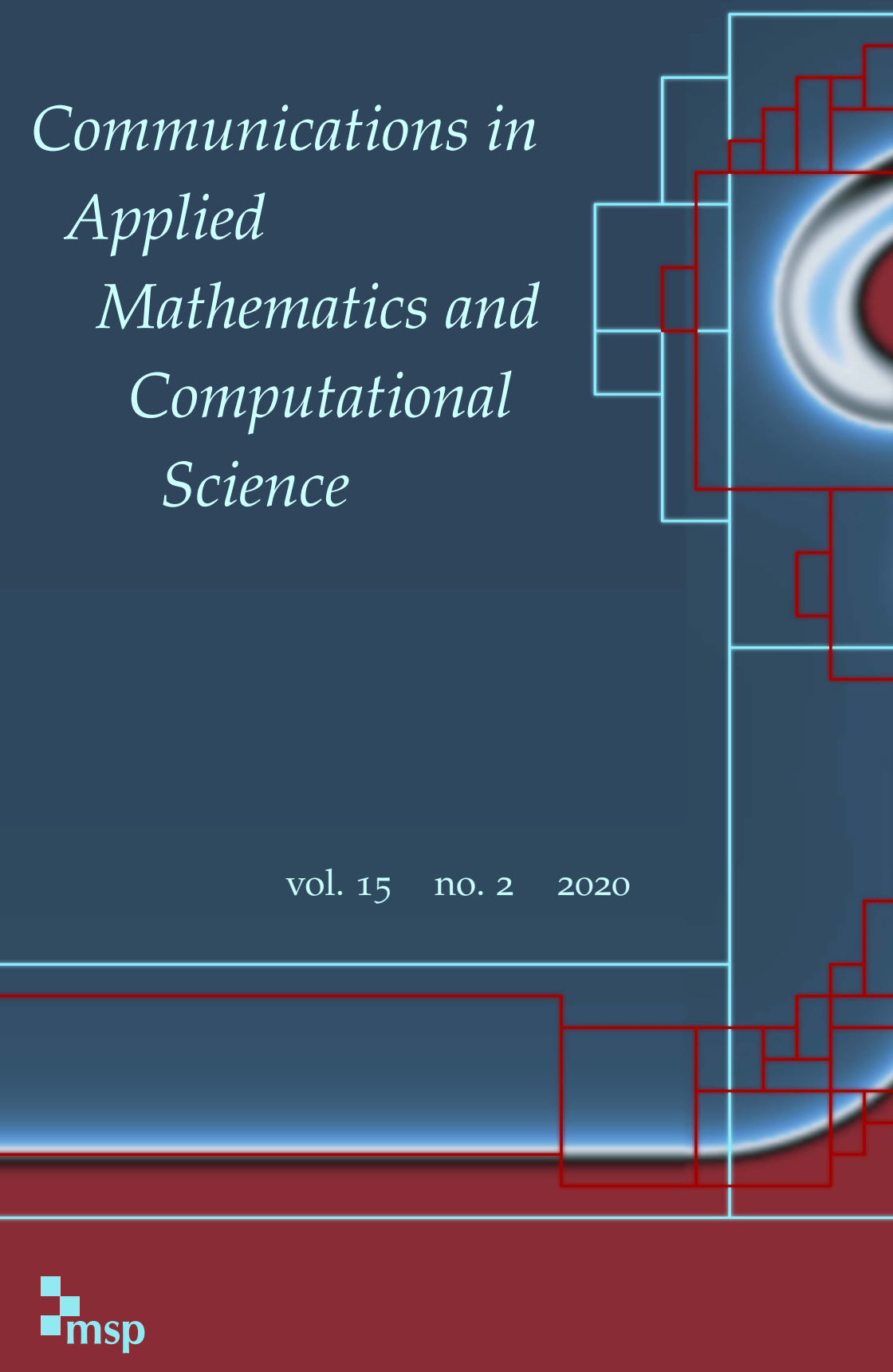
Communications in
Applied
Mathematics and
Computational
Science
vol. 15 no. 2 2020
msp

Communications in Applied Mathematics and Computational Science

msp.org/camcos

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

The subscription price for 2020 is US $110/year for the electronic version, and $165/year (+$15, if shipping outside the US) for print and electronic. Subscriptions, requests for back issues from the last three years and changes of subscriber 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 MSP.

PUBLISHED BY

mathematical sciences publishers

nonprofit scientific publishing

http://msp.org/

COMM. APP. MATH. AND COMP. SCI.
Vol. 15, No. 2, 2020
https://doi.org/10.2140/camcos.2020.15.115

msp

SIMULATING SINGLE-COIL MRI FROM THE RESPONSES OF MULTIPLE COILS

MARK TYGERT AND JURE ZBONTAR

We convert the information-rich measurements of parallel and phased-array MRI into noisier data that a corresponding single-coil scanner could have taken. Specifically, we replace the responses from multiple receivers with a linear combination that emulates the response from only a single, aggregate receiver, replete with the low signal-to-noise ratio and phase problems of any single one of the original receivers (combining several receivers is necessary, however, since the original receivers usually have limited spatial sensitivity). This enables experimentation in the simpler context of a single-coil scanner prior to development of algorithms for the full complexity of multiple receiver coils.

1. Introduction

Typical modern magnetic resonance imaging (MRI) scanners (such as all those represented in the fastMRI data of [13]) include *multiple* receivers — so-called "coils" of conductors — for making measurements. In this brief yet self-contained technical note, we seek to conveniently simulate the response of a single-coil machine, complete with errors that are reasonably representative of the measurement errors which a real scanner would make, given only the responses from a multicoil, "parallel-imaging" machine. The fastMRI competition issues two challenges — single-coil and multicoil — under the expectation that the single-coil track could provide a simplified stepping stone toward the full multicoil challenge.

The aim of fastMRI is compressed sensing — the accelerated acquisition of image reconstructions by taking fewer measurements than classical signal processing would need to attain the same resolution; compressed sensing yields full-resolution or "superresolved" reconstructions based on so-called "undersampled" measurements, measurements acquired at lower than the Nyquist rate required to reconstruct any arbitrary signal for a given bandwidth. Two approaches to compressed sensing are especially popular: (1) optimization for individual images and (2) machine learning over collections of images. Combinations of the two approaches are also possible.

MSC2010: primary 68U10, 78A50, 78M50, 92C55; secondary 65K10, 78A55, 94A08.
Keywords: magnetic resonance imaging, parallel imaging, multicoil, fastMRI, deep learning, machine learning.

Techniques representative of the first approach (via optimization) include Fourier reconstruction with total-variation regularizers (or related regularizers, such as the sum of the absolute values of wavelet coefficients); representative of the second approach (via machine learning) is statistical modeling with deep neural networks. The terminology "compressed sensing", "compressive sensing", and "compressive sampling" is due to [1; 2]; they took the first approach (via optimization). The second approach (machine learning) is the focus of [13].

Complicating the development of compressed sensing for MRI is the longstanding focus on leveraging multiple receiver coils in the design of MRI scanners, as reviewed by [5]. All scanners of [13] are multicoil by default, designed and optimized for multicoil measurements. However, compressed sensing is more complex with multicoil data. Distilling the multicoil challenge into a single-coil stepping stone could lower barriers to entry into research on compressed sensing for MRI and accelerate the development of accelerated MRI acquisitions, by teasing apart the fundamental mathematical issues from the multicoil complications.

Thus, we need to simulate a single-coil scanner with sufficient fidelity to actual errors in measurement, using only multicoil data. As hoped (and illustrated with numerical examples in Section 3 below), our single-coil emulation does indeed yield phase similar to the notorious phase problems tackled by [4; 6] and others. Encouragingly, the method of [8] is similar to ours and yielded highly successful results in a related setting (though, admittedly, our mathematical motivation, practical application, optimization procedure and targets, and detailed objective differ). Section 2 now introduces our simulation of a single-coil scanner.

2. Emulated single-coil

We propose using a linear combination of the responses from multiple coils for the emulated single-coil (ESC) response. We least-squares fit the complex-valued coefficients in the linear combination to the "ground-truth" reconstruction, estimating the ground truth using the canonical full multicoil reconstruction, the root-sum-square (RSS) of [7]. Most notably, linearly combining the raw coil responses sums complex-valued fluxes directly, rather than summing nonnegative energies as in the RSS. This simulates a single-coil scanner for the reasons discussed next.

2.1. ***Review.*** MRI scanners originally took measurements using only a single coil of wires, recording the magnetic flux through that coil, separately for each pixel or voxel in the spatial Fourier domain ("k-space"). Most large modern scanners instead take most measurements using several separate coils; the simplest such phased array of coils partitions what would have been one large coil into multiple disjoint smaller coils, as illustrated in Figure 1. Squaring the RSS of [7] sums the squares of the absolute values of the spatial inverse Fourier transforms of the magnetic

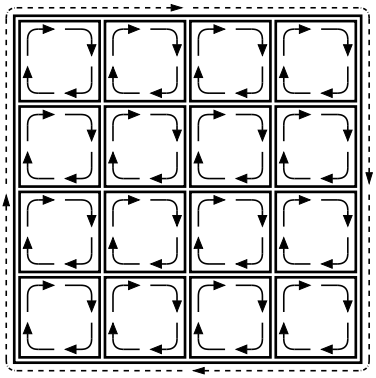

Figure 1. The wide solid lines represent coils of conductors — 16 small ones and 1 large one. The arrows indicate circulation of current through the conductors (with solid lines for the small coils and dashed lines for the large coil). The currents in the small coils would sum to induce the indicated current in the large coil (flux through the small coils would be equivalent to flux through the large coil), neglecting mutual inductances — and actual designs of coil arrays tend to minimize mutual inductances (as per [10]).

fluxes through the coils to obtain the total energy (or power) through the array; the total is the same as what a single coil around the perimeter of the array would have measured for the total energy (or power), but with improved signal-to-noise. (The disjoint partition ensures that the squares of the absolute values of the spatial sensitivities of the coils sum to 1 at any point in space, assuming that the array of coils captures all radiation emanating from the object being imaged.) Most MRI machines use several different partitions of what would have been a single large coil, with those different partitions overlapping each other, again to improve the signal-to-noise ratio. The partitions can leverage polarization to make more orthogonal measurements via quadrature pairs, butterflies, and figure eights; the survey of [10] discusses this together with "bird-cage" and other cylindrical or volumetric coils. In all cases, the basic principle is the same, and the RSS works the same.

2.2. ***Motivation.*** As seen in Figure 1, the flux through a single coil is physically the sum of the fluxes through multiple coils forming a disjoint partition of the single coil that loops around the multiple coils (provided that the multiple coils are all in phase and have the same gain, while neglecting the mutual inductance that coil design typically tries to minimize, as per [10]). This is much like in the proof of Stokes' theorem (which also uses Figure 1). Since we do not know the correct phase offset and gain calibrations, which amount to multiplying a whole coil's measurements by the same complex number, we can fit the complex number so as to match the ground-truth RSS reconstruction as well as possible.

As described, for example, by [10], many MRI machines use "bird-cage" or other volumetric coils rather than surface coils in a flat plane. A phased "single-coil"

array is often a cylindrical array of coils with capacitative (or inductive) couplers between the coils. These couplers effectively introduce phase offsets between the multiple conductors forming the "single-coil" bird-cage. Our linear combination of coil responses accounts for phase offsets and is physically realizable and similar to bird cages used in practice. The use of coils selecting for circular polarization (including the quadrature pairs, butterflies, figure eights, etc., described by [10]) is an additional complication, implicitly resolved by how the least-squares fit of the linear combination to the ground truth decides to combine the coils. The following subsection formalizes the mathematical details.

2.3. ***Mathematical formulation.*** To set notation, we define k to be the number of coils, l to be the number of slices in a volume of cross-sectional images, m to be the height of each slice, and n to be the width of each slice. The measured data from multiple coils then consists of an $(nml) \times k$ matrix A with complex-valued entries. Each of the k coils yields l cross-sectional slices, with each slice being an $m \times n$ image. More precisely, the images in A are cropped inverse Fourier transforms of the original two-dimensional Fourier domain ("k-space") measurements, cropped to the center $m \times n$ block of pixels (the cropping eliminates the usual oversampling to focus on the region of interest to radiologists).

To guide the fitting of parameters for the simulations, we provide "ground-truth" targets by filling an $(nml) \times 1$ column vector b with the RSS reconstructions of [7] obtained from the full multicoil data; specifically, each entry of b is the Euclidean norm of the corresponding $1 \times k$ row in A. While this "ground truth" may be imperfect, the RSS is the canonical choice, used universally throughout medical literature since the work of [7]. Each pixel of each cross-sectional slice in b is the square root of the sum of the energy (or power) flowing for that pixel through the individual coils; the individual coils could be arranged, for example, as illustrated in Figure 1. Each pixel of this RSS reconstruction b collects together all the energy (or power) for that pixel from the individual coils.

Given A and b, we calculate the coefficients in a linear combination of the coils as the (complex-valued) entries in a $k \times 1$ column vector x minimizing

$$\left\| \sqrt{|Ax|} - \sqrt{|b|} \right\|_2^2, \tag{1}$$

where $\| \cdot \|_2$ is the Euclidean norm, $\sqrt{\cdot}$ takes the square root entrywise, and $|\cdot|$ takes the absolute value entrywise, so that $|b|$ is in fact equal to b, as the entries of b are nonnegative. The distance $\left\| \sqrt{|Ax|} - \sqrt{|b|} \right\|_2$ is known as the Hellinger metric of [3]; the square roots amplify entries of small magnitude and attenuate entries of large magnitude. Furthermore, $\sqrt{|Ax|} - \sqrt{|b|} = (|Ax| - |b|)/(\sqrt{|Ax|} + \sqrt{|b|})$, so the Hellinger distance is more like a measure of relative errors than the conventional Euclidean distance $\| |Ax| - |b| \|_2$. Our estimate of $|b|$ (which equals b) is $|Ax|$,

where x minimizes (1). We use absolute values in our estimate $|Ax|$ since the entries of the ground truth we want to match are nonnegative. We use the Hellinger distance since human visual perception is somewhat more calibrated for relative rather than absolute errors; in many cases, omitting the square roots — thereby optimizing mainly for the pixels of greatest intensity — would spoil all but a small fraction. That said, minimizing (1) is far from the only possibility; minimizing (1) performed well in our numerical experiments and facilitates the nonlinear optimization required.

We used the LBFGS (limited-memory Broyden–Fletcher–Goldfarb–Shanno) version of quasi-Newton iterations as implemented in SciPy [11] to solve the nonlinear least-squares problem minimizing (1). We started the iterations with the solution to the linear least-squares problem minimizing

$$\|Ax - b\|_2^2; \tag{2}$$

the objective function in (1) is the same as in (2) aside from taking the square roots of the absolute values of the entries of Ax and of b. As indicated above, the solution x minimizing (2) is unsuitable for the final results, as some entries of Ax can be negative, and only the largest entries of Ax will be accurate in comparison with those optimizing the Hellinger metric. Our final results always use $|Ax|$ with the solution x minimizing (1).

