3; 2]. Such approaches require a priori knowledge about the probabilistic model for the time-dependent GEV parameters, e.g., stationarity and Markov assumptions for the hidden parameter switching process. Additionally, all of the above-mentioned state-of-the-art methods may lead to the ill-posed parameter identification problems (in sense of Hadamard [22]), resulting in the over-fitting of the available data. Most of the above approaches are realized as optimization algorithms for some nonlinear, nonconvex and often nondifferentiable quality functionals. That is, the obtained results are not unique and depend strongly on the initial values and other tuning parameters of the respective computational algorithms. One of the most frequently used methods of transforming an ill-posed problem to a well-posed one is called regularization. This approach is based on imposing some additional (reasonable) assumptions on the original problem formulation, e.g., that the solution should be “small” in an appropriately chosen norm [45]. In statistics and different areas of applied data analysis, approaches like Tikhonov and LASSO regularization are widely used in the context of, e.g., parametric regression and spline-interpolation problems [48; 44; 24], support vector machines [47], compressed sensing and matrix-completion methods [7].

Here we exploit a regularized and nonparametric strategy for general parameter identification in nonstationary problems [26; 27; 41]. It is based on the finite-element discretization (FEM) of the resulting inverse problem subject to bounded variation (BV) of the nonstationary model parameters in time. The FEM-BV framework allows computationally very efficient and highly scalable numerical implementation, either based on the adaptive FEM solvers (usually deployed for the adaptive numerical solution of partial differential equations) [25] or based on adaptive Markov-chain Monte Carlo (MCMC) schemes [18]. Resulting framework was demonstrated to be appropriate for a wide range of nonstationary inverse problems and applications, ranging from climate/weather research [26; 27], molecular dynamics [29] and DNA sequence analysis [41] to sociology [28] and economics [41].

In this paper, we present a unified approach for joint solving of all of the above discussed difficulties (i.e., the problem of systematically missing covariates, numerical complexity and the ill-posedness/over-fitting problem) in data-based analysis of block maxima. To address the issue of missing covariates, we exploit the central limit theorem for independent variables and express each GEV parameter by a fully non-stationary regression model, based only on resolved covariates, with a nonstationary additive noise. The resulting nonstationarity of (1) is interpolated by a linear convex combination of $K \geq 1$ local stationary models and a nonstationary switching process between them. The corresponding inverse problem is regularized by employing FEM-BV methodology [25; 41]. The resulting quality functional is optimized by adjusting the adaptive MCMC methodology (originally proposed in [18]) to a numerical solution of the resulting FEM-BV-GEV problem in optimization formulation.

The FEM-BV-GEV approach described in the current manuscript avoids a priori assumptions on stationarity and trend behavior of the GEV parameters. The proposed method allows an explicit data-driven recovery of the implicit impacts of unresolved modes in the situations when these unresolved modes are neither i.i.d. nor available in the measurement. The local linearity of the GEV parameter functions enables direct interpretation of the influence of covariates on the underlying dynamics of block maxima but can lead to the biased results in cases when the dynamics is locally highly nonlinear (i.e., in the scenarios where parametric GEV-CDN methodology based on neural networks is supposed to produce better estimates). We show that under appropriate assumptions FEM-BV-GEV includes/extends standard techniques based on HMM or kernel smoothing and, therefore, consider the nonlinear GEV-CDN approach as a main competitor for the FEM-BV-GEV in a series of numerical studies. This work shows that the resulting numerical framework, despite the local linearity, addresses the above-mentioned difficulties of the standard methods and demonstrates high robustness with respect to systematically missing covariates and is computationally highly efficient. These issues make the proposed methodology an adequate tool for analysis of extremes in very large realistic applications.

This manuscript is organized as follows. In Section 2, we derive in detail the FEM-BV-GEV approach. We compare FEM-BV-GEV to the state-of-the-art methods from conceptual and applied viewpoints in Section 3. The iterative numerical algorithm of FEM-BV-GEV is described in Section 4. In Section 5, we exemplify the application of FEM-BV-GEV and compare its performance with GEV-CDN [8] on test-cases and regression analysis of temperature extremes (30-day maxima for the period between 1950 and 2011) for Lugano and Berlin. Comparison is performed according to the four criteria: (1) information content of the models (jointly measuring complexity and quality of the model fit), (2) robustness with respect to the systematically missing information, (3) computational complexity and (4) understandability/interpretability of the models.

2. FEM-BV-GEV

In this work, we focus on the fully time-dependent GEV distribution defined by its probability density function (pdf)

$$f(x; \mu(t), \sigma(t), \xi(t)) = c(t) \exp\left(-\left[1 + \xi(t) \frac{x - \mu(t)}{\sigma(t)}\right]^{-1/\xi(t)}\right), \quad (2)$$

where t denotes the time variable and $c(t)$ the normalization constant

$$c(t) = \frac{1}{\sigma(t)} \left[1 + \xi(t) \frac{x - \mu(t)}{\sigma(t)}\right]^{-1/\xi(t)-1} \quad (3)$$

and the model parameters have to fulfill the constraints

$$\left[1 + \xi(t) \frac{x - \mu(t)}{\sigma(t)}\right] > 0 \quad \text{and} \quad \sigma(t) > 0 \quad \forall t. \quad (4)$$

In order to address the time-dependence, we intend to express each GEV parameter as a function dependent on covariates as a linear regression model. However, in real applications, one is usually confronted with the problem that some (or most) potentially relevant covariates are missing in the measurements.

One possible source for the systematically missing covariates is the multiscale dynamics of the underlying process; e.g., processes in climate or molecular dynamics may involve multiple time and length scales [39; 40; 11]. That is, only observing modes on a slow time scale (resolved modes), we neglect modes on the faster scale (unresolved modes). An additional reason for the missing information/measurements is that, even on just one single time scale, one cannot resolve all covariates because one is interested in regression models with a finite number of degrees of freedom. In particular, this is true for regression analysis of extremes because of the relatively small statistics. Thus, we have to select a set of resolved covariates and to account for the influence coming from the systematically unresolved/missing information.

Several disciplines cover the issue of missing information; e.g., in statistical regression analysis, the issue of unresolved information is often addressed under the theme “unobserved heterogeneity” [6]. Thereby, the unobserved covariates are included via a stationary error term into the regression model and the posterior model depends on the a priori assumption about the distribution of this error term. However, there is often no closed expression of the posterior.

In this work, we reduce the involved linear regression model by splitting it into two linear parts, corresponding to resolved and unresolved modes, and incorporate the influence of unresolved modes as a nonstationary additive noise. In the following, we consider all possible modes dependent on time t and split them into resolved $U(t) \in \mathbb{R}^S$ and unresolved $U^{\text{un}}(t) \in \mathbb{R}^Q$ factors, further on denoted as $U_t = (u_1(t), \dots, u_S(t))$ and $U_t^{\text{un}} = (u_1^{\text{un}}(t), \dots, u_Q^{\text{un}}(t))$. Then we normalize the

latter and obtain

$$\mu(U_t, U_t^{\text{un}}) = \mu_0 + \sum_{s=1}^S \mu_s u_s(t) + \frac{1}{Q} \sum_{q=1}^Q v_q u_q^{\text{un}}(t), \quad (5)$$

where $\mu_s, s = 1, \dots, S$, and $v_q, q = 1, \dots, Q$, are the regression coefficients. Under the assumption that the unresolved modes are i.i.d. for all t , application of the central limit theorem reduces the unresolved modes to the additive noise

$$\mu(U_t) = \mu_0 + \sum_{s=1}^S \mu_s u_s(t) + \epsilon \quad \text{with } \epsilon \sim \mathcal{N}(\hat{\mu}, \hat{\sigma}). \quad (6)$$

In real applications, the i.i.d. assumption may be too strong. Instead, we can apply the central limit theorem for independent variables in a formulation that requires a much weaker Lindeberg condition [36]. And in case the modes are not independent, the Karhunen–Loève transformation can be used to decorrelate the processes [37; 33]. Thus, under the assumption that the Lindeberg condition holds, we rewrite (5):

$$\begin{aligned} \mu(U_t, U_t^{\text{un}}) = \mu_0 + \sum_{s=1}^S \mu_s u_s(t) \\ + \underbrace{\frac{1}{Q} \sum_{q=1}^Q v_q (u_q^{\text{un}}(t) - \mathbb{E}[u_q^{\text{un}}(t)])}_{\rightarrow \epsilon(t)} + \frac{1}{Q} \sum_{q=1}^Q v_q \mathbb{E}[u_q^{\text{un}}(t)]. \end{aligned} \quad (7)$$

By inserting $\mu_0(t) = \mu_0 + (1/Q) \sum_{q=1}^Q v_q \mathbb{E}[u_q^{\text{un}}(t)]$ and $\epsilon(t) \sim \mathcal{N}(0, \hat{\sigma}(t))$ into (7), we obtain the reduced, nonstationary regression model:

$$\mu(t, U_t) = \mu_0(t) + \sum_{s=1}^S \mu_s u_s(t) + \epsilon(t). \quad (8)$$

Note that in the regression formulation (8) parameter $\mu_0(t)$ is a time-dependent function and not a constant number as in the case of parametric statistics. That is, application of parametric approaches would produce biased results, and thus, nonparametric statistical methods would be required in such situations. Moreover, without loss of consistency, we generalize (8) by releasing the stationarity assumption of the coefficients μ_s for $s = 1, \dots, S$:

$$\mu(t, U_t) = \mu_0(t) + \sum_{s=1}^S \mu_s(t) u_s(t) + \epsilon(t). \quad (9)$$

Analogously to (9), we express the scale and the shape parameters:

$$\sigma(t, U_t) = \sigma_0(t) + \sum_{s=1}^S \sigma_s(t) u_s(t) + \tilde{\varepsilon}(t), \quad \tilde{\varepsilon}(t) \sim \mathcal{N}(0, \tilde{\sigma}(t)), \quad (10)$$

$$\xi(t, U_t) = \xi_0(t) + \sum_{s=1}^S \xi_s(t) u_s(t) + \bar{\varepsilon}(t), \quad \bar{\varepsilon}(t) \sim \mathcal{N}(0, \bar{\sigma}(t)). \quad (11)$$

The regression models in (9)–(11), which are reduced to resolved covariates only, become stochastic. Thereby, each GEV model parameter has, e.g., a normal distribution as a prior in Bayesian inference context [13]. Since there is no closed formulation for the resulting posterior, MCMC-based algorithms can be used to obtain the distribution of the posterior [13]. In the current manuscript, we focus on the mean behavior of parameters and thus omit the normal distributed noise terms in (9)–(11). Please note, by considering the mean behavior, we obtain deterministic model parameters, which still account for the unresolved information through the nonstationary bias/off-set terms $\mu_0(t)$, $\sigma_0(t)$ and $\xi_0(t)$. The consideration of the complete stochastic regression model with explicit error terms remains for future study. Finally, the nonstationary GEV distribution (2) is parametrized by

$$\Theta(t) = (\mu_0(t), \dots, \mu_S(t), \sigma_0(t), \dots, \sigma_S(t), \xi_0(t), \dots, \xi_S(t)). \quad (12)$$

In this work, we aim to avoid a priori probabilistic or deterministic assumptions on $\Theta(t)$. Instead, we approximate the nonstationary distribution of block maxima by $K \geq 1$ local stationary GEV distributions and a hidden/latent switching process. Thereby, we consider a nonparametric and nonstationary hidden switching process in order to avoid a priori assumptions like stationarity or Gaussian or Markovian behavior (necessary for mixture models and HMMs [38; 3; 2]). Elimination of a priori assumptions implies ill-posedness of the optimization problem in the sense of [22]: in each time step, there will be infinitely more unknown variables than observations. To regularize the problem, we apply the FEM-BV methodology for time-series analysis introduced in [25; 26; 27]. FEM-BV formulates the inverse problem for nonstationary dynamical systems as a regularized variational problem by discretizing the hidden switching process with finite elements and restricting its bounded variation. In the following, we formulate the FEM-BV-GEV approach in two steps: (a) interpolation and (b) regularization.

2.1. FEM-BV interpolation. The FEM-BV approach assumes that the model parameter $\Theta(t)$ changes slower than the observed series $X(t)$ (in the following denoted by X_t with $t = 1, \dots, T$). Then the underlying dynamics can be approximated by a set of $K \geq 1$ local stationary models and a nonstationary switching process $\Gamma(t) = (\gamma_1(t), \dots, \gamma_K(t))$. For that, FEM-BV interpolates the model distance function $g(X_t, \Theta(t))$ (describing the error of the nonstationary model with parameters

$\Theta(t)$ at time t in reproducing the observation X_t by a linear convex combination of K local stationary model distance functions.

The FEM-BV approach can be ranged into the class of clustering-based methods, where the K local stationary models correspond to K clusters and the switching process $\Gamma(t)$ is the affiliation of the data to one of the clusters. Most popular standard methods of data clustering (such as K -means, Gaussian mixture model, hidden Markov models, etc.) can be obtained as unregularized special cases of this more general FEM-BV-framework. These standard methods can be obtained in FEM-BV-framework for some specific choices of the model error function $g(X_t, \Theta(t))$; e.g., the choice of the l^2 -distance between X_t and $\Theta(t)$ as $g(X_t, \Theta(t))$ with no further regularization of $\Gamma(t)$ results in the standard K -means clustering [41]. The connection to classical mixture models and hidden Markov models is discussed in Section 3. We apply the FEM-BV interpolation to our problem by considering for each local GEV model the parametrization

$$\mu_i(U_t) = \mu_{i0} + \sum_{s=1}^S \mu_{is} u_s(t), \quad i = 1, \dots, K, \quad (13)$$

and analogous expressions for $\sigma_i(U_t)$ and $\xi_i(U_t)$ and defining the local model distance function as the local negative log-likelihood function with $\theta_i = (\mu_{i0}, \dots, \mu_{iS}, \sigma_{i0}, \dots, \sigma_{iS}, \xi_{i0}, \dots, \xi_{iS})$, $i = 1, \dots, K$,

$$g_{\text{GEV}}(X_t, \theta_i) = \log(\sigma_i(U_t)) + \left(1 + \xi_i(U_t) \frac{X_t - \mu_i(U_t)}{\sigma_i(U_t)}\right)^{-1/\xi_i(U_t)} + \left(1 + \frac{1}{\xi_i(U_t)}\right) \log\left(1 + \xi_i(U_t) \frac{X_t - \mu_i(U_t)}{\sigma_i(U_t)}\right). \quad (14)$$

Then for $\Theta = (\theta_1, \dots, \theta_K)$, the averaged (interpolated) model distance functional is defined by

$$\mathcal{L}(\Gamma(t), \Theta) = \sum_{t=1}^T \sum_{i=1}^K \gamma_i(t) g_{\text{GEV}}(X_t, \theta_i) \quad (15)$$

with constraints on model parameters

$$\left[1 + \xi_i(U_t) \frac{X_t - \mu_i(U_t)}{\sigma_i(U_t)}\right] > 0 \quad \text{and} \quad \sigma_i(U_t) > 0 \quad (16)$$

for $t = 1, \dots, T$ and $i = 1, \dots, K$

and with convexity constraints on $\Gamma(t) = (\gamma_1(t), \dots, \gamma_K(t))$

$$\sum_{i=1}^K \gamma_i(t) = 1, \quad t = 1, \dots, T, \quad (17)$$

$$\gamma_i(t) \geq 0, \quad t = 1, \dots, T, \quad i = 1, \dots, K. \quad (18)$$

2.2. FEM-BV regularization on $\Gamma(t)$. Minimization of (15) with constraints (16)–(18) is ill-posed. FEM-BV regularization exploits the observation that many real processes feature persistent behavior. Persistence can be imposed through the bounded variation of the switching process $\Gamma(t)$ by

$$|\gamma_i|_{\text{BV}(0,T)} = \sum_{t=1}^{T-1} |\gamma_i(t+1) - \gamma_i(t)| \leq C_i, \quad i = 1, \dots, K, \quad (19)$$

where C_i denotes the maximal number of allowed transitions between the model i and all the other models in the time interval $[1, T]$; further on, we will refer to $C = \max\{C_1, \dots, C_K\}$. Please notice that, since the natural boundary of C is given by T (the number of observed time steps), involving constraint (19) into the optimization problem does not confine the solution space. Now the minimization of (15) with constraints (16)–(19) is well-posed according to $\Gamma(t)$. In the following, we denote the minimization problem

$$(\Gamma^*(t), \Theta^*) = \underset{\Gamma(t), \Theta}{\operatorname{argmin}} \mathcal{L}(\Gamma(t), \Theta) \quad \text{with respect to constraints (16)–(19)} \quad (20)$$

as the FEM-BV-GEV approach and the optimal FEM-BV-GEV parameters as $(\Gamma^*(t), \Theta^*)$.

2.3. Model selection. In this section, we discuss how to choose the optimal FEM-BV-GEV parameters K and C . Moreover, we aim to detect the most significant combination of resolved covariates $U_t \in \mathbb{R}^S$ (S is the number of resolved covariates); in the following, we denote each combination by u_{comb} . Thus, for different K , C and u_{comb} , we apply the FEM-BV-GEV approach and obtain a candidate model M . Denoting the number of all possible K as N_K and the number of all possible C as N_C , we obtain in total

$$N_K N_C \sum_{s=1}^S \frac{S!}{(S-s)! s!} \quad (21)$$

different models and choose the optimal one according to model selection criteria, e.g., the second-order Akaike information criteria (AIC_c) [30]

$$\text{AIC}_c = 2L + 2|M| + \frac{2|M|(|M| + 1)}{T - |M| - 1}, \quad (22)$$

where L is the negative log-likelihood function for the estimated model M , $|M|$ denotes the number of parameters in model M and T is the length of the data. In FEM-BV-GEV formulation, the averaged model distance functional (15) corresponds to the averaged negative log-likelihood (NLL): $-L = \mathcal{L}(\Gamma(t), \Theta)$. The number of parameters depends on K , C and the dimension of u_{comb} so that

$|M| = |M(K, C, u_{\text{comb}})|$. AIC_c is a valid estimate for information content of data with finite length [5]. We compute AIC_c for each model M and choose the best model, denoted by M^* , with respect to $\min(\text{AIC}_c)$. Thus, incorporating the model selection criteria, the “complete FEM-BV-GEV formulation” is given by (20) and (22). In case S is big, the number of models M in (21) increases very fast and running over all possible combinations of U_t becomes computationally expensive. Instead, we can incorporate the LASSO shrinkage technique [44] on model parameter Θ :

$$|\theta_i|_{L_1} \leq C_L, \quad i = 1, \dots, K. \quad (23)$$

This constraint provides not only the most significant u_{comb} by setting the coefficients of insignificant covariates to zero but also improves the prediction accuracy of the regression by shrinking the coefficients [48; 24]. Also here we have to find the optimal C_L . Thus, with N_L as the number of all possible C_L , the total number of all models is $N_K N_C N_L$. The “LASSO FEM-BV-GEV formulation” is described by (20), (23) and (22), where now $|M| = |M(K, C, S)|$.

3. Conceptual comparison with standard methods

The FEM-BV-GEV is a semiparametric approach as a combination of the parametric GEV and the nonparametric FEM-BV description of the hidden switching process. The influence of unresolved factors, expressed as the nonstationarity of model parameters (9)–(11), is reflected by $\Gamma(t)$. The key issue that makes the FEM-BV-GEV problem well-posed is the fact that decreasing the value of C in (19) results in shrinking of the parameter space for $\Gamma(t)$, limiting the number of the local minima for $\mathcal{L}(\Gamma(t), \Theta)$ in (20). The current realization of the FEM-BV-GEV approach assigns $\gamma_i(t) \in \{0, 1\}$, $i = 1, \dots, K$, for all t . That is, according to the result in [41], interpolation of the model distance function provides the direct interpolation of the nonstationarity of the model parameter $\Theta(t)$:

$$\Theta(t) \approx \sum_{i=1}^K \gamma_i(t) \theta_i. \quad (24)$$

Moreover, the FEM-BV-GEV approach includes some state-of-the-art approaches as special cases: in case the whole information is provided for the regression analysis of extremes, FEM-BV-GEV with $C = 0$ (no transitions between the models and thus $K = 1$) corresponds to stationary parametric regression models and results in a well-posed inverse problem. For $K \geq 2$, FEM-BV-GEV provides a nonlocal extension of the nonparametric kernel smoothing approach: equivalently to adaptive multimodal optimization, the nonstationary switching process $\Gamma(t)$ enables us to consider all observations that underlie similar dynamics as one ensemble (in contrast

to, e.g., methods based on a moving window [16; 10] where the kernel is a priori chosen as some fixed local parametric function, e.g., a Gaussian of a certain width).

Furthermore, under an additional a priori assumption that $\Gamma(t)$ is a homogenous Markov process, FEM-BV-GEV is restrained to the family of hidden Markov models (HMMs) as applied to GEV, e.g., becoming equivalent to the standard HMM-based methods of extreme value analysis [2; 1]. However, the difference between the FEM-BV-GEV and Bayesian techniques for analysis of block maxima, e.g., based on mixture models or HMMs, is in the modeling of the hidden process or the hidden variable, respectively. In more detail, the main conceptual advantage of FEM-BV in its general form over the Bayesian mixture models, e.g., HMM, is that FEM-BV is a nonparametric approach. HMM is a purely parametric approach with strong a priori assumptions. In the HMM context, the hidden process is parametrized by a probabilistic model, e.g., homogenous Markov, and requires an initial hidden probability. In contrast, FEM-BV avoids a priori parametrization and is also applicable beyond these assumptions. The only two assumptions involved in FEM-BV are (1) $\Gamma(t)$ is considered as a function in some (very broad) function space, e.g., BV-space and (2) the smoothness of $\Gamma(t)$ is bounded in the corresponding function space (compare also discussions in [41]). The estimation of $\Gamma(t)$ results in a linear minimization problem [25] or can be carried out using MCMC techniques [18]. Both approaches result in very efficient computational numerical schemes scaling well for very large problems unlike the expectation maximization algorithm (the most prominent and widely used algorithm for Bayesian mixture models).

Thus, exploiting the advantages of FEM-BV and involving stochastic mode reduction for linear regression analysis, the proposed FEM-BV-GEV provides an unbiased estimator for GEV parameters in context of systematically missing information. However, the linearity assumption for the GEV parameters may impose a disadvantage as soon as the influence of covariates on the dynamics of extremes is nonlinear. Considering a set of numerical examples, we will compare the proposed FEM-BV-GEV to the intrinsically nonlinear GEV-CDN methodology, which exploits a conditional density network (CDN) for nonlinear regression analysis based on time-dependent covariates with time-independent (i.e., stationary) neuron weights and biases [8].