Remark. Anuroop Sriram points out that the mean of the noise on the "ground-truth" RSS will be strictly positive in regions of the image where the actual object being imaged is absent, as the RSS is always nonnegative. Thus, when we fit the linear combination of the coil responses, we should try to match the RSS only where the RSS is substantially above a threshold that is significantly above the noise level. For the current data from fastMRI of [13], essentially the entire cropped image of interest meets this criterion, so we try to match the full cropped image.

3. Numerical examples

We compare the emulated single-coil (ESC) method of Section 2 with the leading singular vector of the singular value decomposition (SVD) of A from (1), as in the "eigencoil" or "coil compression" reviewed by [5]. ESC could be regarded as a special kind of the "virtual-coil" methods discussed by [5]. We also display the RSS of [7] for reference as our best estimate of the ground truth (the second paragraph of Section 2.3 above reviews the RSS).

For our data — scans of 1594 knees from [13] — the parameters from Section 2 take the values $k = 15$ receiver coils, $m = 320$ rows, and $n = 320$ columns in a cross-sectional image, with Figure 2 specifying l, the number of cross-sectional slices. We used a memory of 10 vectors in the limited-memory BFGS quasi-Newton iterations to minimize (1) starting from the linear least-squares solution minimizing (2).

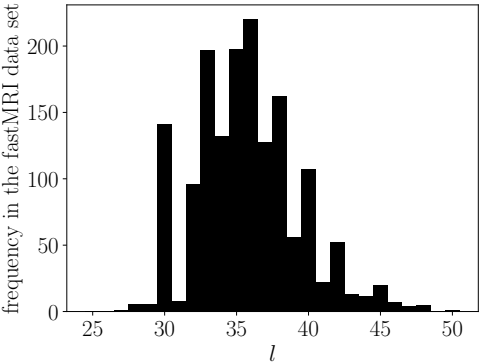

Figure 2. Number of examples in the fastMRI data set of [13] for each possible value of l (the parameter l is the number of cross-sectional slices).

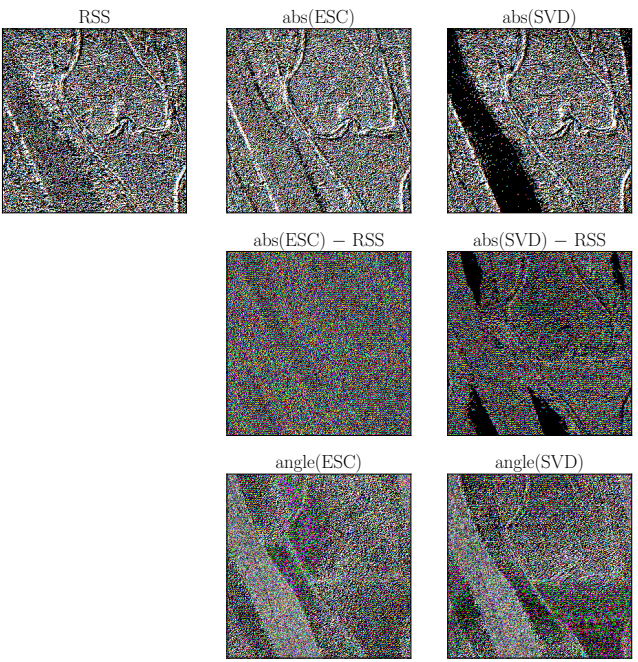

Figure 3. First randomly chosen example; "abs" refers to the absolute value of a complex number and "angle" refers to the phase.

In Figures 3–8, the first row shows (the magnitude of) the reconstructions. The second row shows the difference between the RSS ground truth and the magnitude of each method's reconstruction. The third row shows the phase of the reconstructions (estimates of the RSS ground truth discard the phase). Figures 12–14 in the Appendix display the entries of the solution x minimizing (1), where the ESC reconstruction is Ax.

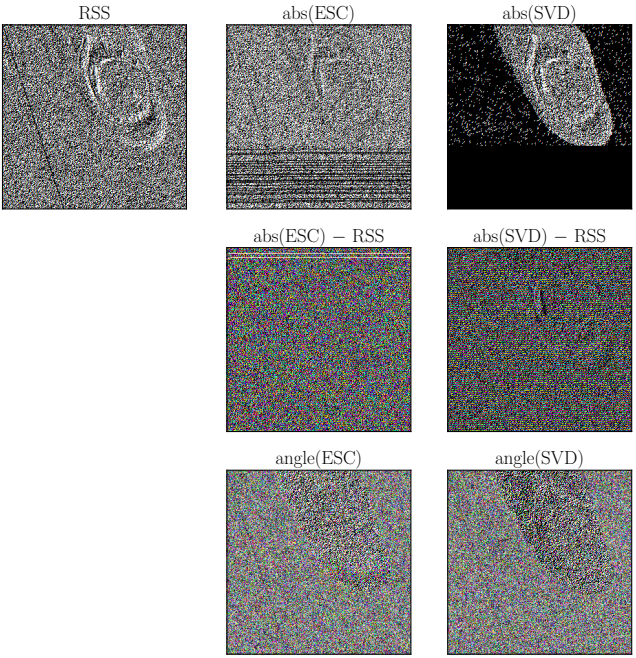

Figure 4. Second randomly chosen example.

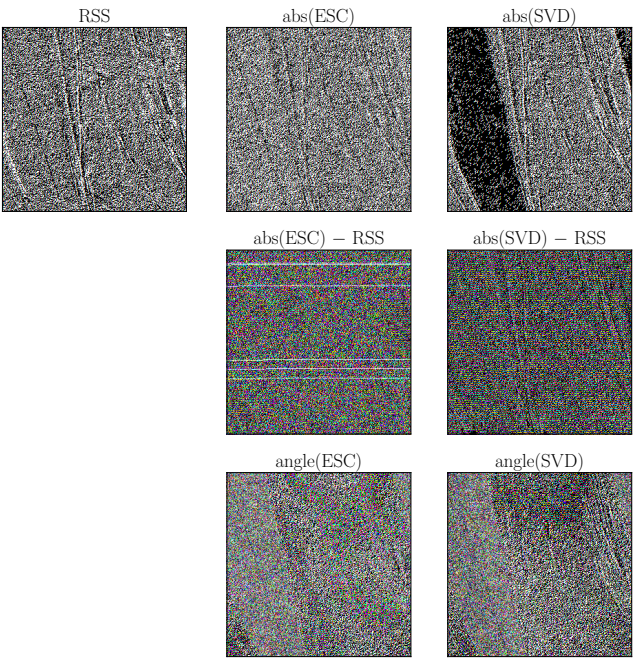

Figure 5. Third randomly chosen example.

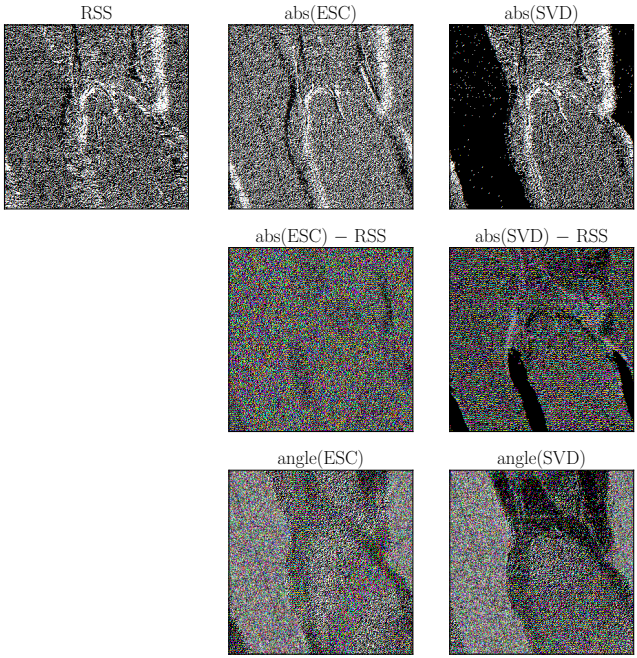

Figure 6. Fourth randomly chosen example.

Figures 3–8 show that the SVD-based compression to a single eigenmode for simulating the single coil yields dramatic differences from the ground-truth RSS, introducing severe artifacts. Compressing to only one eigenmode is apparently too drastic.

Figures 3–8 show that the reconstruction from ESC matches the ground-truth RSS reasonably well, but is noisy. The noise on the reconstruction from ESC is actually desirable for simulating a single-coil scanner (since a single-coil scanner is likely to be noisier than the ground truth estimated from the full multicoil data). This matches our expectation that ESC emulates a physically realizable process for converting a multicoil scanner into a single-coil scanner. The scanner which ESC simulates is genuinely physically realizable, even though it only approximates single-coil scanners that have actually been manufactured.

Figures 9–11 display examples for which ESC deviates substantially from the ground-truth RSS. These are among the worst problems we identified (though of course "worst" is somewhat subjective — SSIM of [12] was helpful in finding bad cases); about 4% of the volumes scanned exhibited some indication of such a problem, with about 10% of the cross-sectional slices within those volumes showing signs of the problems displayed in Figures 9–11. Overall, then, around 0.4% of the two-dimensional images in the data set display significant artifacts.

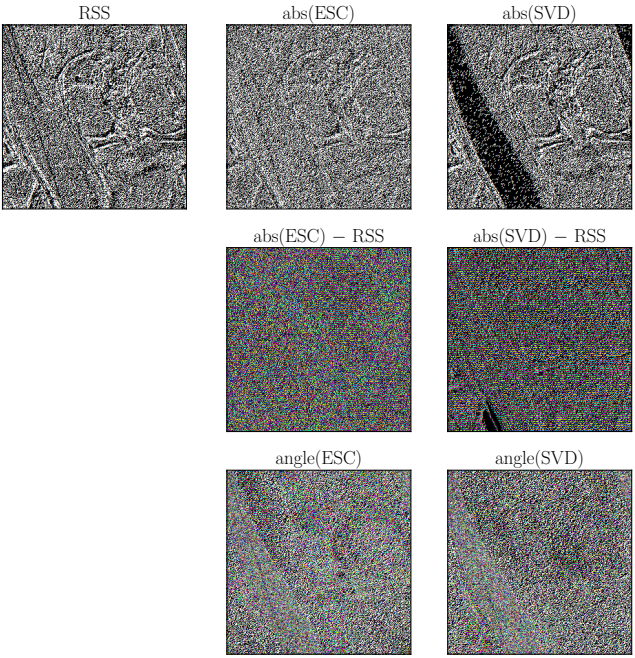

Figure 7. Fifth randomly chosen example.

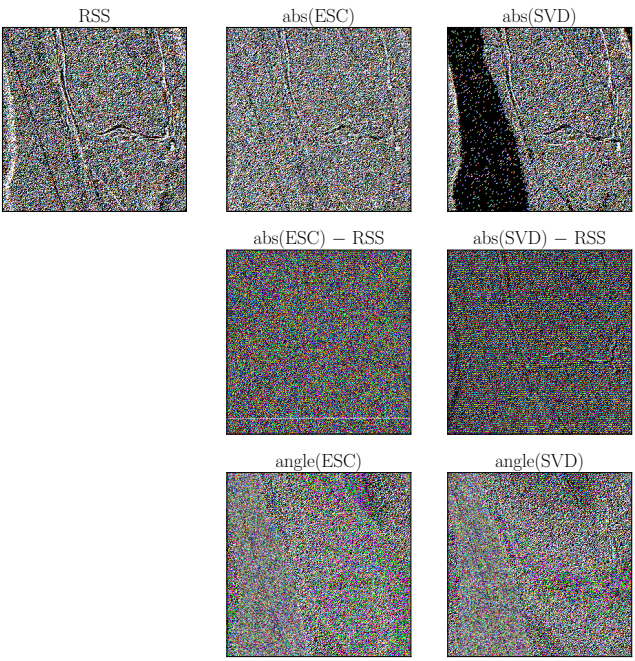

Figure 8. Sixth randomly chosen example.

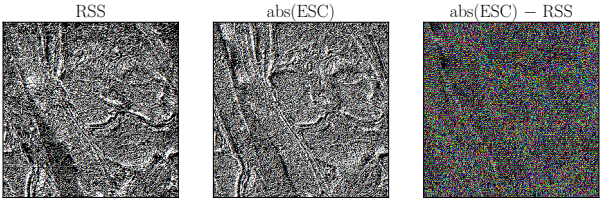

Figure 9. "Worst" abs(ESC)−RSS.

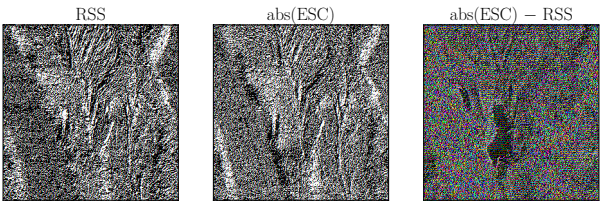

Figure 10. Second "worst" abs(ESC)−RSS.

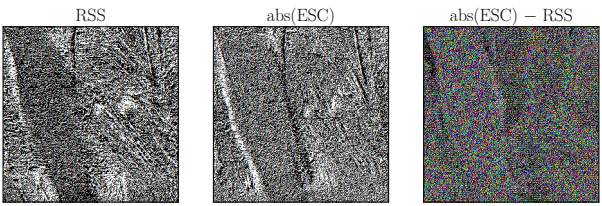

Figure 11. Third "worst" abs(ESC)−RSS.

The main culprit seems to be ringing artifacts near the boundary of the object being imaged. Whether such occasional ringing oscillations are an artifact of our fitting procedure or are inherent in the data when converting from multiple receivers remains unclear; however, the "ground-truth" RSS reconstructions do exhibit similar (albeit abated) artifacts. Having even a small number of cases with artifacts is indisputably undesirable, but then again data from the real world acquired in actual clinical practice is seldom ideal. Seeing as "ground-truth" RSS reconstructions exhibit similar (though less intense) artifacts, conducting simulations that avoid such artifacts may be impossible with this imperfect data.

Remark. The method proposed in the present paper is heavily data-driven. If computational simulations or excellent empirical estimates of the "physical coil sensitivity maps" defined in [5] could be calculated with high fidelity to the physical reality represented in the fastMRI data of [13], then less data-intensive methods might be feasible. Unfortunately, actual MRI scanners are exceptionally complex and deviate significantly from textbook idealizations; interactions of the objects

being imaged with the scanners further distort the sensitivities of receiver coils, as detailed, for example, by [4; 6]. While autocalibration procedures such as those from ESPIRiT of [9] work well for multicoil reconstruction, their estimates of sensitivity maps are specifically tailored for multicoil reconstruction, not for constructing single-coil measurements from multicoil.

4. Conclusion

Although our simulation of a single-coil scanner might appear perverse and pointless at first glance — after all, we throw away the information necessary to realize the gains in signal-to-noise ratios that imaging with multiple receivers provides — maintaining fidelity to a physically realizable simple MRI machine in fact yields a convenient test case, a reliable stepping stone toward the complexity of full parallel-MRI reconstruction. Less than 0.5% of the resulting emulated single-coil (ESC) images exhibit artifacts such as those displayed in Figures 9–11; Figures 3–8 display typical examples. The obvious alternative to our scheme would be to take measurements on an actually manufactured single-coil MRI scanner; however, MRI is extremely expensive, especially in the medical domain considered for the fastMRI data set of [13].

Acknowledgements

We would like to thank Matt Muckley, Erich Owens, Mike Rabbat, Dan Sodickson, and Anuroop Sriram.

Appendix: Entries of the solution vectors

Figures 12–14 display the entries of the solution x minimizing (1), where the ESC reconstruction of $|b|$ in (1) is $|Ax|$ from (1). No especially interesting patterns

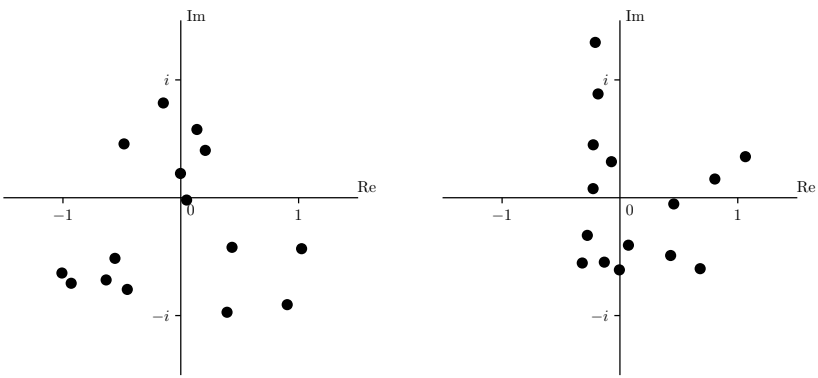

Figure 12. Entries of x minimizing (1) corresponding to the data in Figures 3 (left) and 4 (right).

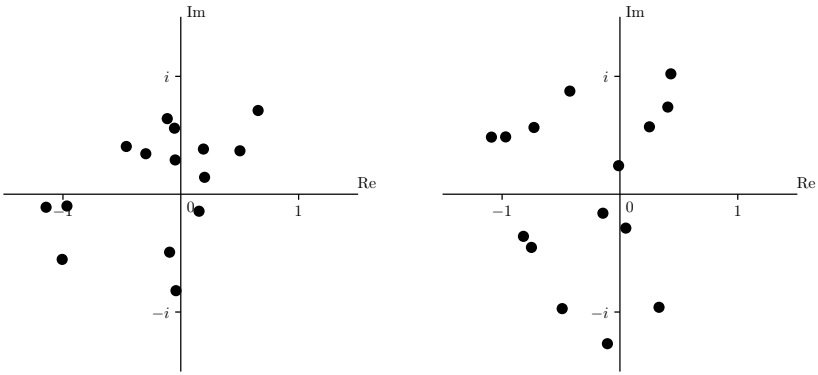

Figure 13. Entries of x minimizing (1) corresponding to the data in Figures 5 (left) and 6 (right).

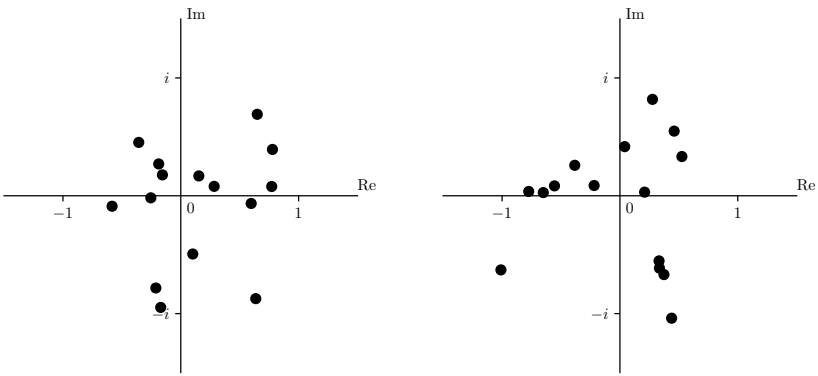

Figure 14. Entries of x minimizing (1) corresponding to the data in Figures 7 (left) and 8 (right).

appear to be evident; the coefficients seem to scatter somewhat randomly around the unit disk in the complex plane, not concentrating anywhere in particular.

References

[1] E. J. Candes and T. Tao, *Decoding by linear programming*, IEEE Trans. Inform. Theory **51** (2005), no. 12, 4203–4215. MR Zbl

[2] D. L. Donoho, *Compressed sensing*, IEEE Trans. Inform. Theory **52** (2006), no. 4, 1289–1306. MR Zbl

[3] E. Hellinger, *Die Orthogonalinvarianten quadratischer Formen von unendlichvielen Variabelen*, Ph.D. thesis, Georg-August-Universität zu Göttingen, 1907. Zbl

[4] J. H. Lee, J. M. Pauly, and D. G. Nishimura, *Partial k-space reconstruction for radial k-space trajectories in magnetic resonance imaging*, US patent US7277597B2, Stanford University, 2007.

[5] D. K. Ohliger, Michael A. Sodickson, *An introduction to coil array design for parallel MRI*, NMR Biomed. **19** (2006), no. 3, 300–315.

[6] J. M. Pauly, *Partial k-space reconstruction*, preprint, 2005.

[7] P. B. Roemer, W. A. Edelstein, C. E. Hayes, S. P. Souza, and O. M. Mueller, *The NMR phased array*, Magn. Reson. Med. **16** (1990), no. 2, 192–225.

[8] K. Setsompop, L. L. Wald, V. Alagappan, B. A. Gagoski, and E. Adalsteinsson, *Magnitude least squares optimization for parallel radio frequency excitation design demonstrated at 7 Tesla with eight channels*, Magn. Reson. Med. **59** (2008), no. 4, 908–915.

[9] M. Uecker, P. Lai, M. J. Murphy, P. Virtue, M. Elad, J. M. Pauly, S. S. Vasanawala, and M. Lustig, *ESPIRiT — an eigenvalue approach to autocalibrating parallel MRI: where SENSE meets GRAPPA*, Magn. Reson. Med. **71** (2014), no. 3, 990–1001.

[10] J. T. Vaughan and J. R. Griffiths (eds.), *RF coils for MRI*, Wiley, New York, 2012.

[11] P. Virtanen, R. Gommers, T. E. Oliphant, M. Haberland, T. Reddy, D. Cournapeau, E. Burovski, P. Peterson, W. Weckesser, J. Bright, S. J. van der Walt, M. Brett, J. Wilson, K. Jarrod Millman, N. Mayorov, A. R. J. Nelson, E. Jones, R. Kern, E. Larson, C. Carey, İ. Polat, Y. Feng, E. W. Moore, J. VanderPlas, D. Laxalde, J. Perktold, R. Cimrman, I. Henriksen, E. A. Quintero, C. R. Harris, A. M. Archibald, A. H. Ribeiro, F. Pedregosa, P. van Mulbregt, and contributors, *SciPy 1.0: fundamental algorithms for scientific computing in Python*, Nat. Methods **17** (2020), 261–272.

[12] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, *Image quality assessment: from error visibility to structural similarity*, IEEE Trans. Image Process. **13** (2004), no. 4, 600–612.

[13] J. Zbontar, F. Knoll, A. Sriram, T. Murrell, Z. Huang, M. J. Muckley, A. Defazio, R. Stern, P. Johnson, M. Bruno, M. Parente, K. J. Geras, J. Katsnelson, H. Chandarana, Z. Zhang, M. Drozdzal, A. Romero, M. Rabbat, P. Vincent, N. Yakubova, J. Pinkerton, D. Wang, E. Owens, C. L. Zitnick, M. P. Recht, D. K. Sodickson, and Y. W. Lui, *fastMRI: an open dataset and benchmarks For accelerated MRI*, preprint, 2019. arXiv

Received May 27, 2019. Revised May 1, 2020.

MARK TYGERT: mark@tygert.com
Artificial Intelligence Research, Facebook, Menlo Park, CA, United States

JURE ZBONTAR: jzb@fb.com
Artificial Intelligence Research, Facebook, New York, NY, United States

COMM. APP. MATH. AND COMP. SCI.
Vol. 15, No. 2, 2020
https://doi.org/10.2140/camcos.2020.15.129

ON A COMPUTATIONALLY SCALABLE SPARSE FORMULATION OF THE MULTIDIMENSIONAL AND NONSTATIONARY MAXIMUM ENTROPY PRINCIPLE

ILLIA HORENKO, GANNA MARCHENKO AND PATRICK GAGLIARDINI

Data-driven modeling and computational predictions based on the maximum entropy principle (MaxEnt principle) aim to find as simple as possible — but not simpler than necessary — models that allow one to avoid the data-overfitting problem. We derive a multivariate nonparametric and nonstationary formulation of the MaxEnt principle and show that its solution can be approximated through a numerical maximization of the sparse constrained optimization problem with regularization. Application of the resulting algorithm to popular financial benchmarks reveals memoryless models allowing for simple and qualitative descriptions of data of the major stock market indices. We compare the obtained MaxEnt models to the heteroscedastic models from computational econometrics (GARCH, GARCH-GJR, MS-GARCH, and GARCH-PML4) in terms of the model fit, complexity, and prediction quality. We compare the resulting model log-likelihoods, the values of the Bayesian information criterion, posterior model probabilities, the quality of the data autocorrelation function fits, as well as the value-at-risk prediction quality. We show that all of the seven considered major financial benchmark time series (DJI, SPX, FTSE, STOXX, SMI, HSI, and N225) are better described by conditionally memoryless MaxEnt models with nonstationary regime-switching than by the common econometric models with finite memory. This analysis also reveals a sparse network of statistically significant temporal relations for the positive and negative latent variance changes among different markets. The code is provided for open access.

The maximum entropy principle (MaxEnt principle) was originally introduced in physics and information theory to search for least-biased probabilistic data descriptions that match certain statistical properties of the data, e.g., the data distribution moments [18]. The MaxEnt principle implies that the most unbiased distribution for the data is the one that admits the most uncertainty, measured in terms of entropy. Depending on the constraints imposed in the entropy maximization problem,

The authors thank Michael Rockinger (HEC Lausanne) for providing the code for the PML4 parameter estimator we used for comparison. This work was funded by the SNSF (project 140829) and DFG SPP 1114 (Mercator Fellowship of Horenko).
MSC2010: 28D20, 37M10, 37M25, 91B84.
Keywords: machine learning, financial time series, maximum entropy, heteroscedasticity, sparsity.

different parametric probability distributions that can be described by this principle are Gaussian, exponential, Laplace, Cauchy, chi-squared, and gamma distributions, among others. This MaxEnt-based probabilistic modeling approach has been successfully applied to many problems ranging from biology [26; 24] and natural language processing [3; 25] to applications from economics and finance [31; 30].

In contrast to the parametric MaxEnt modeling, where the particular parametric distribution models (Gaussian, exponential, etc.) are dependent on the fixed finite set of constant parameters, computation of nonparametric MaxEnt densities even in one dimension is not a trivial task, as was previously discussed in [1; 23; 15]. In [22] a systematic mathematical derivation for a nonstationary extension of the nonparametric MaxEnt methodology for one-dimensional time series problems was introduced. This BV-entropy framework imposes a mild bounded-variation (BV) assumption on the time dependence of the nonstationary moments of the underlying nonstationary probability distribution function (p.d.f.). In [22] it was shown that the original ill posed nonstationary MaxEnt principle — formulated as an entropy maximization problem with time-dependent moment constraints — can be sharply bounded from below via a well posed and computationally scalable maximization problem. This lower-bound problem appeared to be a nonparametric regime-switching entropy maximization problem with K locally stationary regimes, subject to K l_1-constraints on the regime-specific vectors of moments and to a BV-constraint for the latent regime transitions. The linear BV-constraint happens to bound from above with C the maximum number of regime switches — and controls the persistence of the obtained MaxEnt regime transition models. It was also shown that the optimal values of C and K can be estimated by deploying common model selection criteria like the Bayesian information criterion (BIC).

However, this model formulation is confined to one-dimensional time series analysis problems only. A direct extension of this one-dimensional BV-entropy methodology to multiple dimensions is hampered by the curse of dimension and the overfitting problems: linear growth in the number of problem dimensions will result in the exponential growth in the number of underlying multivariate MaxEnt parameters that have to be determined from the data statistics. In many practical applications — for example in economics and finance — there is only one historical realization of the process that is available, with no possibility to obtain other sampling realizations from some "model". For every particular dimension at every particular time there is only one data point available. This means that a completely nonparametric approach to statistical analysis of such data leads to ill posed problems [13].

In the following, we describe a computational algorithm that achieves a sparse multidimensional extension of the one-dimensional BV-entropy model from [22], where the coupling between the n individual one-dimensional entropy maximization

problems will be achieved through a sparsifying regularization constraint that controls appropriate function space distances (l_2, l_1, or BV) between the individual latent factors.

The resulting algorithm (further referred to as TV-entropy) allows a computationally tractable search for the most unbiased multivariate time-dependent distribution of the data, minimizing the optimal number of locally stationary hidden regimes and sparsifying their time-persistent regime-switching dynamic and the regime-specific distribution parameter vectors. Then we illustrate its performance and compare it to common approaches on the test model system (a regime-switching Gaussian, Figure 1). Finally, we apply this algorithm and the common GARCH tools to a set of seven popular financial stock market indices and show that these simpler memoryless MaxEnt models of volatility outperform the popular heteroscedastic methods with finite memory for all benchmarks considered (see Figures 2–4 and Table 2).