4. Implementation

In the following, we discuss the algorithmic implementation of the “FEM-BV-GEV framework”. The FEM-BV-GEV approach was integrated in the existing FEM-BV MATLAB toolbox and can be provided by the authors on email request. The main steps of the general FEM-BV formulation are sketched in [Algorithm 1](#): (1) for different K , C and u_{comb} , a candidate model M is estimated (see [Algorithm 1](#), line 4),

input : Observed series X_t , list of u_{comb} , list of K , list of C
output : Optimal model M^* (u_{comb}^* and $(\Gamma^*(t), \Theta^*)$)

```

1 for  $u_{\text{comb}}$  do
2   for  $K$  list do
3     for  $C$  list do
4       Step 1  $(\Gamma^*(t), \Theta^*) = \text{getOptimalParameterSet}(u_{\text{comb}}, K, C)$ . For fixed  $u_{\text{comb}}$ ,
          $K$  and  $C$  estimate the global optimal parameter set  $(\Gamma^*(t), \Theta^*)$  (compare
         Algorithm 2).
5       Step 2  $M^* = \text{updateOptimalModel}(\Gamma^*(t), \Theta^*, u_{\text{comb}}^*)$ . Estimate the IC value
         according to (22) for every model  $M$ . If the current IC value is smaller than the
         previous one, assign  $M^* = M$ .
```

Algorithm 1: The general FEM-BV algorithm.

and (2) the optimal model M^* , i.e., optimal K^* , C^* and u_{comb}^* , is chosen according to AIC_c in (22) (see Algorithm 1, line 5). Thereby, for a fixed set $\{K, C, u_{\text{comb}}\}$, a model M is obtained by solving (20). The minimization is implemented as a subspace iteration (see Algorithm 2): starting with a randomly initialized $\Gamma(t)$, in an alternating order, we estimate Θ for a fixed $\Gamma(t)$ and then $\Gamma(t)$ for a fixed Θ , thereby obviously reducing in each step the value of (15). The subspace iteration converges to a local optimum. The convergence is achieved if the decrease of the averaged model distance functional (15) is less than a predefined minimization threshold Tol . To obtain the global optimum FEM-BV framework involves an annealing-like strategy: in each annealing step, $\Gamma(t)$ is initialized randomly (for more details on the general FEM-BV annealing-like algorithmic strategy sketched above, see [41]).

The two steps of the subspace iteration are carried out as follows. For a fixed parameter Θ , $\Gamma(t)$ is discretized by the finite element method and estimation of $\Gamma_{\text{opt}}(t)$ results in a linear constrained minimization problem that can be solved using standard numerical tools, e.g., the simplex method [26; 41]. For a fixed $\Gamma(t)$, Θ_{opt} is obtained by minimizing the averaged negative log-likelihood functional (15) with respect to the constraints (16). For minimization, we take advantage of the fact that the averaged model distance functional (15) is uncoupled for different $i = 1, \dots, K$. Thus, Θ_{opt} can be estimated by solving

$$\min_{\theta_i} \sum_{t=1}^T \gamma_i(t) g_{\text{GEV}}(X_t, \theta_i), \quad (25)$$

with respect to constraints (16) for $i = 1, \dots, K$, K times with standard likelihood maximization techniques [13; 19]. Note that the corresponding function in (25) is strongly nonlinear and nonconvex. Additionally, in practical applications, it

input : Observed series X_t , fixed $\{K, C, u_{\text{comb}}\}$, minimization threshold value Tol, number of annealing steps annealing, maximal number of subspace iterations maxSubspace

output : Global optimal parameter set $(\Gamma^*(t), \Theta^*)$

- 1 $\mathcal{L}(\Gamma^*(t), \Theta^*) = \inf$
- 2 **for** $a = 1$: annealing **do**
- 3 $\Gamma_{\text{old}}(t)$ generate randomly with respect to constraints (17)–(19)
- 4 $\Theta_{\text{old}} = \operatorname{argmin}_{\Theta} \mathcal{L}(\Gamma_{\text{old}}(t), \Theta)$
- 5 **while** $|\mathcal{L}(\Gamma_{\text{opt}}(t), \Theta_{\text{opt}}) - \mathcal{L}(\Gamma_{\text{old}}(t), \Theta_{\text{old}})| > \text{Tol}$ *or* maxSubspace **do**
- 6 **Step 1** $\Gamma_{\text{opt}}(t) = \operatorname{argmin} \mathcal{L}(\Gamma(t), \Theta_{\text{old}})$. The constrained minimization with respect to $\Gamma(t)$ results for BV-regularization in a linear problem; standard methods, e.g., simplex method, can be applied.
- 7 **Step 2** $\Theta_{\text{opt}} = \operatorname{argmin} \mathcal{L}(\Gamma_{\text{opt}}(t), \Theta)$. The required numerical optimization method with respect to Θ depends on the model distance function $g(\cdot)$. In FEM-BV-GEV, $g(\cdot)$ is the negative log-likelihood and the minimization is carried out by applying the MCMC method (compare Algorithm 3).
- 8 **if** $\mathcal{L}(\Gamma^*(t), \Theta^*) > \mathcal{L}(\Gamma_{\text{opt}}(t), \Theta_{\text{opt}})$ **then**
- 9 $\Theta^* = \Theta_{\text{opt}}$
- 10 $\Gamma^*(t) = \Gamma_{\text{opt}}(t)$

Algorithm 2: getOptimalParameterSet: annealing and subspace iteration.

may be nondifferentiable (or may exhibit very large values of the first derivative). Because of these reasons, minimization using standard gradient-based methods like Newton’s method and gradient-descent approaches would strongly depend on the initial value and on the boundedness of the first derivatives (e.g., as in the case of the Levenberg–Marquardt optimization algorithm deployed in GEV-CDN [8]). To avoid this difficulty, we consider a gradient-free optimization technique based on the Metropolis algorithm, which is a Markov-chain Monte Carlo (MCMC) method. In particular, we employ the adaptive MCMC methodology proposed in [18], where the adaptive MCMC optimization method considers the Boltzmann distribution as the target density:

$$\pi(\cdot) = \frac{1}{z} \exp(-\beta h(\cdot)) \quad (26)$$

with normalization constant z , inverse temperature parameter β and some energy function $h(\cdot)$. For $\beta \rightarrow \infty$, Boltzmann-distributed samples converge towards the minimal energy of $h(\cdot)$. The adaptivity of the MCMC in [18] comes from adjusting the noise, used for proposing the next sample, and from increasing β (i.e., from annealing). Thus, this approach can be used as an optimization method to get Θ_{opt} for fixed $\Gamma(t)$. For that, we set $h(\Theta) = \mathcal{L}(\Gamma(t), \Theta)$ and modify the MCMC in [18] by adjusting the initialization and the proposed next step (taking into account the

constraints (16) and the dimensionality of Θ). The main steps of the deployed adaptive MCMC are sketched in Algorithm 3 in Appendix A.

We would like to emphasize that in each run of the MCMC algorithm it is sufficient to sample a parameter Θ_{new} that provides a smaller value of $\mathcal{L}(\Gamma(t), \Theta_{\text{new}})$ instead of sampling the whole distribution (refer to Algorithm 3, lines 3 and 9). The subspace iteration deployed by FEM-BV improves in each step the parameters in the sense of minimizing $\mathcal{L}(\Gamma(t), \Theta)$ and provides the optimal parameter set $(\Theta_{\text{opt}}, \Gamma_{\text{opt}})$ for each annealing step. Moreover, MCMC optimization techniques do not depend on the initial start values: the MCMC algorithm also allows us to accept parameters with higher value of the functional (15); thus, there is a chance to obtain the global minima starting from a bad initial value. As will be demonstrated on the numerical examples below in Section 5, the deployed MCMC optimization technique is efficient in terms of computational time.

5. Numerical examples

In this section, we illustrate the proposed FEM-BV-GEV methodology on two test cases and real data. The two test cases are used to investigate the robustness with respect to the systematically missing covariates, the approximation of nonstationary behavior and the computational performance of the framework (with respect to accuracy and computational time). In the real-data example, we analyze a series of block maxima surface temperatures for locations Lugano, Switzerland and Berlin, Germany. In each application, the performance of the FEM-BV-GEV framework is compared to the GEV-CDN approach. GEV-CDN exploits a conditional density network (CDN) for nonlinear regression analysis based on time-dependent covariates with constant weights and biases [8]. The GEV-CDN analysis is performed using the package GEV-CDN provided in the statistical toolbox R [8; 9]. The main tuning parameters of GEV-CDN are the number of hidden neurons (further on denoted by N_H) in the network, the hidden layer transfer function (identity or logistic function) and the number of trials (to avoid the local optima). In all of the numerical examples considered below, an optimal configuration of GEV-CDN with respect to these tuning parameters was determined according to the AIC_c criterion in the way as described in [8].

5.1. Stationary test case. The first example aims to verify the regression analysis of block maxima based only on resolved covariates. We would like to roughly mimic the true underlying dynamics of block maxima in real meteorological applications. Therefore, as covariates, we consider a linear trend, a periodic function with a one-year period and daily averaged measurements of the total solar intensity (TSI) [20; 21].¹ In general, the TSI factor describes the total amount of the solar radiative

¹Data were retrieved from <http://www.pmodwrc.ch/pmod.php?topic=tsi/composite/SolarConstant>.

	Optimal models for stationary test case			
	Settings	NLL	$ M $	AIC_c
FEM-BV-GEV	$K = 3, C = 4$	1.7173×10^3	38	3.5144×10^3
GEV-CDN	$N_H = 12$	2.1116×10^3	75	4.3703×10^3

Table 1. Optimal results for FEM-BV-GEV and GEV-CDN for a stationary test case. The exact negative log-likelihood for X_t using the original parameters is $NLL_{\text{exact}} = 1.7042 \times 10^3$. As described below, smaller values of NLL indicate the models with a better fit, whereas smaller values of AIC_c indicate more informative models.

energy that is hitting the earth's upper atmosphere [21]. However, for this example, we consider only a segment of the TSI measurements (starting from the year 1950) of length $T = 800$, and thus, this factor is only responsible for more fluctuation in the generated block maxima. Now, with covariates $\widehat{U}_t = (u_1(t), u_2(t), u_3(t))$ defined by

$$u_1(t) = \frac{1}{400}t, \quad u_2(t) = \sin\left(\frac{\pi}{2} + \frac{1}{365}2\pi t\right), \quad u_3(t) = \text{TSI}, \quad (27)$$

we generate an artificial series of block maxima using the following parametrization of the GEV model (2):

$$\mu(\widehat{U}_t) = +1 - 5u_1(t) + 2u_2(t) + 1u_3(t), \quad (28)$$

$$\sigma(\widehat{U}_t) = +2.1018 - 0.7132u_1(t) - 0.8203u_2(t) + 0.1356u_3(t), \quad (29)$$

$$\xi(\widehat{U}_t) = -0.0627 - 0.4051u_1(t) + 0.0022u_2(t) - 0.0026u_3(t). \quad (30)$$

By assigning a relatively high coefficient to the factor $u_1(t)$ in (28), we stress the linear-trend behavior in the dynamics of block maxima. The coefficients in (29)–(30) were generated randomly. We use MATLAB function `gevrnd` for sampling:

$$X_t \sim \text{GEV}(\mu(\widehat{U}_t), \sigma(\widehat{U}_t), \xi(\widehat{U}_t)) \quad \text{for } t = 1, \dots, 800. \quad (31)$$

In the next step, we split the covariates \widehat{U}_t into resolved and unresolved subsets $U_t = (u_2(t), u_3(t))$ and $U_t^{\text{un}} = u_1(t)$, respectively, and apply the FEM-BV-GEV and GEV-CDN methods for solving the inverse problem. For given X_t and U_t , we fit the model parameters to describe the distribution of X_t . We want to emphasize that by deliberately missing the most relevant covariate, the linear trend, we would expect both methods to react to this issue by exploiting the intrinsic nonlinearity in the case of GEV-CDN and the nonstationarity in the case of FEM-BV-GEV.