Method

We start with n-dimensional multivariate time series data $x_{t,i}$ for all $t \in \{1, \ldots, T\}$ and all $i \in \{1, \ldots, n\}$ on a closed interval $\mathcal{X} = \mathcal{X}_1 \times \cdots \times \mathcal{X}_n \subset \mathbb{R}^n$. The goal is to find the most descriptive and least complex model for this data. We will first assume that x_t is conditionally independent in time, and would like to estimate its unknown marginal time-dependent probability densities $f_{t,i}(x)$. In the context of the MaxEnt principle, the density $f_{t,i}(x)$ can be identified by solving an optimization problem which consists of finding a time-dependent density function with the maximum entropy among all distribution functions that match the data in the first $m + 1$ sample moments at all of the given time instances t in the dimension i. There is a long tradition in statistics and econometrics of adopting inference methodologies based on matching sample moments (generalized method of moments (GMM) estimation [12]). Moreover, MaxEnt methods are often invoked in multinomial choice problems [11; 21]. Maximization of the expected entropy with respect to a time-dependent multivariate probability density $f_{t,i}(x)$ (where the expectation is taken over the time t and the dimension n) can be written as

$$\max_{f_{t,i}(x) \text{ for all } t,\, i} \left\{ \mathbb{E}_{t,i}[H[f_{t,i}(x)]] = \mathbb{E}_{t,i}\left[-\int_{\mathcal{X}_i} f_{t,i}(x) \ln f_{t,i}(x)\, dx \right] \right\}, \tag{1}$$

subject to

$$\int_{\mathcal{X}_i} x^j f_{t,i}(x)\, dx = \mu_{t,i}(j) \quad \text{for all } j \in \{0, \ldots, m\},\ i \in \{1, \ldots, n\}, \text{ and } t \in \{1, \ldots, T\}, \tag{2}$$

and $\mu_{t,i} \in \mathcal{R}^{m+1}$ are time- and dimension-dependent sample moments. Then the optimal $f^*_{t,i}(x)$ can be derived by computing first-order optimality conditions of

the corresponding optimization problem (1)–(2), providing the formulation

$$f_{t,i}(x) = \exp\left[-\sum_{j=0}^{m} \Lambda_{t,i}(j)x^j\right] \quad \text{for all } t, i \tag{3}$$

such that

$$\int_{\mathscr{X}_i} x^n \exp\left[-\sum_{j=0}^{m} \Lambda_{t,i}(j)x^j\right] dx = \mu_{t,i}(n) \quad \text{for all } n \in \{0, \dots, m\}, \tag{4}$$

$$\mu_{t,i}(0) = 1, \tag{5}$$

where $\Lambda_{t,i} \in \mathscr{R}^{m+1}$ are unknown time-dependent parameters (Lagrange multipliers) of MaxEnt distributions. One way to compute $\Lambda(\cdot)$ would be to maximize the log-likelihood function based on the obtained densities $f_{t,i}(x)$, i.e.,

$$\max_{\Lambda(\cdot)} \mathscr{L}(\Lambda(\cdot)) = -\sum_{i=1}^{n}\sum_{t=1}^{T}\left(\sum_{j=1}^{m} \Lambda_{t,i}(j)x_t^j + \ln Z_{\Lambda_{t,i}}\right), \tag{6}$$

$$\text{where} \quad Z_{\Lambda_{t,i}} = \int_{\mathscr{X}_i} \exp\left[-\sum_{j=1}^{m} \Lambda_{t,i}(j)x^j\right] dx = \exp[\Lambda_{t,i}(0)]. \tag{7}$$

The first-order optimality conditions of the problem (6) are equivalent to conditions (3)–(5); therefore, $\Lambda(\cdot)$ maximizing the criterion in (6) are the optimal values of the MaxEnt densities parameters in (3)–(5).

In realistic applications when only one historical realization sequence $\{x_1, \dots, x_T\}$ is available, there is no straightforward solution to the nonstationary problem (6), as at each time instance t we have many more parameters than we have observed data. To solve this problem, we are going to introduce the two following assumptions.

Assumption 1. Total variation (a TV-norm) of the MaxEnt parameters $\Lambda(\cdot)$ is bounded; i.e.,

$$|\Lambda|_{\mathrm{TV}} = \sum_{t_1,t_2=1}^{T}\sum_{i_1,i_2=1}^{n} |\Lambda_{t_1,i_1}(\cdot) - \Lambda_{t_2,i_2}(\cdot)|_1 = C < +\infty. \tag{8}$$

Assumption 2. There exist $K \ll nT$ distinct sets of parameters $\lambda^{(k)}$, $k = 1, \dots, K$, and $\gamma_{t,i} = [\gamma_{t,i}^{(1)}, \dots, \gamma_{t,i}^{(K)}]$ (with $\sum_{k=1}^{K} \gamma_{t,i}^{(k)} = 1$ and $\gamma_{t,i}^{(K)} \geq 0$ for all t, i, k), such that for any t, i, and k the vector $\Lambda_{t,i}$ can be expressed as a convex linear combination

$$\Lambda_{t,i} = \sum_{k=1}^{K} \gamma_{t,i}^{(k)} \lambda^{(k)}. \tag{9}$$

Assumptions 1 and 2 introduce sparsity in $\Lambda_{t,i}$ across time and space indices t and i. Very importantly, these assumptions do not rely on any ordering in the space dimension. Indeed, in many practical applications, e.g., in economics and finance,

a natural ordering across assets, institutions, markets, etc., does not exist. On a practical side, for real financial data these assumptions will be fulfilled automatically if one sets both constants K and C to be large enough (for example, setting $K = nT$ always fulfills Assumption 2). In the practical applications the aim will be in finding the computationally scalable lower-bound estimates of these constants. Substituting condition (9) in (8), using Jensen's inequality, and inserting into (6) the obtained inequality constraint as the penalty term of the Karush–Kuhn–Tucker conditions leads to the following lower-bound approximation of the MaxEnt problem (6):

$$\max_{\lambda,\gamma} L(\lambda,\gamma) = -\sum_{k=1}^{\boldsymbol{K}}\Bigg[\sum_{i=1,\,t=1}^{n,T}\gamma_{t,i}^{(k)}\Bigg(\sum_{j=1}^{m}\lambda_j^{(k)}x_{t,i}^{j}+Z_i^{(k)}\Bigg) + \sigma_C|\lambda^{(k)}|_1\sum_{t_1,t_2=1}^{T}\sum_{i_1,i_2=1}^{n}|\gamma_{t_1,i_1}^{(k)}-\gamma_{t_2,i_2}^{(k)}|\Bigg] \quad (10)$$

such that

$$Z_i^{(k)} = \ln\int_{\mathscr{X}_i}\exp\Bigg[-\sum_{j=1}^{m}\lambda_{(j)}^{(k)}x^j\Bigg]dx, \quad \gamma_{t,i}^{(k)} \geq 0, \quad \sum_{k=1}^{\boldsymbol{K}}\gamma_{t,i}^{(k)} = 1, \quad \sigma_C \geq 0. \quad (11)$$

A solution of the obtained lower-bound problem is an approximation to the solution of the original nonstationary ill posed MaxEnt problem (1). The optimization criterion in (10) is nonlinear and nonconvex — implying that the problem can have more than one locally optimal solution. However, there are three properties of this optimization problem formulation that can be exploited numerically: (i) when λ is kept fixed, (10)–(11) becomes a uniquely solvable linear programming (LP) problem with respect to γ — and can be solved very efficiently by means of common LP-tools (e.g., with the simplex method); (ii) if $\sigma_C = 0$, solving (10)–(11) becomes equivalent to solving n independent one-dimensional nonstationary entropy maximization problems from the BV-entropy method in [22]; (iii) the algebraic structure of the term containing σ_C is similar to the LASSO-regularization formulation very popular in machine learning [29], and increasing σ_C will result in increasing the sparsity of λ and in penalizing the temporal and cross-sectional variations in γ — making the obtained MaxEnt model approximations more simple, sparse, and persistent across dimensions and in time. For a numerical solution of the problem (10)–(11) with a fixed set of parameters K and σ_C we will adapt the subspace algorithms introduced in [16; 17] (that were further developed in [9; 27]): to get the best possible use of the algebraic structure in (10)–(11), resulting algorithms should iterate between two distinct optimization problems, where the problem (10)–(11) is solved with respect to Λ (for current fixed values of Γ) and in the following step with respect to Γ (for current fixed values of Λ). The Λ-optimization step will involve the independent solution of $\boldsymbol{K}$ regularized stationary MaxEnt problems. As a result, an iterative

procedure would converge monotonically to a local maximum solution of the problem (10)–(11). This procedure — further referred to as TV-entropy — should be repeated for different combinations of the input parameters $\boldsymbol{K}, \sigma_C$ (from some predefined discrete sets), and common model discrimination tools like the Akaike information criterion [6], cross-validation [19], or bootstrap [6; 19] will be deployed to identify the most optimal combination of — hopefully small — parameters $\boldsymbol{K}, \sigma_C$.

It is straightforward to verify that the second term in (10) can alternatively be formulated as a set of two linear inequality constraints

$$|\lambda^{(k)}|_1 \leq C_{\lambda^{(k)}} \quad \text{for all } k = 1, \ldots, \boldsymbol{K}, \tag{12}$$

$$\sum_{t_1,t_2=1}^{T} \sum_{i_1,i_2=1}^{n} |\gamma_{t_1,i_1}^{(k)} - \gamma_{t_2,i_2}^{(k)}| \leq C_\gamma \quad \text{for all } k = 1, \ldots, \boldsymbol{K}, \tag{13}$$

where $C_\gamma \sum_{k=1}^{\boldsymbol{K}} C_{\lambda^{(k)}} \leq C$, with C being the global TV-constant defined in (8). This formulation allows a separate handling of constraints for the regime-switching and for the MaxEnt parameters with two separate sets of constraining variables $C_{\lambda^{(k)}}$ and C_γ. In contrast, the Tykhonov-like formulation (10)–(11) allows joint simultaneous control for all of the variables by means of a single constraining variable σ_C.

Results

First, we illustrate an application of the TV-entropy methodology introduced above and its comparison to the common regime identification methods like HMM and the adaptive Gaussian moving window (GMW) in a study with artificial simulated data. For this simulated data study we use the model system proposed in [22]: we generate multiple samples from the regime-switching model with two Gaussian regimes, i.e., $\tilde{x}_t = \sum_{i=1}^{2} \gamma_t^{(i)} \tilde{x}_t^{(i)}$, where $\tilde{x}_t^{(i)} \sim \mathcal{N}(0, v^{(i)})$ with $v^{(1)} = 1$ and $v^{(2)} = \tilde{v}$. For each value of $v^{(2)}$ we generate 100 samples with 1000 points. The regime-switching weights are discrete and satisfy the convexity constraints in (11), imposing only one active regime at time t. There are three regime transitions in the data-generating process that occur every 250 points. The time-dependent variance signal can then be computed as $v_t = \sum_{i=1}^{2} \gamma_t^{(i)} v^{(i)}$. This test system is designed to be favorable for HMM and GMW — since their common variants rely on the Gaussianity assumption for the realizations. The TV-entropy models were estimated with two regimes, six density regime parameters, and between one and ten regime switches. We used ten annealing steps for estimating both the TV-entropy and HMM models. The corresponding optimal parameters were chosen using BIC. For the GMW model we used bandwidth value $b = \{10, 30, 50\}$ to obtain various levels of persistence. As demonstrated in Figure 1, TV-entropy outperforms these common models in data classification and reconstruction of the variance signal used in the data-generating process.

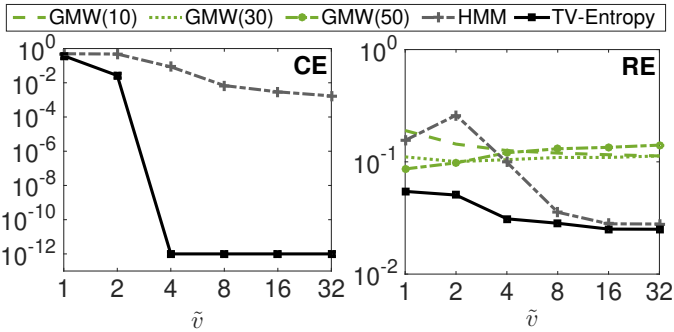

Figure 1. Average classification error (CE) between the true and reconstructed switching regime paths (left) and average relative error (RE) between the true and the reconstructed variance signals (right) in log scale obtained for 100 two-regime Gaussian samples with $\mathcal{N}(0, 1)$ and $\mathcal{N}(0, \tilde{v})$ regimes.

	stationary, without regime transitions	nonstationary, with regime transitions
parametric	GARCH/GARCH-GJR	MS-GARCH
semi- or nonparametric	GARCH-PML4	TV-entropy

Table 1. Classification of the volatility models used in the comparison.

In the following empirical study we compare the performance of the TV-entropy approach (10)–(11) to different popular variants of GARCH models (Table 1). For the sake of simplicity, we will start with $\sigma_C = 0$. For a comparison with common econometric models we use a classic GARCH[1] model with normally distributed innovations [4] and a GARCH-PML4 model that uses the MaxEnt principle to achieve the less restrictive description of the density [15]. Additionally, we employ the GARCH-GJR model that incorporates the asymmetric influence of positive and negative news on volatility [10] and the MS-GARCH model that assumes the presence of several hidden GARCH regimes in data with the regime transitions governed by a Markov chain [2].

As benchmark problems we considered the daily percentage log-returns time series of the seven major world market indices: DJI, SPX, FTSE, STOXX, SMI, HSI, and N225.[2] The data is available in the Oxford-Man Institute's "Realized

[1] With this abbreviation we refer to the GARCH(1, 1) model.

[2] DJI (Dow Jones Industrial Average Index, United States), SPX (Standard & Poor's 500 Index, United States), FTSE (Financial Times Stock Exchange 100 Index, United Kingdom), STOXX (EURO STOXX 50 Index, European Union), SMI (Swiss Market Index, Switzerland), HSI (Hang Seng Index, Honk Kong and China), N225 (Nikkei Stock Average Index, Japan).

index	T	skewness	kurtosis
DJI	2864	0.08	10.96
SPX	2862	−0.08	10.11
FTSE	2878	−0.09	6.81
STOXX	2896	−0.12	7.61
SMI	2875	0.03	9.26
HSI	2603	0.13	16.38
N225	2773	−0.41	13.30

Table 2. Description of the data samples, where T is a sample size.

K	C_γ	N	k_i	$C_{\lambda^{(i)}}$
$\{1, 2, 3, 4\}$	$\{1, 2, 3, \ldots, 50\}$	10	6	$+\infty$

Table 3. Input parameters, where K is the number of hidden regimes, C_γ the maximum number of transitions per regime, N the number of annealing steps used during estimation, k_i a number of moment constraints in the MaxEnt regime i, and $C_{\lambda^{(i)}}$ the l_1 bound.

Library" [14].[3] The sample size and some related statistics of all considered samples are gathered in Table 2. All considered samples are not normally distributed.[4] High kurtosis points to the presence of the fat tails in data, and nonzero skewness suggests asymmetry in the distribution of the returns. These properties are generally consistent with empirical data.

To reduce the number of estimation routines needed to obtain the optimal parameters of the TV-entropy, we split the estimation procedure into two stages. First, models are estimated for all combinations of the input parameters outlined in Table 3 without l_1-regularization of the regime parameters. The optimal number of regimes (K^*) and transition upper bounds (C_γ^*) for each data set are chosen based on minimal BIC value.