FEM-BV-GEV is supplied with $K_{\text{list}} = \{1, 2, 3\}$, $C_{\text{list}} = \{2 : 1 : 6\}$ and the following configurations: the number of annealing steps is set to 100, the maximal number of the subspace iterations to 150 and the minimization threshold to $\text{Tol} = 5.0 \times 10^{-5}$. The GEV-CDN approach is configured with $N_H = \{1, 2 : 2 : 18\}$, the hidden transfer function is the logistic function and the number of trials is 100. The results are summarized in Table 1, featuring the minimal AIC_c values achieved by the respective

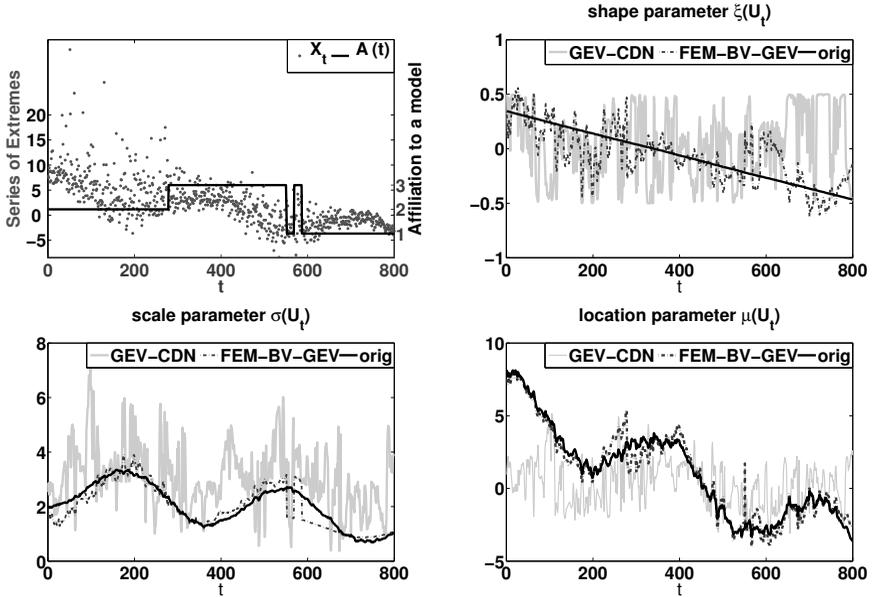


Figure 1. Stationary test case: this figure shows the results for the application of FEM-BV-GEV and GEV-CDN to (31). The upper left figure shows the artificially generated series of extremes X_t versus the optimal switching process $\Gamma^*(t)$, expressed by the affiliation vector $A(t)$. The remaining panels represent the evaluation of the shape, scale and location parameters according to the original (black solid line), optimal FEM-BV-GEV (dashed line) and GEV-CDN (gray solid line) parameters.

methods. Resulting optimal models are $K = 3$ and $C = 4$ for FEM-BV-GEV and $N_H = 12$ for CDN-GEV. The regression analysis of X_t based on resolved covariates was performed better by the FEM-BV-GEV than by the GEV-CDN approach (with a smaller NLL and a lower total number of model parameters). As seen from Figure 1, top left, the optimal switching process $\Gamma^*(t)$, expressed by the affiliation $A(t) \in \mathbb{R}$ (with $A(t) = \{i : i = \arg\max \gamma_i^*(t) \text{ over } i = 1, \dots, K\}$), assigns X_t to three different models. Thereby, it explicitly resolves the implicit linear trend in the systematically missing covariate U^{un} via a switching process that subsequently goes through three local parameter regimes. We cannot compare the original and the resulting coefficients for the regression models explicitly. Instead, we evaluate the approximated $\mu^*(U_t)$, $\xi^*(U_t)$ and $\sigma^*(U_t)$ according to the FEM-BV-GEV and the GEV-CDN models and compare them with the original evaluations according to (28)–(30). The comparison is shown in Figure 1. The top right, bottom left and bottom right panels represent the shape, the scale and the location parameters, respectively. The parameters obtained from FEM-BV-GEV resolve the underlying trend very reliably. In contrast, due to the intrinsic assumption that the neuron weights and biases are constant, GEV-CDN is not able to recover the impact of this missing covariate.

	Optimal models for nonstationary test case			
	Settings	NLL	$ M $	AIC_c
FEM-BV-GEV	$K = 2, C = 12$	1.2041×10^3	37	2.4859×10^3
GEV-CDN	$N_H = 7$	1.2545×10^3	52	2.6203×10^3

Table 2. Optimal results for FEM-BV-GEV ($K = 2$ and $C = 12$) and GEV-CDN ($N_H = 7$) for the nonstationary test case. The exact negative log-likelihood for X_t using the original parameters is $NLL_{\text{exact}} = 1.2289 \times 10^3$.

5.2. Nonstationary test case. Now we consider a nonstationary test case and use it to verify the accuracy and the performance of the FEM-BV-GEV. We generate X_t according to a mixture model with a nonstationary switching process

$$X_t \sim \gamma_1(t) \text{GEV}_1 + \gamma_2(t) \text{GEV}_2, \quad (32)$$

where GEV_1 is parametrized according to (28)–(30) and GEV_2 according to

$$\mu_2(\widehat{U}_t) = -0.5 - 3u_1(t) + 0.5u_2(t) + 0.5u_3(t), \quad (33)$$

$$\sigma_2(\widehat{U}_t) = +0.6729 + 0.0183u_1(t) - 0.4131u_2(t) + 0.1378u_3(t), \quad (34)$$

$$\xi_2(\widehat{U}_t) = -0.0780 - 0.1398u_1(t) - 0.1608u_2(t) + 0.0266u_3(t). \quad (35)$$

Here we consider the same covariates \widehat{U}_t as in the stationary case. The nonstationary switching process $\Gamma(t) = (\gamma_1(t), \gamma_2(t))$ is generated artificially with $C = 6$ switches. Now for given X_t and $U_t = (u_1(t), u_2(t), u_3(t))$, we apply FEM-BV-GEV and the GEV-CDN approach to capture the nonstationarity of (32). FEM-BV-GEV is supplied with $K_{\text{list}} = \{1, 2, 3\}$ and $C_{\text{list}} = \{2 : 1 : 14\}$; remaining configurations are the same as for the stationary test case. Also the configurations of the GEV-CDN approach do not change. Because we provide the full information, $U_t = \widehat{U}_t$, to both methods, they both perform well; compare Table 2 and Figure 2. FEM-BV-GEV approximates the dynamics of X_t with less parameters and a smaller NLL. The inconsistency of the number of switches in $\Gamma^*(t)$ with $C = 12$ (Figure 2 upper left panel) and the original $\Gamma(t)$ with $C = 6$ can be neglected due to the relatively large confidence intervals for $\Gamma^*(t)$ and Θ^* (compare Appendix B, Figure 4 and Table 5).

Also GEV-CDN captures the underlying trend in parameters; compare Figure 2. The computational performance of FEM-BV-GEV and GEV-CDN is compared by considering the CPU time for one annealing step dependent on the increasing number of parameters (configurations do not change). The results are shown in Figure 3. The plots contain the average CPU time over 100 runs. FEM-BV-GEV obviously outperforms the GEV-CDN approach with respect to the computational performance for the growing number of parameters (e.g., corresponding to the larger number of involved covariates or hidden neurons).

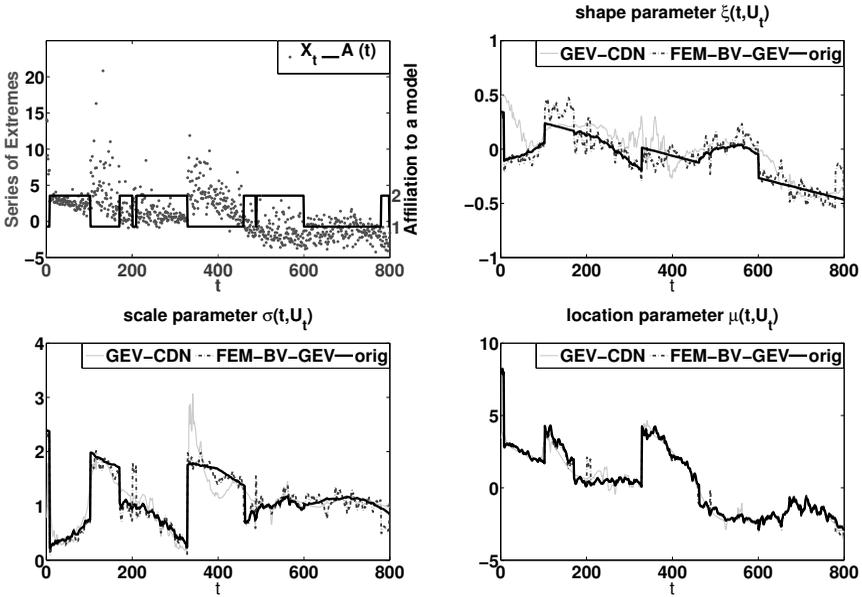


Figure 2. Nonstationary test case: this figure shows the results for the application of FEM-BV-GEV and GEV-CDN to (32). The upper left figure shows the artificial generated series of extremes X_t versus the optimal switching process $\Gamma^*(t)$, expressed by the affiliation vector $A(t)$. The remaining panels represent the evaluation of the shape, scale and location parameters according to original (black solid line), optimal FEM-BV-GEV (dashed line) and GEV-CDN (gray solid line) parameters.

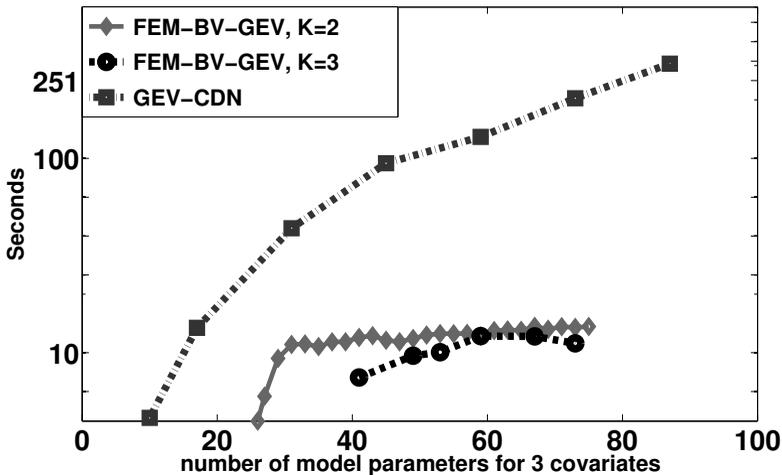


Figure 3. Nonstationary test case: this figure compares the computational time performance of FEM-BV-GEV (diamonds marker for $K = 2$ and circles for $K = 3$) and GEV-CDN (square markers) using a logarithmic time scale (seconds). The number of covariates is fixed; thus, the increase of the number of model parameters is due to increasing of C for FEM-BV-GEV and the number of hidden neurons for GEV-CDN.

5.3. Real-data application. In this section, we apply FEM-BV-GEV and GEV-CDN to real data, where we do not a priori have the knowledge about the underlying dynamics and we have to deal with unresolved modes. In the following, we consider historical daily records of temperature from January 1, 1950 until January 1, 2011 for Lugano, Switzerland (46° N, 8.9667° E) and Berlin, Germany (52.4649° N, 13.3017° E)² [32]. We restrict the data to this period because observations for some of the involved factors are only available starting from 1950. Before extracting 30-day block maxima, we deseasonalize the data. The dedicated series of block maxima for each location contains 742 maxima in the observed period. For the regression analysis, we consider the following set of covariates:

- (1) arctic oscillation (AO),²
- (2) North Atlantic oscillation (NAO),³
- (3) total solar irradiance (TSI), averaged over one day [20; 21],⁴
- (4) ENSO, represented through mean sea surface temperature anomalies in the Nino3.4 region [46],
- (5) $\log(\text{CO}_2)$, with logarithmic dependence according to [43],
- (6) seasonal periodical phase $\text{Per}_I = \sin\left(\frac{1}{365}2\pi t\right)$,
- (7) seasonal periodical phase $\text{Per}_{II} = \sin\left(\frac{3}{2.1}\pi + \frac{1}{365}2\pi t\right)$, and
- (8) Madden–Julian oscillation (MJO) containing the first two empirical orthogonal functions.⁵

The covariates $U_t \in \mathbb{R}^8$ are scaled, with $u_s(t) \in [-1, 1]$ for $s = 1, \dots, 8$, so we can interpret their relative influences on trends in model parameters. For the following GEV regression analysis, we consider the covariates at the same time steps when the maxima in each block are observed. First of all, we want to extract the most significant u_{comb} out of all possible, in total 255, combinations of U_t . For this task, we use the FEM-BV-GEV framework with the following configurations: $K_{\text{list}} = \{1, 2, 3\}$, $C_{\text{list}} = \{5 : 5 : 100\}$, the number of annealing steps is fixed to 100, the number of the subspace iterations is set to 250 and the minimization threshold is set to $\text{Tol} = 5.0 \times 10^{-5}$. Then, according to the minimal AIC_c , we obtain for each location the optimal model including the most significant combination, denoted by u_{comb}^* . For Lugano, u_{comb}^* is $[\text{NAO}, \log(\text{CO}_2), \text{Per}_I, \text{Per}_{II}]$, and for Berlin, $u_{\text{comb}}^* = [\text{AO}, \text{NAO}, \text{Per}_I]$. In the second step, we compare FEM-BV-GEV and GEV-CDN applied to two different settings: (a) we provide the complete set

²Data were retrieved from the NOAA's National Climatic Data Center web page.

³Data were retrieved from <ftp://ftp.cpc.ncep.noaa.gov/cwlinks/>.

⁴Data were retrieved from <http://pmodwrc.ch/pmod.php?topic=tsi/composite/SolarConstant>.

⁵Data were retrieved from <http://cawcr.gov.au/staff/mwheeler/maproom/RMM/>.

	Location Lugano with $u_{\text{comb}}^* = [\text{NAO}, \log(\text{CO}_2), \text{Per}_I, \text{Per}_{II}]$			Location Lugano with $\hat{u}_{\text{comb}}^* = [\text{NAO}, \text{Per}_I, \text{Per}_{II}]$		
	NLL	$ M $	AIC_c	NLL	$ M $	AIC_c
FEM-BV-GEV	1.5739×10^3	70	3.3026×10^3	1.6089×10^3	64	3.3580×10^3
GEV-CDN	1.4940×10^3	115	3.2606×10^3	1.6729×10^3	45	3.4416×10^9

Table 3. Comparison of FEM-BV-GEV and GEV-CDN according to AIC_c model selection criteria for Lugano according to the resolved and unresolved modes. The optimal models for resolved modes are FEM-BV-GEV with $K = 2$ and $C = 40$ and GEV-CDN with $N_H = 14$. The optimal models for unresolved modes are FEM-BV-GEV with $K = 2$ and $C = 40$ and GEV-CDN with $N_H = 6$.

	Location Berlin with $u_{\text{comb}}^* = [\text{AO}, \text{NAO}, \text{Per}_I]$			Location Berlin with $\hat{u}_{\text{comb}}^* = [\text{NAO}, \text{Per}_I]$		
	NLL	$ M $	AIC_c	NLL	$ M $	AIC_c
FEM-BV-GEV	1.6428×10^3	109	3.5415×10^3	1.6756×10^3	89	3.5538×10^3
GEV-CDN	1.7818×10^3	45	3.6595×10^3	1.7927×10^3	39	3.6678×10^3

Table 4. Comparison of FEM-BV-GEV and GEV-CDN according to AIC_c model selection criteria for Berlin according to the resolved and unresolved modes. The optimal models for resolved modes are FEM-BV-GEV with $K = 2$ and $C = 85$ and GEV-CDN with $N_H = 6$. The optimal models for unresolved modes are FEM-BV-GEV with $K = 2$ and $C = 70$ and GEV-CDN with $N_H = 6$.

of optimal covariates for the regression analysis u_{comb}^* , and (b) we provide an incomplete set $\hat{u}_{\text{comb}}^* = [\text{NAO}, \text{Per}_I, \text{Per}_{II}]$ and keep back $\log(\text{CO}_2)$ for Lugano and provide $\hat{u}_{\text{comb}}^* = [\text{NAO}, \text{Per}_I]$ and keep back AO for Berlin. Note that u_{comb}^* is significant according to the FEM-BV-GEV approach and one could argue that for the GEV-CDN approach another set of covariates could be more important.⁶ In return, in real application, we will never know a priori which covariates may be important, and in any case, we do not dispose of complete system measurements. Moreover, the influence of u_{comb}^* on the dynamics of block maxima can be interpreted physically; refer to [Appendix C](#). The results for settings (a) and (b) are shown in [Table 3](#) for Lugano and in [Table 4](#) for Berlin. Thereby, the optimal GEV-CDN model is chosen from $N_H = \{2 : 2 : 16\}$. Additionally, the more interested reader can find a short postinference according to the optimal models in [Appendix C](#): we compute the expectation value of block maxima with the corresponding quantiles for both locations and discuss its behavior.

⁶Application of GEV-CDN to identify the most significant combination of covariates is not feasible because of prohibitively high computational cost to get through all 255 covariates combinations (see [Figure 3](#) for computational-cost comparisons of the two methods).

Comparing the optimal FEM-BV-GEV and GEV-CDN models, we can conclude that in the case when the set of covariates is “complete” the nonlinear GEV-CDN provides a better description of the block maxima for Lugano in terms of information theory (as measured by AIC_c), suggesting that the underlying dynamics is nonlinear rather than nonstationary. In contrast, FEM-BV-GEV provides a better description of block maxima for Berlin. Moreover, in case some information is “missing”, the nonstationary FEM-BV-GEV approach approximates the underlying dynamics better by reflecting the unresolved modes through the switching process for both considered cases (Berlin and Lugano).

6. Conclusion and outlook

In this work, we presented an extension of the GEV methodology for statistical regression analysis of block maxima with systematically missing covariates. We applied the strategy of stochastic covariate reduction and expressed the GEV parameters as fully nonstationary regression models based on resolved covariates only. The involved nonstationarity is interpolated by $K \geq 1$ local models and a nonstationary hidden switching process. The corresponding inverse problem was regularized using the nonparametric FEM-BV methodology by assuming persistence of the switching process (the number of switches between the local models is limited by a parameter C). The well-posed inverse problem is solved by deploying the gradient-free optimization methodology based on the Metropolis algorithm. The selection of optimal K and C and the significant subset of resolved covariates is carried out using the AIC_c information criteria. The proposed FEM-BV-GEV framework allows a computationally efficient, semiparametric and nonstationary analysis and goes beyond strong a priori probabilistic and deterministic assumptions typical for standard approaches deploying, e.g., mixture models, hidden Markov models, spline interpolation or neural networks. FEM-BV-GEV includes methods based on parametric regression, hidden Markov models and local kernel smoothing as special cases. However, the linear regression, which is involved in FEM-BV-GEV and provides an easily interpretable and understandable statistical model, becomes a weakness as soon as the influence of covariates is strongly nonlinear. For that reason, we considered the GEV-CDN approach as a main competitor. GEV-CDN is based on GEV and neural networks: each model parameter is described as a nonlinear function of covariates with constant coefficients exploiting the conditional density network.

We compared the two methods on test cases and real data according to the four criteria: (1) information content of the models, (2) ability to handle unresolved covariates, (3) computational complexity and (4) interpretability of the models. The results in [Section 5.1](#) show that if some relevant information is missing then the nonstationary FEM-BV-GEV approach approximates the underlying dynamics

better by reflecting the unresolved modes through the switching process. In contrast, the GEV-CDN approach seems to average out the underlying trends and in these situations performs worse. The second example (compare [Section 5.2](#)) investigates the performance of the two methods applied to data, which is generated according to a switching process and different models. The FEM-BV-GEV approach performs better. GEV-CDN seems to capture the general dynamics but requires more model parameters to describe the underlying switching process (implicitly resolving the nonstationary switching process through the nonlinear stationary function). The third example (compare [Section 5.3](#)) demonstrates the performance of FEM-BV-GEV and GEV-CDN on real data analyzing 30-day block maxima surface temperatures for locations Lugano, Switzerland and Berlin, Germany. The FEM-BV-GEV approach allows a better description of block maxima for Berlin. Thereby, FEM-BV-GEV indicates two different models ($K = 2$) pointing to systematically missing covariates in the statistical regression analysis. GEV-CDN performs better applied to block maxima for Lugano. The better performance of the nonlinear GEV-CDN approach might lead to the suggestion that the dynamics of the block maxima at Lugano can be better explained by the stationary nonlinear regression than by the nonstationary linear one. Additionally, FEM-BV-GEV outperforms GEV-CDN in the interpretability and understandability of the models and in the far more favorable computational complexity and scalability. Consequently, we conclude that the FEM-BV-GEV approach should be used in cases where not all potentially significant covariates can be resolved explicitly and the observed data (series of extremes and the number of resolved covariates) is big; correspondingly, GEV-CDN should be applied in cases where the complete information is known and the provided statistics are rather small.

Outlook. A point of interest in data-based analysis of block maxima in the context of the FEM-BV-GEV approach is to understand the dynamics of the switching process, implicitly reflecting the dynamics of the most relevant covariates that are systematically missing in the analyzed data. One can either try to find a set of covariates to resolve the observed dynamics or parametrize the switching process. The latter can be done by considering the switching process as a discrete time series and study the dynamics with time-series analysis methods, e.g., a FEM-BV-Markov method [27]. Another extension of the presented FEM-BV-GEV methodology goes toward space-time modeling of block maxima. The spatial extension of FEM-BV-GEV requires appropriate regularization in space, e.g., based on distances between the locations where the measurements are taken. Besides that, by replacing GEV by the generalized Pareto distribution (GPD) methodology, we can straightforwardly derive the FEM-BV-GPD framework to study threshold exceedances. All these issues are matters of future research.

```

input :  $X_t$  series of extremes,  $u_{\text{comb}}, \Gamma(t), \mathcal{L}(\Gamma, \Theta_{\text{old}})$ 
output :  $\Theta_{\text{opt}}$ 
1  $\Theta_{\text{new}} = \text{generateInitialValue}(\Gamma(t), X_t, u_{\text{comb}})$ 
2 if  $\mathcal{L}(\Gamma(t), \Theta_{\text{new}}) < \mathcal{L}(\Gamma(t), \Theta_{\text{old}})$  then
3      $\Theta_{\text{opt}} = \Theta_{\text{new}}$ 
4     return  $\Theta_{\text{opt}}$ 
5 initialize  $\delta, \beta, \Sigma, \text{counterAccept} = 0$ 
6 for  $\text{sampleStep} = 1 : \text{sampleSizeMCMC}$  do
7      $\Theta_{\text{next}} = \text{proposeNext}(\Theta_{\text{new}}, \Gamma(t), X_t, u_{\text{comb}}, \Sigma, \text{noise}, \beta)$ 
8     if  $\mathcal{L}(\Gamma(t), \Theta_{\text{next}}) < \mathcal{L}(\Gamma(t), \Theta_{\text{old}})$  then
9          $\Theta_{\text{opt}} = \Theta_{\text{next}}$ 
10        return  $\Theta_{\text{opt}}$ 
11    else if  $\text{checkAcceptance}(\beta, \Theta_{\text{next}}, \Theta_{\text{new}})$  then
12         $\Theta_{\text{new}} = \Theta_{\text{next}}$ 
13         $\text{counterAccept} += 1$ 
14         $\text{updateCovMatrix}(\Theta_{\text{new}}, \Sigma)$ 
15    if  $\text{sampleStep} \geq 50$  then
16         $[\delta, \beta] = \text{adaptStep}(\delta, \beta, \text{counterAccept}, \text{sampleStep})$ 
17  $\Theta_{\text{opt}} = \Theta_{\text{old}}$ 
    
```

Algorithm 3: MCMC-based optimization algorithm for fixed $\Gamma(t)$.