Next, in order to identify the optimal number of local parameters needed to describe data in each of the regimes, we estimate TV-entropy models with various values of l_1-regularization bounds, while keeping the number of hidden regimes and transitions fixed according to their optimal values (K^*, C_γ^*) obtained at the previous step. The range of l_1 bounds $C_{\lambda^{(i)}}$ is data-dependent and varies across samples. To choose the appropriate range for each sample we analyze the corresponding unregularized solution.

[3] See https://github.com/Ganna85/TV-Entropy for the data set and source code used in numerical experiments.

[4] For the normal distribution the skewness and kurtosis should be equal to zero and four, respectively.

index	K^*	C_γ^*	k^*
DJI	3	21	[6, 6, 4]
SPX	3	18	[6, 6, 4]
FTSE	4	12	[6, 6, 6, 6]
STOXX	3	16	[6, 6, 4]
SMI	3	19	[6, 6, 4]
HSI	3	8	[6, 4, 4]
N225	3	11	[6, 6, 4]

Table 4. Best regularized TV-entropy models, where K^*, C_γ^* are the optimal values of parameters with respect to the BIC and k^* is the vectors with the optimal number of regime parameters.

As shown in Table 4, the application of the TV-entropy model reveals three hidden regimes in all of the considered benchmarks, with an exception of the FTSE index data, where the four-regime model is identified as mBIC-optimal. The obtained regime-switching dynamic appears to be persistent for all of the considered benchmarks. Particularly, the highest number of regime transitions allowed per regime is 21 in the case of DJI and the lowest number is 8 in the case of HSI. Using l_1 regularization led to identification of simpler models with respect to the number of regime parameters needed to describe the data. In all samples, except FTSE, we were able to eliminate irrelevant parameters, as reflected in the values of k^* of Table 4 representing the optimal number of the parameters in each regime. The corresponding regime densities with reduced parameters (shown in Figures 2 and 3) can be interpreted as curved exponential distributions. For the interpretation of the results it is important to note that the integration domain is finite and rescaled to the interval $[-1, 1]$ at the time of estimation. This approach allows us to resolve integrability issues otherwise present in fitting densities with maximum entropy. Apart from the MaxEnt distributions estimated by the method (in black), we fit the Gaussian densities to the data in each regime (in red). As shown in Table 2, the unconditional densities of all seven data sets are skewed with heavy tails. As seen from the comparison, the nonparametric TV-entropy densities provide a better fit for fat tails and asymmetry exhibited by the underlying densities, compared to the Gaussian distribution function.[5] This effect is most prominent in the regimes where the data points contributing to heavy mass at the tails are observed (for instance regime #3 of HSI in Figure 3).

As shown in Table 5, the memoryless TV-entropy model outperformed all of the finite-memory GARCH models across all seven benchmarks with respect to the

[5]The Gaussian density is fully described with two moments (e.g., the mean and the variance), and it corresponds to the MaxEnt distribution with the first two moment constraints ($k = 2$).

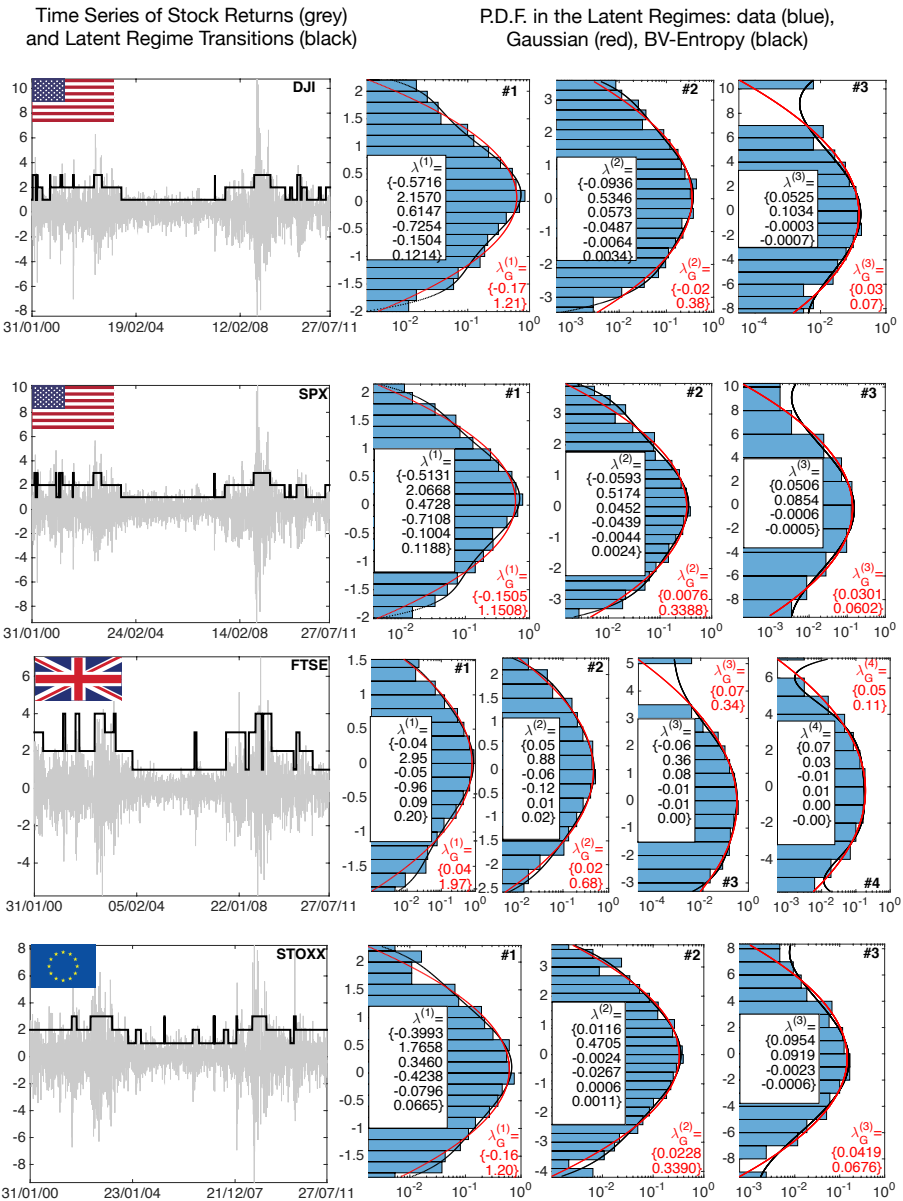

Figure 2. The optimal regime-switching paths obtained by the sparse multidimensional MaxEnt model (10)–(11) and the comparison of histograms of the corresponding regimes' data (blue bars) to the estimated MaxEnt densities (black) and fitted Gaussian densities (red) for DJI, SPX, FTSE, and STOXX index data.

log-likelihood and the BIC values. The highest log-likelihood value suggests the superior fit, while the lowest BIC value suggests the best balance between the fit and the complexity among all the considered models. As demonstrated by the posterior

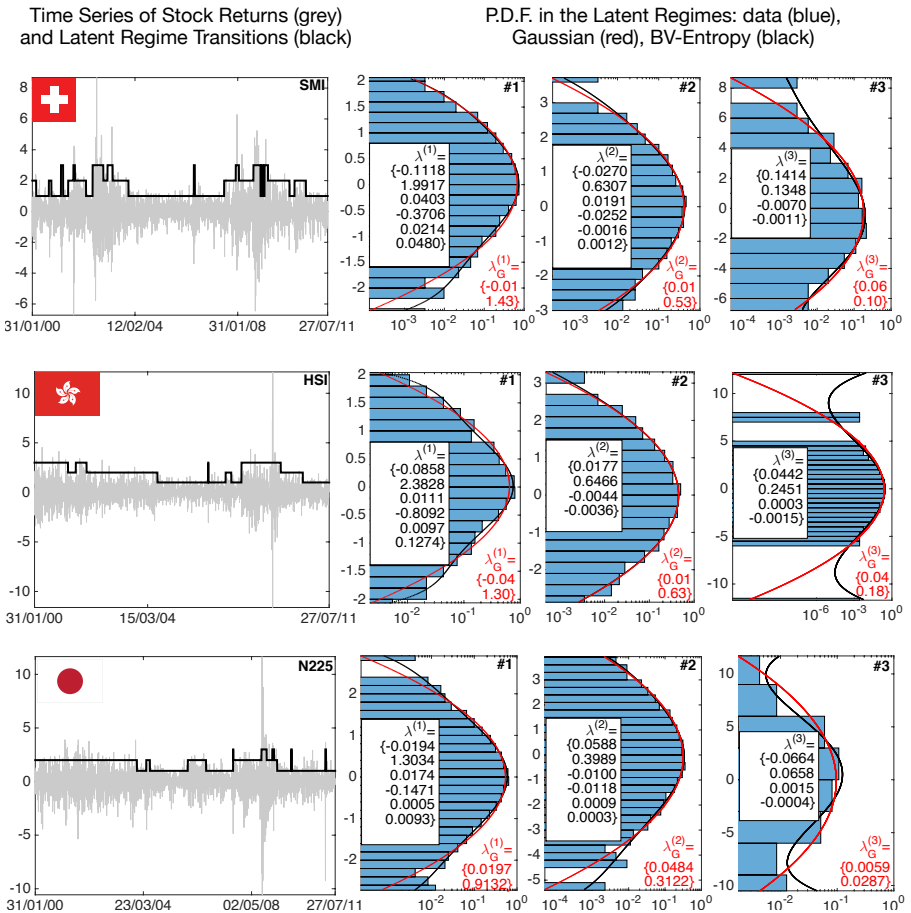

Figure 3. The optimal regime-switching paths obtained by the sparse multidimensional MaxEnt model (10)–(11) and the comparison of histograms of the corresponding regimes' data (blue bars) to the estimated MaxEnt densities (black) and fitted Gaussian densities (red) for SMI, HSI, and N225 index data.

probabilities inferred from Schwartz weights [6], the difference in BIC values is highly significant. These results indicate that the assumption of finite autoregressive memory imposed by the GARCH models is redundant for all of the considered time series data, once nonstationary regime switches are taken into account.

One of the central arguments for using the GARCH models is based on their ability to fit the autocorrelation function of $|x_t|^d$ as a function of lag times and exponents d [7]. As was shown in [7], there is little to no autocorrelation in daily returns. The highest autocorrelation is observed in the absolute returns and it is significant even at very large lags, suggesting a presence of autoregressive memory in the data. In the following we commence a simulation study where we draw

index	TV-entropy	GARCH	G-GJR	G-PML4	MS-GARCH
DJI	−3906.43	−4058.89	−4000.98	−4034.36	−4048.55
SPX	−4016.21	−4150.95	−4091.07	−4128.38	−4139.24
FTSE	−3465.43	−3580.84	−3552.85	−3563.42	−3577.93
STOXX	−4415.74	−4545.38	−4473.82	−4528.83	−4534.32
SMI	−3496.79	−3641.14	−3601.21	−3625.24	−3631.99
HSI	−3443.06	−3522.12	−3521.07	−3522.29	−3508.54
N225	−4003.36	−4100.27	−4080.22	−4067.87	−4094.51
DJI	7948.18	8141.66	8033.80	8108.52	8113.10
SPX	8167.73	8325.78	8213.99	8296.56	8294.47
FTSE	7129.99	7185.58	7137.57	7166.67	7171.86
STOXX	8966.99	9114.66	8979.52	9097.52	9084.64
SMI	7136.94	7306.16	7234.27	7290.30	7279.97
HSI	7004.08	7067.84	7073.59	7083.91	7047.08
N225	8141.50	8224.32	8192.15	8175.38	8205.02
DJI	(1.00)	(0.00)	(0.00)	(0.00)	(0.00)
SPX	(1.00)	(0.00)	(0.00)	(0.00)	(0.00)
FTSE	(0.98)	(0.00)	(0.02)	(0.00)	(0.00)
STOXX	(1.00)	(0.00)	(0.00)	(0.00)	(0.00)
SMI	(1.00)	(0.00)	(0.00)	(0.00)	(0.00)
HSI	(1.00)	(0.00)	(0.00)	(0.00)	(0.00)
N225	(1.00)	(0.00)	(0.00)	(0.00)	(0.00)

Table 5. Log-likelihood (top), BIC (middle), and posterior probability (bottom) values obtained for in-sample analysis.

1000 samples from all previously obtained optimal models. We then compute the mean values of autocorrelation coefficients at each lag and compare them to the autocorrelation coefficients obtained on real data. Specifically, we analyze the autocorrelation function of $|x_t|^d$ as a function of d and a lag as shown in the left panels of Figure 4 for SMI and STOXX index data. The summary of results for all of the considered indices is shown in Table 6.

Consistently with [7], the highest serial correlation is observed around the value $d = 1$ for all of the considered benchmarks. As can be seen from Figure 4, even at the very large lags there is a significant serial correlation as the values of the coefficients lay outside the confidence intervals (light gray) constructed under the null hypothesis that the obtained series are completely random. Results from Figure 4 demonstrate that the simpler, conditionally memoryless TV-entropy models (green) provide the closest approximation of the sample autocorrelation function (gray), as compared to all of the considered GARCH models.

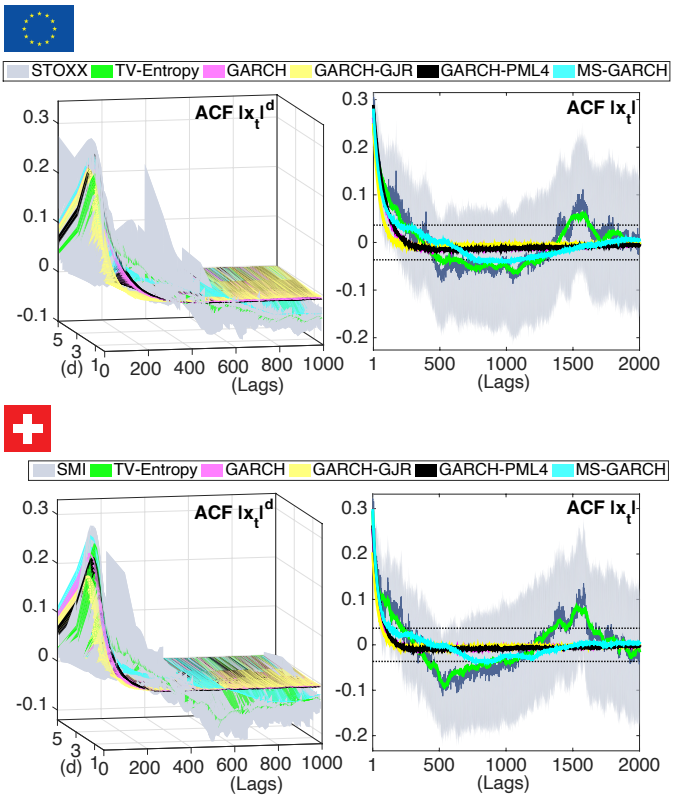

Figure 4. Autocorrelation function of the absolute returns as a function of lag, power d (left), and $d = 1$ (right) computed from the STOXX (top) and SMI (bottom) index data and the corresponding mean values inferred from the samples generated according to the optimal parameters of all considered models. The gray dashed lines are i.i.d. confidence intervals, and the gray shaded areas are confidence intervals under the assumption of the MA process.