Appendix A: Details of the adaptive MCMC algorithm

In the following, we point out the main steps of the deployed MCMC-based optimization; see [Algorithm 3](#). The algorithm is based on [18] and differs mainly in two steps: lines 3 and 6 (explained in more details in the next two sections). Please note that the convergence condition for this algorithm is fulfilled if MCMC proposes a new parameter set that provides a smaller $\mathcal{L}(\Gamma, \Theta_{\text{new}})$ value for fixed $\Gamma(t)$. In practical applications, in the beginning of the FEM-BV subspace-minimization procedure, [Algorithm 3](#) proposes a better parameter set already after few steps. However, the number of samplings is limited by the parameter `sampleSizeMCMC`; see [Algorithm 3](#), line 5. In particular, it is recommended to limit the number of samples because as soon as we get into the area of the local optima it becomes hard to propose a better parameter set. And if the algorithm fails, meaning it does not provide a better set of parameters, it returns with $\Theta_{\text{opt}} = \Theta_{\text{old}}$; see [Algorithm 3](#), line 16. For the applications presented in this work, this parameter was assigned to `sampleSizeMCMC = 1000`.

A.1. Generate an initial value. MCMC sampling starts with generating an initial value (we refer to [Algorithm 3](#), line 3). The scale and the shape parameters have to

fulfill the constraints

$$0 < \sigma_i(U_t) = \sigma_i^{(0)} + \sum_{s=1}^S \sigma_i^{(s)} u_s(t) < \text{const}, \quad i = 1, \dots, K, \quad \forall t, \quad (1)$$

$$-0.5 < \xi_i(U_t) = \xi_i^{(0)} + \sum_{s=1}^S \xi_i^{(s)} u_s(t) < 0.5, \quad i = 1, \dots, K, \quad \forall t \quad (2)$$

(constraint (2) ensures a regular likelihood estimator [14]). Applying a simple uniform distribution would not necessarily provide an appropriate initial value. To hold the constraints, we reformulate them: since $\xi_i(U_t)$ and $\sigma_i(U_t)$ attain their unique maximum/minimum values in one of the corners of the convex hull defined by U_t , $t = 1, \dots, T$ [27], it is sufficient to fulfill the constraints (1)–(2) on all corners of the convex hull of U_t . Using a matrix $A \in \mathbb{R}^{(S+1) \times 2^S}$ that contains all combinations of maximal/minimal values of $U(t)$, $t = 1, \dots, T$, we can reformulate the constraint for $\xi_i = (\xi_i^{(0)}, \dots, \xi_i^{(S)})$:

$$\begin{aligned} -A\xi_i &< -lb_\xi, & lb_\xi &= -0.5 \cdot \mathbf{1} \in \mathbb{R}^{2^S}, \\ A\xi_i &< +ub_\xi, & ub_\xi &= 0.5 \cdot \mathbf{1} \in \mathbb{R}^{2^S}. \end{aligned}$$

The same applies for σ . Finally, if we slightly strengthen the constraints

$$\sigma_i(U_t) \in [\epsilon, \text{const}] \quad \text{and} \quad \xi_i(U_t) \in [-0.5 + \epsilon, 0.5 - \epsilon]$$

with $\epsilon > 0$ small and $\text{const} \in \mathbb{R}$ some high value, we can use some convex sampler to get random, uniformly distributed values within this convex hull. The same approach can be applied to sample $\mu_i = (\mu_i^{(0)}, \dots, \mu_i^{(S)})$ in a way such that the constraint (16) in Section 2 is fulfilled. Another way is to estimate the initial value for μ_i by applying ordinary least squares [14]. Note that this estimation is not considered as the trend estimate for the GEV distribution but as a procedure to generate an initial value that is adjusted within the MCMC and the subspace (Algorithm 2 in Section 4) procedure. Both possibilities are implemented in the FEM-BV-GEV framework. In this paper, the second one was deployed.

A.2. Propose next. The performance of the Metropolis algorithm can be improved with an appropriate proposal distribution [4; 34]. However, it is not obvious which proposal density should be chosen for the current target density. In this work, we refer to the discussions in [4] and deploy the adaptive Metropolis algorithm where the next proposal, denoted here by Y_{n+1} , is sampled according to a mixture distribution with respect to the information of all previous accepted samples, denoted here by X_0, \dots, X_n :

$$Y_{n+1} \sim (1 - \delta) \mathcal{N}\left(X_n, \frac{2.38^2}{d} \Sigma_n\right) + \delta \mathcal{N}(X_n, \Sigma_0), \quad (3)$$

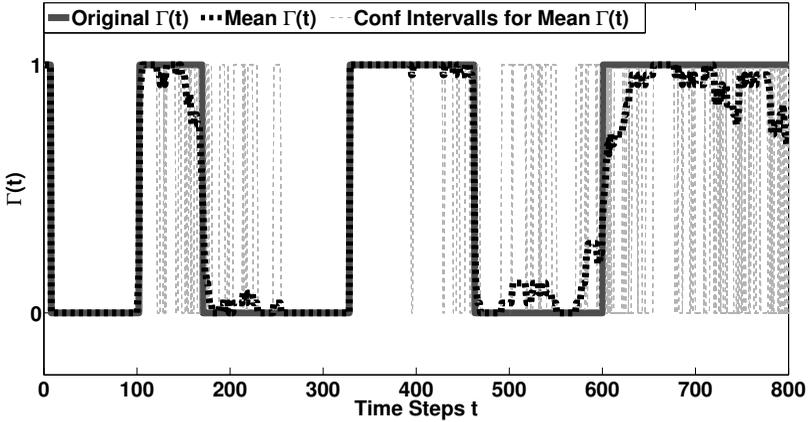


Figure 4. Nonstationary test case: the figure compares the averaged optimal $\widehat{\Gamma}^*(t)$ (black dashed line) with its confidence intervals (gray dashed lines) versus the original switching process (gray solid line).

where d is the dimension of X_n and $\Sigma_n \in \mathbb{R}^{d \times d}$ corresponds to the empirical covariance matrix of X_0, \dots, X_n . The parameter $0 < \delta < 1$ controls the acceptance rate of the Metropolis algorithm; the acceptance rate is increasing for $\delta \rightarrow 1$ and decreasing for $\delta \rightarrow 0$. For more details on the adaption step, see [18].

Appendix B: Confidence intervals

In the following, we refer to the nonstationary test case in Section 5 and provide the confidence intervals for the optimal estimation of $(\Gamma^*(t), \Theta^*)$ to verify the accuracy of the proposed FEM-BV-GEV approach. We obtain the confidence intervals via bootstrapping procedure [12]; i.e., we resample $X_t, t = 1, \dots, T$, according to (32) in Section 5 N times and apply FEM-BV-GEV (with $K_{\text{list}} = \{1, 2, 3\}$ and $C_{\text{list}} = \{2 : 1 : 14\}$) each time. Then each optimal result $(\Gamma^*(t), \Theta^*)$ is stored, and we can estimate the averaged parameters as well as the confidence intervals. For this example, we consider $N = 150$. Figure 4 shows the original $\Gamma(t)$ (gray solid line), the averaged optimal $\widehat{\Gamma}^*(t)$ (black dashed line) and all other realizations of $\Gamma^*(t)$ that can be considered as the confidence intervals (gray dashed lines). Table 5 contains the corresponding confidence intervals for the averaged optimal model parameters $\hat{\theta}_i^*$ for $i = 1, 2$.

Appendix C: Postinference

In this section, we discuss the postinference for Lugano and Berlin according to the optimal FEM-BV-GEV and GEV-CDN models. The locally linear FEM-BV-GEV model allows direct interpretation of the influence of covariates on the dynamics of GEV parameters; see Table 6 for Lugano and Table 7 for Berlin. For

	Model Parameters							
	μ_0	μ_1	μ_2	μ_3	σ_0	σ_1	σ_2	σ_3
					ξ_0	ξ_1	ξ_2	ξ_3
θ_1	1.0000	-5.0000	2.0000	-1.0000	1.4983	-0.6987	0.1937	0.0353
					-0.0627	-0.4051	0.0022	-0.0026
$\hat{\theta}_1^*$	1.0502	-4.9507	1.9786	-1.0385	1.4821	-0.6654	0.1888	0.0297
					-0.0719	-0.4490	0.0421	-0.0355
std±	0.1478	0.1690	0.2492	0.3418	0.1085	0.1652	0.0979	0.2109
					0.0500	0.0550	0.0446	0.1051
θ_2	-0.5000	-3.0000	0.5000	0.5000	0.6729	0.0183	-0.4131	0.1378
					-0.0780	-0.1398	-0.1608	0.0266
$\hat{\theta}_2^*$	-0.4938	-2.9626	0.5342	0.5300	0.7187	0.1113	-0.3720	0.1944
					-0.0852	-0.1472	-0.1542	0.0122
std±	0.0715	0.1427	0.0897	0.1066	0.0573	0.1108	0.0645	0.0740
					0.0684	0.1219	0.1175	0.1537

Table 5. The original parameters θ_1 and θ_2 according to (33)–(35) and (28)–(30) in Section 5, averaged optimal parameters $\hat{\theta}_1^*$ and $\hat{\theta}_2^*$ and the corresponding standard deviations (std±).

	Model Parameters for Lugano with $u_{\text{comb}}^* = [\text{NAO}, \log(\text{CO}_2), \text{Per}_I, \text{Per}_{II}]$									
	μ_0	μ_1	μ_2	μ_3	μ_4	σ_0	σ_1	σ_2	σ_3	σ_4
						ξ_0	ξ_1	ξ_2	ξ_3	ξ_4
θ_1^*	3.92	0.78	-2.12	1.70	-0.34	1.71	0.60	-0.17	0.39	-0.42
						-0.05	0.16	0.03	-0.19	-0.09
θ_2^*	4.29	-0.19	1.97	0.74	-1.39	1.99	-0.10	0.05	0.21	-0.61
						-0.37	0.39	0.40	-0.15	-0.09

Table 6. The table contains optimal parameters θ_1^* and θ_2^* (the values are rounded to two decimal places).