Next, all of the obtained models will be used to produce volatility forecasts and, consequently, the one-day-ahead value-at-risk (VaR) forecasts. We compare the online VaR prediction quality for the considered methods and benchmarks. Once we observe the new data point, we confirm or correct the assigned regime value Γ using the optimal parameters obtained during the in-sample estimation. To compare the quality of the VaR forecasts, we commence the unconditional coverage test [20]. As shown in Table 7, there is no particular model that would be clearly preferred for all of the considered assets. However, since the results are not statistically distinguishable (the confidence intervals are overlapping), it can be concluded that the conditionally memoryless TV-entropy — relying on fewer tunable model parameters — is a good alternative to the traditional GARCH models in forecasting one-day-ahead VaR. Next, we analyze the relationships between the

index	TV-entropy	GARCH	GARCH-GJR	GARCH-PML4	MS-GARCH
DJI	✓ 0.00052	0.00120	0.00140	0.00088	0.00085
SPX	✓ 0.00037	0.00140	0.00180	0.00098	0.00073
FTSE	✓ 0.00024	0.00099	0.00210	0.00091	0.00065
STOXX	✓ 0.00034	0.00120	0.00170	0.00110	0.00073
SMI	✓ 0.00030	0.00190	0.00220	0.00180	0.00110
HSI	✓ 0.00036	0.00086	0.00093	0.00085	0.00062
N225	✓ 0.00041	0.00060	0.00063	0.00058	0.00084

Table 6. The mean squared error between sample and simulated autocorrelation coefficients of $|x_t|^d$ of the first 1000 lags for all considered models; ✓ denotes models with the best fit of the data autocorrelation.

index	TV-entropy	GARCH	G-GJR	G-PML4	MS-GARCH
DJI	0.044	0.042	0.043	0.041	0.043
SPX	0.045	0.045	0.047	0.042	0.046
FTSE	0.046	0.059	0.059	0.055	0.057
STOXX	0.054	0.052	0.052	0.048	0.052
SMI	0.048	0.049	0.051	0.047	0.049
HSI	0.056	0.054	0.051	0.054	0.015
N225	0.044	0.047	0.051	0.044	0.046
DJI	0.013	0.018	0.015	0.014	0.018
SPX	0.013	0.017	0.018	0.014	0.017
FTSE	0.015	0.021	0.023	0.017	0.021
STOXX	0.015	0.017	0.022	0.014	0.017
SMI	0.014	0.018	0.019	0.014	0.016
HSI	0.017	0.020	0.019	0.020	0.003
N225	0.013	0.016	0.017	0.012	0.019

Table 7. Unconditional coverage test results for 95% and 99% VaR. The expectation of VaR violations should be close to 5% and 1%, respectively.

latent volatility transitions inferred by TV-entropy (see the left panels of Figures 2 and 3). Identified MaxEnt regimes are characterized by different volatility levels. As a result, the regime transitions are jumps indicating an increase or a decrease of the volatility level. For every considered asset we construct two distinct binary variables describing such a behavior, where zeros indicate no jump at a current time instance, and ones stand for transition to the regime with only increased ("up") or only decreased ("down") volatility. We then perform a pairwise comparison of the obtained categorical time series (characterized by "up/up", "up/down", "down/up", and "down/down" directions) using Fisher's exact test [8] and analyze the resulting p-values to identify the statistically significant relations between the regime transitions.

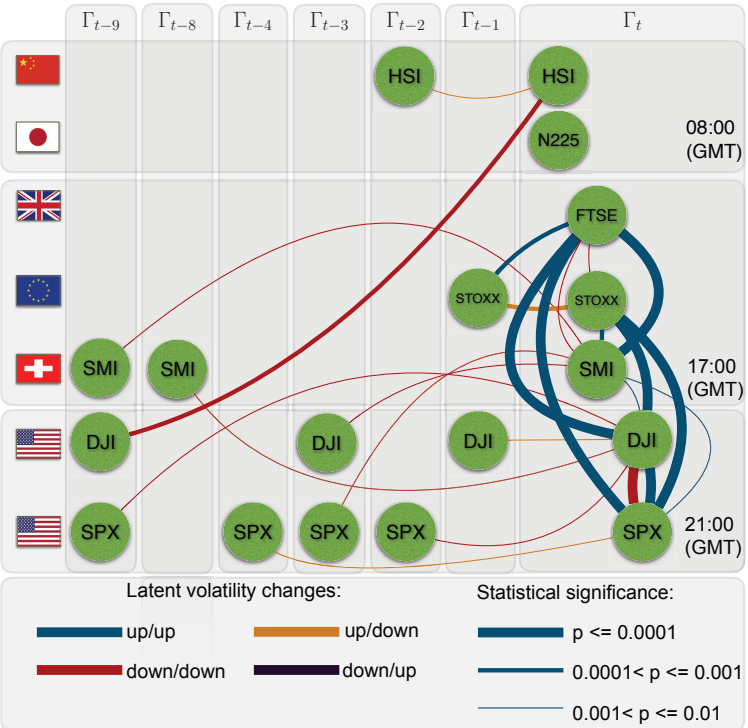

Figure 5. Latent structure relation graph, showing how the regime transitions are correlated between assets over time. The connection lines are scaled according to the p-value obtained by Fisher's exact test from the vales of latent regime variables Γ obtained with (10)–(11).

Shown in Figure 5 are only the most significant relationships ($p \leq 0.01$) among these categorical time series of latent regime transitions. The assets in the American and in the European regions appear to be heavily connected through their latent regime transitions. These connections are mostly governed by the volatility increase on the short time scale and volatility decrease on longer time scales. As for the considered Asian markets, the inferred latent influence of the American and European markets is present but it appears to be delayed in time and driven mostly by a decrease in the latent volatility levels.

Discussion

In this work we presented a sparse extension of the nonstationary MaxEnt methodology from one to multiple dimensions, aiming to identify the most qualitative (in terms of the log-likelihood) and the least complex (in terms of the information content and the required number of tunable parameters) representation for multidimensional time series data.

In an application to analysis of financial time series, we show that one of the important distinctions between the entropy-based and the common heteroscedastic approaches used in economics and finance is an assumption about the finite autoregressive memory in the underlying data-generating process. Traditionally, the observed data is assumed to be explicitly dependent on its past realizations. In the presented application of the MaxEnt framework, we do not impose additional assumptions about memory and do not include tunable parameters that describe it. The hypothesis that realizations are independent within the regimes has previously been explored in [5] and [28] in the context of parametric HMMs, where the authors showed that proposed models can reproduce the empirical properties of daily returns, especially in the case when conditional distributions are not normally distributed. However, these approaches also impose an explicit a priori memory assumption (Markovianity) on the level of the regime-switching process, and it remains unclear whether this assumption is necessary and/or sufficient for realistic financial data. The TV-entropy approach reveals that the volatility for all of the considered benchmarks is best described by the time-dependent persistent process, where persistence is identified through the adaptive regularization (11) of the regime-switching process and from the fact that within every regime the volatility remains stationary and i.i.d.

Starting with the intrinsically multivariate but ill posed MaxEnt formulation in (1)–(2), we derived that by introducing two (mild) Assumptions 1 and 2 its solution can be approximated from below with a well posed solution of the sparse regularized problem (10)–(11). Problem (10)–(11) is still multivariate since the hidden regime variables $\gamma_{t,i}^{(k)}$ change with the dimension i, time t, and regime k. As shown in Figure 5, they capture the latent multivariate relation structure over different dimensions and times.

In this manuscript we analyzed seven leading world market indices across America, Europe, and Asia. We found that the nonparametric TV-entropy approach outperforms all of the considered benchmark models for in-sample analysis in terms of the log-likelihood, simplicity (the number of free parameters), the information content (BIC and the posterior model probabilities), and the quality when describing the underlying autocorrelation function behavior. The out-of-sample study indicates that TV-entropy methodology is an effective alternative to the GARCH models for forecasting of the one-day-ahead VaR. The TV-entropy approach could closely reproduce serial correlation patterns found in data, especially at the large lags (unlike any of the GARCH models considered). This study indicates that nonstationary MS-GARCH (combining GARCH with the Markovian regime transition model) allows for a better description of the serial correlation than single-regime GARCH models, but it is not able to match the data as accurately as the TV-entropy model. Finally, Figure 5 illustrates how the regime transition processes inferred from the data allows us to identify the statistically significant temporal relations between

latent volatility level transitions across different markets. In particular, these findings indicate that negative news has a stronger short-term impact on the markets, as we observe that statistically most-significant latent connections are associated with the increase in volatility levels.

References

[1] N. Agmon, Y. Alhassid, and R. D. Levine, *An algorithm for finding the distribution of maximal entropy*, J. Comput. Phys. **30** (1979), no. 2, 250–258. Zbl

[2] L. Bauwens, A. Dufays, and J. V. K. Rombouts, *Marginal likelihood for Markov-switching and change-point GARCH models*, J. Econometrics **178** (2014), no. 2–3, 508–522. MR Zbl

[3] A. L. Berger, V. J. Della Pietra, and S. A. Della Pietra, *A maximum entropy approach to natural language processing*, Comput. Linguist. **22** (1996), no. 1, 39–71.

[4] T. Bollerslev, *Generalized autoregressive conditional heteroskedasticity*, J. Econometrics **31** (1986), no. 3, 307–327. MR Zbl

[5] J. Bulla and I. Bulla, *Stylized facts of financial time series and hidden semi-Markov models*, Comput. Statist. Data Anal. **51** (2006), no. 4, 2192–2209. MR Zbl

[6] K. P. Burnham and D. R. Anderson, *Model selection and multimodel inference: a practical information-theoretic approach*, 2nd ed., Springer, 2002. MR Zbl

[7] Z. Ding, C. W. J. Granger, and R. F. Engle, *A long memory property of stock market returns and a new model*, J. Empir. Financ. **1** (1993), no. 1, 83–106.

[8] R. A. Fisher, *On the interpretation of χ^2 from contingency tables, and the calculation of p*, J. R. Stat. Soc. **85** (1922), no. 1, 87–94.

[9] S. Gerber and I. Horenko, *Improving clustering by imposing network information*, AAAS Sci. Adv. **1** (2015), no. 7, e1500163.

[10] L. R. Glosten, R. Jagannathan, and D. E. Runkle, *On the relation between the expected value and the volatility of the nominal excess return on stocks*, J. Financ. **48** (1993), no. 5, 1779–1801.

[11] A. Golan, G. Judge, and J. M. Perloff, *A maximum entropy approach to recovering information from multinomial response data*, J. Amer. Statist. Assoc. **91** (1996), no. 434, 841–853. MR Zbl

[12] L. P. Hansen, *Large sample properties of generalized method of moments estimators*, Econometrica **50** (1982), no. 4, 1029–1054. MR Zbl

[13] W. Härdle, *Applied nonparametric regression*, Econometric Society Monographs, no. 19, Cambridge University, 1990. MR Zbl

[14] G. Heber, A. Lunde, N. Shephard, and K. Sheppard, *Oxford-Man Institute's Realized Library*, Oxford University, 2009.

[15] A. Holly, A. Monfort, and M. Rockinger, *Fourth order pseudo maximum likelihood methods*, J. Econometrics **162** (2011), no. 2, 278–293. MR Zbl

[16] I. Horenko, *Finite element approach to clustering of multidimensional time series*, SIAM J. Sci. Comput. **32** (2010), no. 1, 62–83. MR Zbl

[17] ———, *On the identification of nonstationary factor models and their application to atmospheric data analysis*, J. Atmos. Sci. **67** (2010), no. 5, 1559–1574.

[18] E. T. Jaynes, *Information theory and statistical mechanics*, Phys. Rev. (2) **106** (1957), 620–630. MR Zbl

[19] R. Kohavi, *A study of cross-validation and bootstrap for accuracy estimation and model selection*, IJCAI '95: proceedings of the 14th International Joint Conference on Artificial Intelligence, vol. 2, Morgan Kaufmann, San Francisco, 1995, pp. 1137–1143.

[20] P. H. Kupiec, *Techniques for verifying the accuracy of risk measurement models*, J. Deriv. **3** (1995), no. 2, 73–84.

[21] C. F. Manski and D. McFadden (eds.), *Structural analysis of discrete data with econometric applications*, MIT, Cambridge, MA, 1981. MR Zbl

[22] G. Marchenko, P. Gagliardini, and I. Horenko, *Towards a computationally tractable maximum entropy principle for nonstationary financial time series*, SIAM J. Financial Math. **9** (2018), no. 4, 1249–1285. MR Zbl

[23] L. R. Mead and N. Papanicolaou, *Maximum entropy in the problem of moments*, J. Math. Phys. **25** (1984), no. 8, 2404–2417. MR

[24] T. Mora, A. M. Walczak, W. Bialek, and C. G. Callan, Jr., *Maximum entropy models for antibody diversity*, P. Natl. Acad. Sci. USA **107** (2010), no. 12, 5405–5410.

[25] K. Nigam, J. Lafferty, and A. McCallum, *Using maximum entropy for text classification*, IJCAI '99: workshop on machine learning for information filtering, 1999, pp. 61–67.

[26] S. J. Phillips, R. P. Anderson, and R. E. Schapire, *Maximum entropy modeling of species geographic distributions*, Ecol. Model. **190** (2006), no. 3–4, 231–259.

[27] L. Pospíšil, P. Gagliardini, W. Sawyer, and I. Horenko, *On a scalable nonparametric denoising of time series signals*, Commun. Appl. Math. Comput. Sci. **13** (2018), no. 1, 107–138. MR Zbl

[28] T. Rydén, T. Teräsvirta, and S. Åsbrink, *Stylized facts of daily return series and the hidden Markov model*, J. Appl. Economet. **13** (1998), no. 3, 217–244.

[29] R. Tibshirani, *Regression shrinkage and selection via the lasso*, J. Roy. Statist. Soc. Ser. B **58** (1996), no. 1, 267–288. MR Zbl

[30] X. Wu, *Calculation of maximum entropy densities with application to income distribution*, J. Econometrics **115** (2003), no. 2, 347–354. MR Zbl

[31] A. Zellner and R. A. Highfield, *Calculation of maximum entropy distributions and approximation of marginal posterior distributions*, J. Econometrics **37** (1988), no. 2, 195–209. MR

Received October 8, 2019. Revised May 6, 2020.

ILLIA HORENKO: horenkoi@usi.ch
Institute of Computational Science, Facoltà di scienze informatiche, Università della Svizzera italiana, Lugano, Switzerland

GANNA MARCHENKO: ganna.marchenko@usi.ch
Institute of Computational Science, Facoltà di scienze informatiche, Università della Svizzera italiana, Lugano, Switzerland

PATRICK GAGLIARDINI: patrick.gagliardini@usi.ch
Facoltà di scienze economiche Università della Svizzera italiana, Lugano, Switzerland

and

Swiss Finance Institute, Zürich, Switzerland

COMM. APP. MATH. AND COMP. SCI.
Vol. 15, No. 2, 2020
https://doi.org/10.2140/camcos.2020.15.147

FAST MULTIGRID SOLUTION OF HIGH-ORDER ACCURATE MULTIPHASE STOKES PROBLEMS

ROBERT I. SAYE

A fast multigrid solver is presented for high-order accurate Stokes problems discretized by local discontinuous Galerkin (LDG) methods. The multigrid algorithm consists of a simple V-cycle, using an elementwise block Gauss–Seidel smoother. The efficacy of this approach depends on the LDG pressure penalty stabilization parameter — provided the parameter is suitably chosen, numerical experiment shows that (i) for steady-state Stokes problems, the convergence rate of the multigrid solver can match that of classical geometric multigrid methods for Poisson problems and (ii) for unsteady Stokes problems, the convergence rate further accelerates as the effective Reynolds number is increased. An extensive range of two- and three-dimensional test problems demonstrates the solver performance as well as high-order accuracy — these include cases with periodic, Dirichlet, and stress boundary conditions; variable-viscosity and multiphase embedded interface problems containing density and viscosity discontinuities several orders in magnitude; and test cases with curved geometries using semiunstructured meshes.

1. Introduction

Stokes flow describes the motion of an incompressible viscous fluid at slow speeds, or small scales, and can be used to model a wide range of intricate phenomena, including mantle dynamics, the swimming of microorganisms, the sedimentation of particulates, and the flotation of water droplets in clouds. In the steady-state case, the corresponding governing equations of motion are given by the Stokes equations, which generally take on one of two forms: either

$$\begin{aligned} -\mu\nabla^2\boldsymbol{u} + \nabla p &= \boldsymbol{f}, \\ \nabla\cdot\boldsymbol{u} &= 0, \end{aligned} \tag{1}$$

or, alternatively,

$$\begin{aligned} -\nabla\cdot(\mu(\nabla\boldsymbol{u} + \nabla\boldsymbol{u}^{\mathsf{T}})) + \nabla p &= \boldsymbol{f}, \\ \nabla\cdot\boldsymbol{u} &= 0, \end{aligned} \tag{2}$$

MSC2010: 65N55, 76D07, 65N30, 65F08.
Keywords: Stokes equations, multigrid, high order, multiphase, discontinuous Galerkin methods.

where μ specifies the viscosity of the fluid, $\boldsymbol{u}$ and p describe its velocity and pressure fields, and $\boldsymbol{f}$ specifies the net external forces acting on the fluid. The form given in (1) is here referred to as the *standard form* of the Stokes equations and (2) as the *viscous-stress form*, with the particular choice depending on the end application. For example, the viscous-stress form is generally applicable when μ is variable or if boundary conditions on stress are imposed.

Our motivation in this work is to develop fast multigrid solvers for computing high-order accurate solutions of the Stokes systems (1) or (2), with extensions also to multiphase variants involving interfacial jump conditions in velocity and stress, as well as to time-dependent problems. In particular, we consider a framework based on local discontinuous Galerkin (LDG) methods [15], and build on prior work developing efficient multigrid algorithms for LDG discretizations of elliptic interface (Poisson-like) problems [20; 41]. We show that standard and simple-to-implement geometric multigrid algorithms can be applied to the resulting multiphase Stokes problems — in the steady-state case, results show that the solver can match the speeds of fast geometric multigrid methods for Poisson problems; in the time-dependent case, convergence rates further accelerate as the effective Reynolds number is increased.

In particular, the presented multigrid algorithm consists of a standard V-cycle using an elementwise block Gauss–Seidel smoother — individual blocks correspond to individual mesh elements, such that the elemental degrees of freedom of both velocity and pressure are collected into the same block. Key to the rapid convergence of this approach is a suitable choice of the pressure penalty stabilization parameter underlying the LDG framework — if the parameter is chosen well, then a highly efficient multigrid algorithm is obtained. We discuss how to choose this parameter for steady-state Stokes problems, and develop a simple strategy for generalizing this choice to time-dependent Stokes problems. Extensive tests of the multigrid methods are presented in this paper, including problems which impose Dirichlet or stress boundary conditions, variable-viscosity problems, test cases with curved geometry using semiunstructured meshes, and multiphase embedded interface problems with viscosity and density coefficients exhibiting discontinuities several orders in magnitude.

Previous work. A vast amount of work in computational science and engineering has been devoted to the efficient solution of Stokes systems and saddle point problems in general; for an in-depth review of the correspondingly wide array of different approaches and their applications, see the review [5] of Benzi, Golub, and Liesen. These approaches include, among others, block preconditioner methods, which operate on the viscosity, gradient, and divergence operator block structure of the Stokes equations; Schur complement methods, which manipulate, and usually

approximate, the Schur complement of the saddle point system; and stationary iterative methods, such as the well-known Uzawa method, which alternates between updates of velocity and pressure, holding the other fixed, through the two governing equations in (1). Here, we briefly review work on multigrid-style methods, particularly schemes based on the coupled solution of velocity and pressure, as is relevant to the present work; see also the reviews [47; 48; 34].

As mentioned, the multigrid algorithm developed here uses a block Gauss–Seidel relaxation method, with each block collecting the velocity and pressure degrees of freedom on each mesh element. This approach is similar in essence to the "symmetric coupled Gauss–Seidel method" of Vanka [45; 46] and can also be considered as a kind of "box relaxation" scheme [7]. Vanka-type smoothers, originally devised for staggered grid finite difference methods, visit each grid cell, solve for the velocity and pressure unknowns simultaneously via a local Stokes-like problem, and then move onto the next cell. Typically, a damping/under-relaxation parameter is needed to ensure convergence. In the original Vanka method, the Stokes system is restricted to the local variables in each grid cell and off-diagonal entries of the viscous operator are zeroed out to facilitate a simpler update for the unknowns [47; 33]. Variations have led to schemes which include the off-diagonal terms [44; 6] and line-based sweeping methods [44; 36; 35], and have been examined with local mode analyses [43]. Vanka-type smoothers may also be considered as iterative Schwarz solvers, whereby the subdomains of the Schwarz method correspond to the collection of degrees of freedom in each grid cell. Schemes building on this idea have since been developed for finite volume and finite element methods with much of the attention devoted to the choice of relaxation parameters, the choice of subdomains (e.g., whether to use one cell, or patches of cells), and theoretical proofs of convergence in a multigrid setting; see, e.g., [42; 30; 34; 22; 9; 25; 16]. They have also found applications in variable-viscosity Stokes problems [6] and in computational solid mechanics [49; 27]. As an example, in very recent work, Farrell, He, and MacLachlan [19] demonstrated the application of local Fourier analysis on these smoothers and found that smaller patches result in better convergence per floating point operation. Meanwhile, solvers specific to discontinuous Galerkin methods of the Stokes equations have also been devised; here, one possible approach is to exactly enforce the divergence constraint across the multigrid hierarchy through manipulation of the DG spaces, e.g., through $H(\mathrm{div}, \Omega)$-conforming discretizations. In the associated multigrid solvers, the divergence constraint is built into the coarse and fine mesh approximations; see, e.g., [9; 29; 27; 1; 16].

In many of these works, satisfactory multigrid convergence rates are reported, but they generally do not match the speeds of an efficient geometric multigrid method designed for scalar elliptic equations. In some cases, performance degrades as the mesh is refined or as viscosity ratios increase or, in the case of time-dependent Stokes

problems, as the Reynolds number changes. In contrast, for the LDG schemes devised here, we found that a simple block Gauss–Seidel relaxation method, which does not use any under- or over-relaxation parameters, can result in rapid multigrid convergence across a variety of challenging Stokes problems. Another advantage to a block Gauss–Seidel method is its simple implementation and the possibility of parallelism; for example, Gmeiner et al. [23] and Bauer et al. [4] demonstrate massively parallel and GPU implementations, respectively. For example, some of the three-dimensional tests in this paper used half a billion degrees of freedom and scaled to several hundred computing nodes, though we do not report on scaling performance here.

Outline. In the main article, the central ideas and results are presented, while nonessential details of the LDG discretization, grid convergence analyses, and implementation possibilities are deferred to the appendices. First, we outline the essential components of the LDG framework for the multiphase Stokes equations (Section 2). Second, the design of a standard multigrid V-cycle is outlined (Section 3), after which the role of pressure penalty stabilization on multigrid efficiency is examined (Section 4). Then results are presented for a variety of problems for the steady-state and time-dependent Stokes equations (Sections 5 and 6, respectively). We then conclude, summarizing the key observations made in this work along with a discussion of future research avenues (Section 7).

2. Local discontinuous Galerkin methods for multiphase Stokes problems

In this work, we build on the LDG schemes developed by Cockburn et al. [14] and extend them to the variable-viscosity multiphase Stokes problem. The governing equations are written as follows: we seek to determine a velocity field $\boldsymbol{u} : \Omega \to \mathbb{R}^d$ and pressure field $p : \Omega \to \mathbb{R}$ such that

$$\left.\begin{aligned} -\nabla \cdot (\mu_i(\nabla \boldsymbol{u} + \gamma \nabla \boldsymbol{u}^{\mathsf{T}})) + \nabla p &= \boldsymbol{f} \\ -\nabla \cdot \boldsymbol{u} &= f \end{aligned}\right\} \quad \text{in } \Omega_i, \tag{3}$$

subject to the interfacial jump conditions

$$\left.\begin{aligned} [\![\boldsymbol{u}]\!] &= \boldsymbol{g}_{ij} \\ [\![\mu(\nabla \boldsymbol{u} + \gamma \nabla \boldsymbol{u}^{\mathsf{T}})\boldsymbol{n} - p\boldsymbol{n}]\!] &= \boldsymbol{h}_{ij} \end{aligned}\right\} \quad \text{on } \Gamma_{ij} \tag{4}$$

and boundary conditions

$$\begin{aligned} \boldsymbol{u} &= \boldsymbol{g}_\partial \quad \text{on } \Gamma_D, \\ \mu(\nabla \boldsymbol{u} + \gamma \nabla \boldsymbol{u}^{\mathsf{T}})\boldsymbol{n} - p\boldsymbol{n} &= \boldsymbol{h}_\partial \quad \text{on } \Gamma_N, \end{aligned} \tag{5}$$

where Ω is a domain in $\mathbb{R}^d$ divided into one or more subdomains Ω_i (denoted "phases"), $\Gamma_{ij} := \partial\Omega_i \cap \partial\Omega_j$ is the interface between phases i and j, and Γ_D and Γ_N

denote the parts of $\partial\Omega$ on which velocity Dirichlet or stress boundary conditions are imposed, respectively. Here, either $\gamma = 0$ or $\gamma = 1$ depending on whether the Stokes equations are in standard form or viscous-stress form, respectively. The operator $[\![\cdot]\!]$ denotes the jump in a quantity across an interface and $\boldsymbol{n}$ is to be understood from context — on $\partial\Omega$, $\boldsymbol{n}$ denotes the outward unit normal to the domain boundary, whereas for an interface Γ_{ij}, $\boldsymbol{n}$ denotes the unit normal to Γ_{ij}, oriented consistently with the definition of the jump operator. Finally, μ_i is a phase-dependent viscosity coefficient, while $\boldsymbol{f}$, f, $\boldsymbol{g}$, and $\boldsymbol{h}$ provide the data to the multiphase Stokes problem and are given functions defined on Ω, its boundary, and internal interfaces.

Here, we mainly consider meshes arising from Cartesian grids along with semi-unstructured quadtree/octree-based implicitly defined meshes of more complex curved domains. In this setting, it is natural to adopt a tensor product piecewise polynomial space. Let $\mathscr{E} = \bigcup_i E_i$ denote the set of mesh elements, let $p \geq 1$ be an integer,[1] and define $\mathbb{Q}_p(E)$ as the space of tensor product polynomials of (one-dimensional) degree p on element E. For example, $\mathbb{Q}_2$ is the space of biquadratic or triquadratic polynomials, with dimension 9 or 27 in 2D or 3D, respectively. Define the corresponding space of discontinuous piecewise polynomial functions as

$$V_h = \left\{ u : \Omega \to \mathbb{R} \;\middle|\; u|_E \in \mathbb{Q}_p(E) \text{ for every } E \in \mathscr{E} \right\},$$

with analogous definitions for the space of piecewise polynomial vector-valued fields, V_h^d, and the space of matrix-valued fields, $V_h^{d\times d}$. As discussed in [14], it is possible to build LDG methods for the Stokes equations wherein the discrete pressure field has either the same polynomial degree as the discrete velocity field or is in a lower-degree space. In this work, we focus on the case that the two have the same degree; i.e., we seek a discrete solution such that $\boldsymbol{u}_h \in V_h^d$ and $p_h \in V_h$.

In one possible construction of the LDG framework, the governing set of equations (3)–(5) can be discretized in a three-step process: (i) define a discrete stress tensor $\boldsymbol{\tau}_h \in V_h^{d\times d}$ equal to the discretization of $\nabla \boldsymbol{u}_h + \gamma \nabla \boldsymbol{u}_h^{\mathsf{T}}$, taking into account Dirichlet source data $\boldsymbol{g}$, (ii) define $\boldsymbol{\sigma}_h \in V_h^{d\times d}$ as the viscous stress $\mu\boldsymbol{\tau}_h - p_h \mathbb{I}$ via an L^2 projection of $\mu\boldsymbol{\tau}_h$ onto $V_h^{d\times d}$, and (iii) compute a discrete divergence of $\boldsymbol{\sigma}_h$, taking into account Neumann-like data $\boldsymbol{h}$, and add penalty stabilization parameters for both velocity and pressure, setting the result equal to the L^2 projection of the given right-hand side $\boldsymbol{f}$. Details of this construction, along with the associated treatment of the divergence constraint, are provided in Appendix A; here, we summarize the main outcomes of essential relevance. The LDG discretization results in a symmetric linear system for $(\boldsymbol{u}_h, p_h)$ having the form

$$\begin{pmatrix} A & M\mathcal{G} \\ \mathcal{G}^{\mathsf{T}} M & -E \end{pmatrix} \begin{pmatrix} \boldsymbol{u}_h \\ p_h \end{pmatrix} = \begin{pmatrix} \boldsymbol{b}_{\boldsymbol{u}} \\ b_p \end{pmatrix} \tag{6}$$

[1] The meaning of p, whether as pressure or polynomial degree, should be clear from context.

where M is the block-diagonal mass matrix, $\mathcal{G}$ is a discrete gradient operator, and $(\boldsymbol{b_u}, b_p)$ collects the entire influence of the source data $\boldsymbol{f}$, f, $\boldsymbol{g}$, and $\boldsymbol{h}$ onto the right-hand side. Here, A implements the viscous part of the Stokes momentum equations, and can be written in $d \times d$ block form corresponding to its action on the d components of $\boldsymbol{u}_h$, with the (i, j)-th block given by

$$A_{ij} = \delta_{ij}\left(\sum_{k=1}^{d} G_k^{\mathsf{T}} M_\mu G_k\right) + \gamma G_j^{\mathsf{T}} M_\mu G_i + \delta_{ij}\widetilde{E},$$

where M_μ is a μ-weighted mass matrix, G is a second discrete gradient operator closely related to the adjoint of $\mathcal{G}$, and $\widetilde{E}$ is the operator associated with velocity penalty stabilization. Note that if $\gamma = 0$, then A is block diagonal with identical blocks corresponding to a discretization of the Laplacian operator $-\nabla \cdot (\mu \nabla)$. Meanwhile, noting that the adjoint of $\mathcal{G}$ is given by $M^{-1}\mathcal{G}^{\mathsf{T}} M$, one may observe that the divergence constraint of the Stokes equations is implemented in the $(p, \boldsymbol{u})$ block of (6) through an effective discrete divergence operator which is equal to the negative adjoint of $\mathcal{G}$.