	Model Parameters for Berlin with $u_{\text{comb}}^* = [\text{AO}, \text{NAO}, \text{Per}_I]$								
	μ_0	μ_1	μ_2	μ_3	σ_0	σ_1	σ_2	σ_3	
					ξ_0	ξ_1	ξ_2	ξ_3	
θ_1^*	4.73	1.89	0.1	0.38	2.59	-0.52	-0.03	0.6873	
					-0.22	-0.27	-0.40	0.17	
θ_2^*	8.43	2.00	-1.13	0.59	2.15	-0.21	0.69	-0.12	
					-0.32	0.01	-0.10	0.00	

Table 7. The table contains optimal parameters θ_1^* and θ_2^* (the values are rounded to two decimal places).

the neural-networks-based GEV-CDN approach, we obtain a matrix of weights and have to evaluate the parameters according to the transfer function (a logistic function in our particular case), making the fitted models not easy for interpretation and understanding. The identification of these factors is physically meaningful. Positive phase of AO causes dry and hot conditions in Mediterranean regions. AO has a direct influence on atmospheric circulation blocking events: it induces a ridge of high pressure in the mid-latitude jet streams that can cause persistently high temperatures (as well as cold conditions) [23]. Positive phases of NAO cause warm, wet winters in northern and dry winters in southern Europe. Due to the anthropogenic influence of CO₂ concentration, log(CO₂) holds a positive trend with oscillating dynamics (with maximum value in May and minimum in October) [31]. The relevance of Per_I and Per_{II} points to strong seasonal dependence of block maxima in both locations (this is obvious since we consider monthly maxima). In order to study the long-term trend in distribution of block maxima, we evaluate the nonstationary expectation value

$$\mathbb{E}_{K=2}[X_t, t] = \sum_{i=1}^2 \gamma_i(t) \left(\mu_i(U_t) + \sigma_i(U_t) \frac{\tilde{\Gamma}(1 - \xi_i(U_t)) - 1}{\xi_i(U_t)} \right), \tag{1}$$

$$\mathbb{E}_{\text{CDN}}[X_t, t] = \mu_{\text{CDN}}(U_t) + \sigma_{\text{CDN}}(U_t) \frac{\tilde{\Gamma}(1 - \xi_{\text{CDN}}(U_t)) - 1}{\xi_{\text{CDN}}(U_t)} \tag{2}$$

with $t = 1, \dots, 742$, where $K = 2$ corresponds to FEM-BV-GEV (with parametrization according to (13) in Section 2) and CDN to GEV-CDN models and $\tilde{\Gamma}$ denotes the gamma function. Figures 5 and 7 show the results according to FEM-BV-GEV and Figures 6 and 8 according to GEV-CDN. The 0.99- and 0.10-quantiles are the

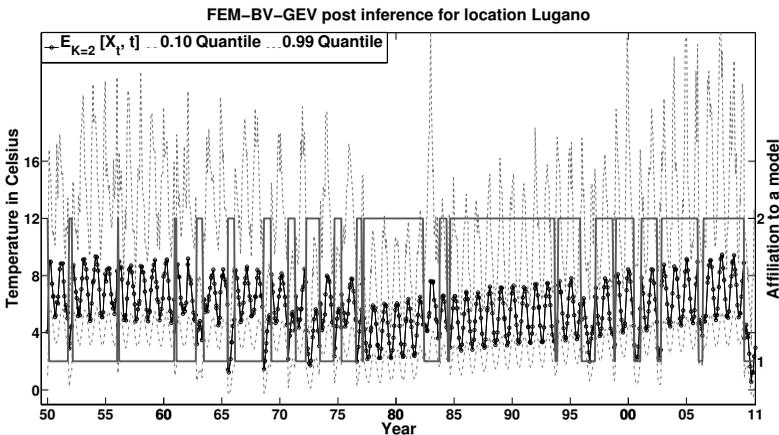


Figure 5. Location Lugano: the plot of the expectation value for the optimal FEM-BV-GEV model, $K = 2$ and $C = 40$.

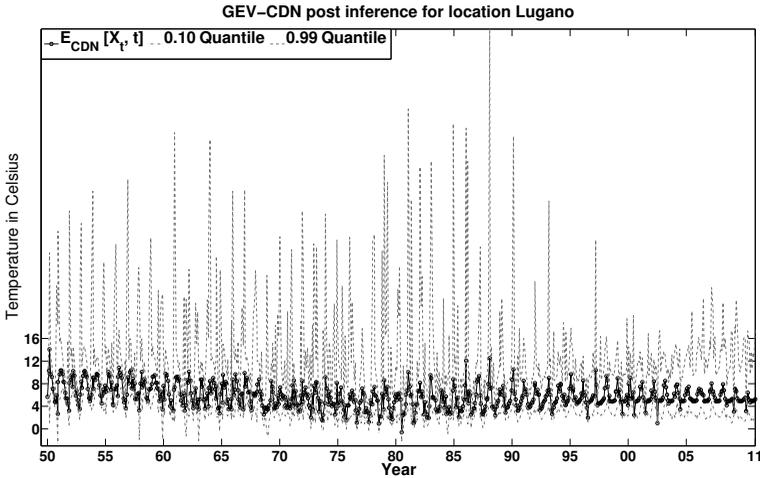


Figure 6. Location Lugano: the plot of the expectation value for the optimal GEV-CDN model with $N_H = 14$.

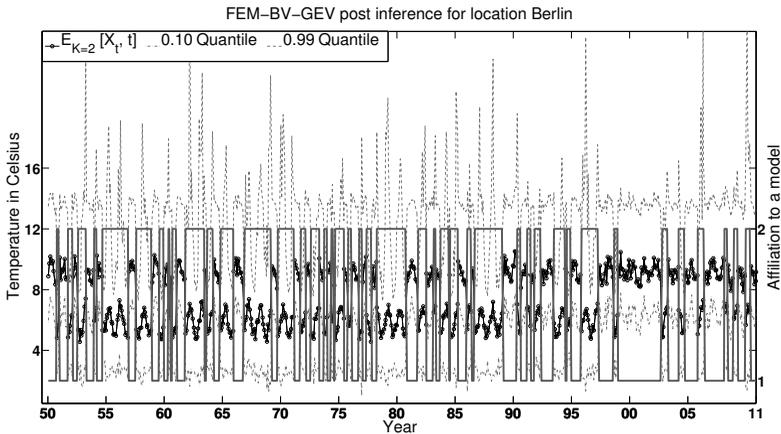


Figure 7. Location Berlin: the plot of the expectation value for the optimal FEM-BV-GEV model, $K = 2$ and $C = 85$.

confidence intervals containing 89% of the distribution. In particular, the 0.99-quantile corresponds to the 100-year return level. According to the FEM-BV-GEV results, the mean for Lugano shows a slightly negative trend in the first model, but after the 1980s, the second model dominates, where $\log(\text{CO}_2)$ has a positive influence and so the trend in block maxima becomes positive. In contrast, according to the GEV-CDN model, there is no obvious trend; however, the confidence intervals for the GEV distribution increase in the last ten years. For Berlin, the trend of the expectation value is separated according to two FEM-BV-GEV models, one model corresponds to higher block maxima. The GEV-CDN model averages these dynamics and provides a unchanging behavior with some outliers.

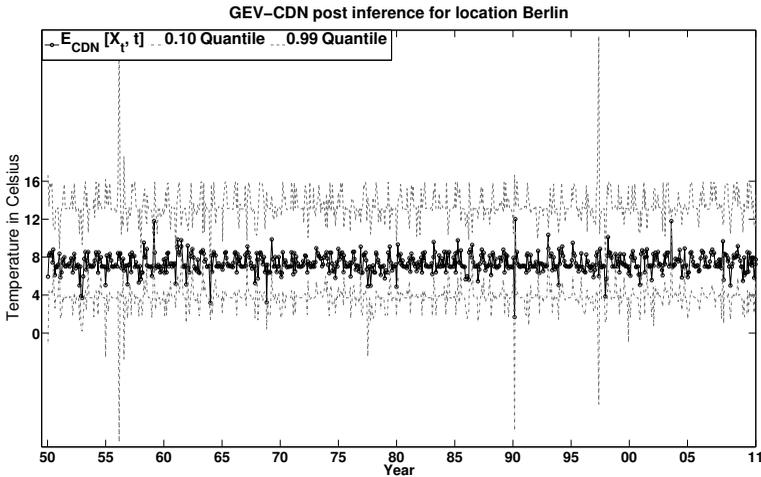


Figure 8. Location Berlin: the plots of the expectation value for the optimal GEV-CDN model, $N_H = 6$.

Acknowledgments

The authors would like to thank the staff and the organizers of the IPAM program “Model and Data Hierarchies for Simulating and Understanding Climate”, March 8–June 11, 2010. Important input regarding the understanding of standard methods to study the dynamics of extremes is due to other participants of the program (special thanks to Richard L. Smith for his talk that motivated the problem formulation considered in this manuscript; we would also like to thank E. Gerber for helpful hints regarding the $\log(\text{CO}_2)$ impact on temperature). We also thank Harald von Waldow and Olivia Romppainen-Martius for a helpful discussion.

References

- [1] B. Betrò, A. Bodini, and Q. A. Cossu, *Using a hidden Markov model to analyse extreme rainfall events in central-east Sardinia*, *Environmetrics* **19** (2008), no. 7, 702–713. [MR 2530393](#)
- [2] ———, *Regional-scale analysis of extreme rainfalls via HMM*, presented at the V International Workshop on Spatio-Temporal Modelling, 2010.
- [3] M.-O. Boldi and A. C. Davison, *A mixture model for multivariate extremes*, *J. R. Stat. Soc., Ser. B, Stat. Methodol.* **69** (2007), no. 2, 217–229. [MR 2325273](#) [Zbl 1120.62030](#)
- [4] S. Brooks, A. Gelman, G. L. Jones, and X.-L. Meng (eds.), *Handbook of Markov chain Monte Carlo*, CRC Press, Boca Raton, FL, 2011. [MR 2012g:62004](#) [Zbl 1218.65001](#)
- [5] K. P. Burnham and D. R. Anderson, *Model selection and multimodel inference: a practical information-theoretic approach*, 2nd ed., Springer, New York, 2002. [MR 1919620](#) [Zbl 1005.62007](#)
- [6] A. C. Cameron and P. K. Trivedi, *Regression analysis of count data*, 2nd ed., *Econometric Society Monographs*, no. 53, Cambridge University Press, 2013. [MR 3155491](#)

- [7] E. J. Candès, J. K. Romberg, and T. Tao, *Stable signal recovery from incomplete and inaccurate measurements*, Comm. Pure Appl. Math. **59** (2006), no. 8, 1207–1223. MR 2007f:94007 Zbl 1098.94009
- [8] A. J. Cannon, *A flexible nonlinear modelling framework for nonstationary generalized extreme value analysis in hydroclimatology*, Hydrol. Process. **24** (2010), no. 24, 673–685.
- [9] ———, *GEVcdn: an R package for nonstationary extreme value analysis by generalized extreme value conditional density estimation network*, Comput. Geosci. **37** (2011), 1532–1533.
- [10] V. Chavez-Demoulin and A. C. Davison, *Generalized additive modelling of sample extremes*, J. R. Stat. Soc., Ser. C, Appl. Stat. **54** (2005), no. 1, 207–222. MR 2134607 Zbl 05188681
- [11] A. J. Chorin and O. H. Hald, *Stochastic tools in mathematics and science*, Surveys and Tutorials in the Applied Mathematical Sciences, no. 1, Springer, New York, 2006. MR 2006j:60001 Zbl 1086.60001
- [12] B. Clarke, E. Fokoué, and H. H. Zhang, *Principles and theory for data mining and machine learning*, Springer, Dordrecht, 2009. MR 2012i:62008 Zbl 1179.62004
- [13] S. G. Coles, *An introduction to statistical modeling of extreme values*, Springer, London, 2001. MR 2003h:62002 Zbl 0980.62043
- [14] S. G. Coles and M. J. Dixon, *Likelihood-based inference for extreme value models*, Extremes **2** (1999), no. 1, 5–23. Zbl 0938.62013
- [15] S. G. Coles and E. A. Powell, *Bayesian methods in extreme value modelling: a review and new developments*, Int. Stat. Rev. **64** (1996), no. 1, 119–136. Zbl 0853.62025
- [16] A. C. Davison and N. I. Ramesh, *Local likelihood smoothing of sample extremes*, J. R. Stat. Soc., Ser. B, Stat. Methodol. **62** (2000), no. 1, 191–208. MR 1747404 Zbl 0942.62058
- [17] L. de Haan and A. Ferreira, *Extreme value theory: an introduction*, Springer, New York, 2006. MR 2007g:62008 Zbl 1101.62002
- [18] J. de Wiljes, A. Majda, and I. Horenko, *An adaptive Markov chain Monte Carlo approach to time series clustering of processes with regime transition behavior*, Multiscale Model. Simul. **11** (2013), no. 2, 415–441. MR 3047436
- [19] P. Embrechts, C. Klüppelberg, and T. Mikosch, *Modelling extremal events for insurance and finance*, 8th ed., Applications of Mathematics, no. 33, Springer, Berlin, 1997. MR 98k:60080 Zbl 0873.62116
- [20] C. Fröhlich, *Observations of irradiance variations*, Space Sci. Rev. **94** (2000), no. 1–2, 15–24.
- [21] ———, *Solar irradiance variability since 1978: revision of the PMOD composite during solar cycle 21*, Space Sci. Rev. **125** (2006), no. 1–4, 53–65.
- [22] J. Hadamard, *Sur les problèmes aux dérivées partielles et leur signification physique*, Princeton University Bulletin **13** (1902), 49–52.
- [23] S. Häkkinen, P. B. Rhines, and D. L. Worthen, *Atmospheric blocking and Atlantic multidecadal ocean variability*, Science **334** (2011), no. 6056, 665–659.
- [24] T. Hastie, R. Tibshirani, and J. Friedman, *The elements of statistical learning: data mining, inference, and prediction*, 2nd ed., Springer, New York, 2009. MR 2012d:62081 Zbl 1273.62005
- [25] I. Horenko, *Finite element approach to clustering of multidimensional time series*, SIAM J. Sci. Comput. **32** (2010), no. 1, 62–83. MR 2011b:62009 Zbl 1206.62150
- [26] ———, *On identification of nonstationary factor models and their application to atmospheric data analysis*, J. Atmos. Sci. **67** (2010), no. 5, 1559–1574.
- [27] ———, *Nonstationarity in multifactor models of discrete jump processes, memory and application to cloud modeling*, J. Atmos. Sci. **68** (2011), no. 7, 1493–1506.

- [28] ———, *On analysis of nonstationary categorical data time series: dynamical dimension reduction, model selection, and applications to computational sociology*, *Multiscale Model. Simul.* **9** (2011), no. 4, 1700–1726. MR 2012j:60196 Zbl 1244.60070
- [29] I. Horenko and C. Schütte, *On metastable conformational analysis of nonequilibrium biomolecular time series*, *Multiscale Model. Simul.* **8** (2010), no. 2, 701–716. MR 2011a:62008 Zbl 05719784
- [30] C. M. Hurvich and C.-L. Tsai, *Regression and time series model selection in small samples*, *Biometrika* **76** (1989), no. 2, 297–307. MR 91c:62083 Zbl 0669.62085
- [31] R. P. Kane and E. R. de Paula, *Atmospheric CO₂ changes at Mahuna Loa, Hawaii*, *J. Atmos. Terr. Phys.* **58** (1996), no. 15, 1673–1681.
- [32] A. M. G. Klein Tank, J. B. Wijngaard, G. P. Können, R. Böhm, G. Demarée, A. Gocheva, M. Miletta, S. Pashiardis, L. Hejkrlik, C. Kern-Hansen, R. Heino, P. Bessemoulin, G. Müller-Westermeier, M. Tzanakou, S. Szalai, T. Pálsdóttir, D. Fitzgerald, S. Rubin, M. Capaldo, M. Maugeri, A. Leitass, A. Bukantis, R. Aberfeld, A. F. V. van Engelen, E. Forland, M. Mietus, F. Coelho, C. Mares, V. Razuvaev, E. Nieplova, T. Cegnar, J. Antonio López, B. Dahlström, A. Moberg, W. Kirchhofer, A. Ceylan, O. Pachaliuk, L. V. Alexander, and P. Petrovic, *Daily dataset of 20th-century surface air temperature and precipitation series for the European Climate Assessment*, *Int. J. Climatol.* **22** (2002), no. 12, 1441–1453.
- [33] W. Kozek, *Optimally Karhunen–Loève-like STFT expansion of nonstationary processes*, *IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 4, IEEE, Piscataway, NJ, 1993, pp. 428–431.
- [34] F. Liang, C. Liu, and R. J. Carroll, *Advanced Markov chain Monte Carlo methods: learning from past samples*, Wiley, Chichester, UK, 2010. MR 2012m:60172 Zbl 1209.62009
- [35] A. R. Lima, A. J. Cannon, and W. W. Hsieh, *Nonlinear regression in environmental sciences by support vector machines combined with evolutionary strategy*, *Comput. Geosci.* **50** (2013), 136–144.
- [36] J. W. Lindeberg, *Eine neue Herleitung des Exponentialgesetzes in der Wahrscheinlichkeitsrechnung*, *Math. Z.* **15** (1922), no. 1, 211–225. MR 1544569 JFM 48.0602.04
- [37] M. Loève, *Probability theory, II*, 4th ed., Graduate Texts in Mathematics, no. 46, Springer, New York, 1978. MR 0651018 Zbl 0385.60001
- [38] A. MacDonald, C. J. Scarratt, D. Lee, B. Darlow, M. Reale, and G. Russell, *A flexible extreme value mixture model*, *Comput. Statist. Data Anal.* **55** (2011), no. 6, 2137–2157. MR 2785120
- [39] A. J. Majda, R. V. Abramov, and M. J. Grote, *Information theory and stochasticity for multiscale nonlinear systems*, CRM Monograph Series, no. 25, American Mathematical Society, Providence, RI, 2005. MR 2006k:76110 Zbl 1082.60002
- [40] E. Meerbach, E. Dittmer, I. Horenko, and C. Schütte, *Multiscale modelling in molecular dynamics: biomolecular conformations as metastable states*, *Computer simulations in condensed matter systems: from materials to chemical biology, I* (M. Ferrario, G. Ciccotti, and K. Binder, eds.), Lecture Notes in Physics, no. 703, Springer, Berlin, 2006, pp. 495–517.
- [41] P. Metzner, L. Putzig, and I. Horenko, *Analysis of persistent nonstationary time series and applications*, *Commun. Appl. Math. Comput. Sci.* **7** (2012), no. 2, 175–229. MR 3005737 Zbl 1275.62067
- [42] S. E. Neville, M. J. Palmer, and M. P. Wand, *Generalized extreme value additive model analysis via mean field variational Bayes*, *Aust. N. Z. J. Stat.* **53** (2011), no. 3, 305–330. MR 2897373
- [43] R. T. Pierrehumbert, *Energy balance models*, 2001 Program in Geophysical Fluid Dynamics (J.-L. Thiffeault, ed.), Woods Hole Oceanographic Institution, Woods Holes, MA, 2001, pp. 72–87.

- [44] R. Tibshirani, *Regression shrinkage and selection via the lasso*, J. R. Stat. Soc., Ser. B, Stat. Methodol. **58** (1996), no. 1, 267–288. [MR 96j:62134](#) [Zbl 0850.62538](#)
- [45] A. N. Tikhonov, *On the solution of ill-posed problems and the method of regularization*, Dokl. Akad. Nauk SSSR **151** (1963), 501–504, In Russian; translated in *Sov. Math., Dokl.* **4** (1963), 1035–1038. [MR 28 #5576](#) [Zbl 03227378](#)
- [46] K. E. Trenberth, *The definition of El Niño*, B. Am. Meteorol. Soc. **78** (1997), 2771–2777.
- [47] V. N. Vapnik, *The nature of statistical learning theory*, Springer, New York, 1995. [MR 98a:68159](#) [Zbl 0833.62008](#)
- [48] G. Wahba, *Spline models for observational data*, CBMS-NSF Regional Conference Series in Applied Mathematics, no. 59, SIAM, Philadelphia, PA, 1990. [MR 91g:62028](#) [Zbl 0813.62001](#)

Received May 20, 2013. Revised November 28, 2013.

OLGA KAISER: olga.kaiser@usi.ch

Institute of Computational Science, Università della Svizzera italiana, Via Giuseppe Buffi 13, CH-6904 Lugano, Switzerland

ILLIA HORENKO: illia.horenko@usi.ch

Institute of Computational Science, Università della Svizzera italiana, Via Giuseppe Buffi 13, CH-6904 Lugano, Switzerland

Guidelines for Authors

Authors may submit manuscripts in PDF format on-line at the Submission page at msp.berkeley.edu/camcos.

Originality. Submission of a manuscript acknowledges that the manuscript is original and is not, in whole or in part, published or under consideration for publication elsewhere. It is understood also that the manuscript will not be submitted elsewhere while under consideration for publication in this journal.

Language. Articles in CAMCoS are usually in English, but articles written in other languages are welcome.

Required items. A brief abstract of about 150 words or less must be included. It should be self-contained and not make any reference to the bibliography. If the article is not in English, two versions of the abstract must be included, one in the language of the article and one in English. Also required are keywords and subject classifications for the article, and, for each author, postal address, affiliation (if appropriate), and email address.

Format. Authors are encouraged to use L^AT_EX but submissions in other varieties of T_EX, and exceptionally in other formats, are acceptable. Initial uploads should be in PDF format; after the refereeing process we will ask you to submit all source material.

References. Bibliographical references should be complete, including article titles and page ranges. All references in the bibliography should be cited in the text. The use of BibT_EX is preferred but not required. Tags will be converted to the house format, however, for submission you may use the format of your choice. Links will be provided to all literature with known web locations and authors are encouraged to provide their own links in addition to those supplied in the editorial process.

Figures. Figures must be of publication quality. After acceptance, you will need to submit the original source files in vector graphics format for all diagrams in your manuscript: vector EPS or vector PDF files are the most useful.

Most drawing and graphing packages (Mathematica, Adobe Illustrator, Corel Draw, MATLAB, etc.) allow the user to save files in one of these formats. Make sure that what you are saving is vector graphics and not a bitmap. If you need help, please write to graphics@msp.org with details about how your graphics were generated.

White space. Forced line breaks or page breaks should not be inserted in the document. There is no point in your trying to optimize line and page breaks in the original manuscript. The manuscript will be reformatted to use the journal's preferred fonts and layout.

Proofs. Page proofs will be made available to authors (or to the designated corresponding author) at a Web site in PDF format. Failure to acknowledge the receipt of proofs or to return corrections within the requested deadline may cause publication to be postponed.

Communications in Applied Mathematics and Computational Science

vol. 9

no. 1

2014

- Discrete nonhomogeneous and nonstationary logistic and Markov regression models for spatiotemporal data with unresolved external influences 1
JANA DE WILJES, LARS PUTZIG and ILLIA HORENKO
- Low Mach number fluctuating hydrodynamics of diffusively mixing fluids 47
ALEKSANDAR DONEV, ANDY NONAKA, YIFEI SUN, THOMAS G. FAI,
ALEJANDRO L. GARCIA AND JOHN B. BELL
- High-order methods for computing distances to implicitly defined surfaces 107
ROBERT I. SAYE
- On inference of statistical regression models for extreme events based on incomplete observation data 143
OLGA KAISER AND ILLIA HORENKO