There is one last operator to define in (6), whose presence is of key importance to multigrid efficiency: the pressure stabilization operator E, which weakly enforces continuity of the pressure field. The symmetric positive semidefinite matrix E is defined such that[2]

$$u^{\mathsf{T}} E v = \int_{\Gamma_0} \tau_p [\![u]\!][\![v]\!] \tag{7}$$

holds for every $u, v \in V_h$. Here, the integral is taken over the union of every noninterfacial interior mesh face, $[\![\cdot]\!]$ denotes the jump across the face, and τ_p is a *pressure stabilization penalty parameter* which scales proportionally to the element size h and inversely proportionally to the (local) viscosity coefficient:

$$\tau_p = \tau h / \mu, \tag{8}$$

where τ is a user-defined constant prefactor. Provided τ is positive, Cockburn et al. [14] (see also extensions [11; 13]) prove the well-posedness of the single-phase symmetric saddle point problem in (6), including satisfaction of the inf-sup conditions. In these cited works, however, the particular choice of τ in (8) is not extensively discussed. One of the main results in this work is to demonstrate that τ can be chosen so as to achieve excellent multigrid solver efficiency when computing solutions to the Stokes equations.

[2] In a convenient abuse of notation, a piecewise polynomial function (e.g., in V_h) may carry the same notation as its corresponding coefficient vector in the basis of V_h, with the precise meaning understood from context. For example, in the identity $u^{\mathsf{T}} M v = \int_\Omega uv$, the left-hand side employs vectors and matrices relative to the chosen basis of V_h, whereas the right-hand side employs the functional form.

In the remainder of this paper, we will refer to the Stokes problem both in its operator block form (6) and through the more succinct notation

$$\mathcal{A}_h x_h = b_h,$$

where $\mathcal{A}_h$ is the symmetric saddle point operator and x_h collects $\boldsymbol{u}_h$ and p_h into one set of unknowns.

3. Multigrid design

Prior work on designing geometric multigrid methods for LDG discretizations of Poisson-like equations [38; 39; 20; 41] shows that one can build an efficient solver through standard multigrid concepts: a V-cycle applied to a mesh hierarchy using straightforward interpolation and restriction operators, together with standard relaxation methods, such as block Gauss–Seidel in which each block corresponds to the collective set of unknowns on each mesh element. Here, we show the same can be done for the discretized Stokes problem (6). (In the following, it is assumed the reader is familiar with the general design of multigrid methods; see, e.g., the books [8; 7; 47; 33] for reviews and applications.)

The multigrid methods designed here may be considered as a "purely geometric" approach, wherein the Stokes problem is discretized on each level of the mesh hierarchy. (A convenient strategy for constructing the coarse mesh problems — without having to explicitly form the coarse meshes themselves — is discussed shortly.) Three preliminary ingredients are needed to specify its design:

- *Mesh hierarchy.* In this work, quadtrees and octrees are used to define the finest mesh. The tree structure naturally defines a hierarchical procedure by which to agglomerate elements to create a nested mesh hierarchy, coarsening by a factor of two in each dimension down each level. Regarding the multiphase case, element agglomeration is permitted only between elements of the same phase — as such, the interface is sharply preserved throughout the entire multigrid hierarchy.
- *Interpolation operator.* Owing to the presence of a nested mesh hierarchy, the interpolation operator I_{2h}^h, which transfers coarse mesh corrections to a fine mesh, is naturally defined via injection. In particular, we define $(I_{2h}^h u)|_{E_f} = u|_{E_c}$, where E_f is a fine mesh element and $E_c \supseteq E_f$ is its corresponding coarse mesh element.
- *Restriction operator.* The restriction operator R_h^{2h}, which transfers the residual of a fine mesh problem to the coarse mesh, is defined as the L^2 projection onto the coarse mesh (or, equivalently, as the adjoint of the interpolation operator). It is related to the interpolation operator via $R_h^{2h} = M_{2h}^{-1}(I_{2h}^h)^{\mathsf{T}} M_h$, where M_h and M_{2h} are the mass matrices of the two meshes.

The last essential, and perhaps most important, multigrid ingredient is the relaxation method. As mentioned earlier, we have used a simple block Gauss–Seidel method, where each block corresponds to the collective set of degrees of freedom (i.e., velocity and pressure combined) on each mesh element. Specifically, consider a repartitioning of $\mathscr{A}x = b$ according to these blocks, such that x_i denotes the set of velocity and pressure values on element i, and $\mathscr{A}_{ij}$ denotes the (i, j)-th block of $\mathscr{A}$, whence $b_i = \sum_j \mathscr{A}_{ij} x_j$. Then, the block Gauss–Seidel method simply sweeps over the elements, in some particular order, replacing $x_i \leftarrow \mathscr{A}_{ii}^{-1}\big(b_i - \sum_{j\neq i} \mathscr{A}_{ij} x_j\big)$. Here, $\mathscr{A}_{ii}$ is the i-th diagonal block of $\mathscr{A}$ and takes on the form of a miniature Stokes operator; referring to (6), we have

$$\mathscr{A}_{ii} = \begin{pmatrix} A_{ii} & M_{ii}\mathcal{G}_{ii} \\ \mathcal{G}_{ii}^{\mathsf{T}} M_{ii} & -E_{ii} \end{pmatrix}.$$

Note that $\mathscr{A}_{ii}$ needs to be inverted in the Gauss–Seidel update of element i. Assuming that the global Stokes saddle point problem $\mathscr{A}x = b$ satisfies the inf-sup conditions, it is straightforward to show that so too does $\mathscr{A}_{ii}$, and hence the local elementwise problem is well-posed; this has also been confirmed through numerous and extensive numerical tests. In our specific implementation, we precompute a symmetric indefinite factorization of $\mathscr{A}_{ii}$ for every i, and use this factorization as a direct solver for each of these mini-Stokes problems in the Gauss–Seidel sweep. Regarding the element ordering, we have opted for a multicolored Gauss–Seidel method. The primary reason for this choice is that a multicolored sweep affords a simpler parallel implementation of the method, both in terms of multithreading and in a distributed environment (e.g., through standard domain decomposition methods using MPI).

Using the defined interpolation and restriction operators and the block Gauss–Seidel relaxation method, the construction of a multigrid V-cycle is relatively standard and is outlined in Algorithm 1.[3] In this algorithm, $\mathscr{A}_h$ is assumed to be precomputed on every level of the mesh; a particularly convenient method for doing so — without having to explicitly mesh each level, build quadrature rules for coarse mesh elements, or build LDG operators via coarse mesh numerical fluxes, etc. — uses the operator coarsening ideas of [20]. In this technique, the discrete gradient and penalty operators underlying (6) are coarsened solely based on the interpolation operator hierarchy, using simple block-sparse linear algebra; these methods are

[3]Note that $(I_{2h}^h)^{\mathsf{T}}$ appears on line 5, rather than the restriction operator R_h^{2h}; this follows from a convenient simplification common to many finite element methods: briefly, viewed as an operator which maps $V_h^d \otimes V_h$ to $V_h^d \otimes V_h$, the discrete Stokes operator is given by $M^{-1}\mathscr{A}$. Therefore, the residual of the fine mesh problem, as a piecewise polynomial function, is $M_h^{-1}(b_h - \mathscr{A}_h x_h)$. This residual is then multiplied by R_h^{2h} to define the source data for the coarse mesh problem $M_{2h}^{-1}\mathscr{A}_{2h} = R_h^{2h} M_h^{-1}(b_h - A_h x_h)$. Rearranging, one obtains line 5.

if $\mathscr{C}_h$ is the bottom level **then**
  Solve $\mathscr{A}_h x_h = b_h$ with bottom solver
else
  Apply block Gauss–Seidel relaxation ν_1 times
  $r_{2h} := (I_{2h}^h)^\mathsf{T}(b_h - \mathscr{A}_h x_h)$
  $x_{2h} := V(\mathscr{C}_{2h}, 0, r_{2h})$
  $x_h \leftarrow x_h + I_{2h}^h x_{2h}$
  Apply block Gauss–Seidel relaxation ν_2 times
return x_h

Algorithm 1. Multigrid V-cycle $V(\mathscr{C}_h, x_h, b_h)$ with ν_1 pre- and ν_2 postsmoothing steps on mesh $\mathscr{C}_h$ of the hierarchy.

further described in Appendix B. Returning to Algorithm 1, note that the V-cycle computes coarse grid corrections for both the velocity and pressure, simultaneously, and there is no need to strictly enforce the divergence constraint on any level of the hierarchy. Regarding the bottom solver, in this work a direct solver using a symmetric indefinite factorization of $\mathscr{A}_h$ on the coarsest mesh is used, together with an appropriate treatment of its associated trivial kernel.[4]

As is typical, applying more and more pre- and postsmoothing steps increases the convergence rate of the multigrid solver, but at greater computational cost. According to a variety of numerical experiments, a general observation made in this work is that a V-cycle with three pre- and postsmoothing steps is a good all-rounder, based on the metric of fastest computation time in reducing solution error by a given factor. On occasion, four presmoothing steps and two postsmoothing steps, or vice versa, performs marginally better, but on a problem-specific basis. Naturally, the optimal choice of multigrid design parameters is implementation- and problem-dependent, influenced by a wide variety of aspects, e.g., the relative computational costs of interpolation, restriction, and relaxation operators, or computing hardware characteristics, such as shared memory or distributed memory architectures and their associated memory communication costs. Further comments on V-cycle design, or counterparts such as W-cycles, are provided in the concluding remarks of Section 7.

To complete the description of the multigrid method, we note that although the V-cycle can be used as a standalone iterative solver, solver efficiency can be further accelerated by using it as a preconditioner of a Krylov method [5]. In this work,

[4]With periodic boundary conditions, the Stokes problem has a trivial kernel of dimension $d+1$, spanned by constant velocity and pressure fields; with velocity Dirichlet boundary conditions, the kernel is one-dimensional, spanned by constant pressure fields; with stress boundary conditions in viscous-stress form, the kernel is spanned by constant velocity fields as well as less trivial velocity modes such as, e.g., the velocity field $(x, y) \mapsto (-y, x)$ in 2D. The bottom solver robustly treats these modes through a simple least squares approach which (pre)computes the symmetric eigendecomposition of $\mathscr{A}_h$, "snapping" any nearly zero eigenvalues to exactly zero.

we have used a single V-cycle (with $\nu_1 = \nu_2 = 3$ pre- and postsmoothing steps) as a left-preconditioner of the GMRES method. Specifically, applying the V-cycle to an initial guess of zero on the fine mesh results in a linear operator, denoted in the remainder of this paper as V; the preconditioned system is then $V\mathscr{A}$. For simplicity, we do not consider restarted variants of GMRES here, in part because experiments show that convergence is generally attained in as few as five to fifteen steps for a ten-fold reduction in the order of magnitude of the residual. Moreover, convergence behavior is generally smooth during the iterations, such that the residual reduces in norm by a nearly constant factor each iteration of the GMRES method.

4. Influence of pressure penalty stabilization on multigrid efficiency

As outlined, the Stokes multigrid solver consists of a standard V-cycle, using a block Gauss–Seidel relaxation method in which each block corresponds to the collective set of degrees of freedom, of both velocity and pressure, on each mesh element. Key to rapid multigrid convergence is an apt choice of the user-defined pressure penalty stabilization prefactor parameter τ in (8). In general terms, if τ is below some positive threshold, the V-cycle fails to converge; above this threshold, there is a range of values for which convergence rates can match that of fast geometric multigrid methods for scalar Poisson problems; and beyond this range, multigrid efficiency will degrade.

To illustrate this behavior, as well as the suitability of the V-cycle as a preconditioner for the Stokes system $\mathscr{A}$, we examine the spectral properties of the preconditioned system $V\mathscr{A}$. In this particular example, we consider the standard form of the Stokes equations ($\gamma = 0$), with periodic boundary conditions on a 128×128 Cartesian grid using $p = 2$ biquadratic elements. As a function of τ, Figure 1 plots three quantities concerning the spectrum[5] of $V\mathscr{A}$: (i) the real part of the rightmost eigenvalue, $\max_i \operatorname{Re}\lambda_i$, (ii) the real part of the leftmost eigenvalue, $\min_i \operatorname{Re}\lambda_i$, and (iii) the greatest imaginary part, $\max_i \operatorname{Im}\lambda_i$. Ideally, the eigenvalues of $V\mathscr{A}$ should be clustered around 1, and we observe this is the case when $\tau \approx 0.1$; furthermore, near this value, the eigenvalues are nearly real. However, if τ is too small, then the leftmost eigenvalue of $V\mathscr{A}$ crosses the imaginary axis; in Figure 1 this occurs when $\tau \lessapprox 10^{-2}$. This represents a breakdown of the V-cycle, as then the (nontrivial) eigenvalues of $V\mathscr{A}$ cease to be bounded away from 0. (In fact, numerical

[5]The spectral analysis deliberately excludes the zero eigenvalues associated with the trivial kernel of the Stokes operator. In addition, regarding the results plotted in Figure 1, the spectral extremes have been estimated via the GMRES method, through computation of the eigenvalues of the upper-Hessenberg matrix of the corresponding Arnoldi iteration. Although only an approximation of the true spectrum of $V\mathscr{A}$, estimation via GMRES is significantly more efficient than forming and computing the spectrum of the dense matrix $V\mathscr{A}$; furthermore, the computed quantities are found in practice to be sufficiently accurate for the present purpose.

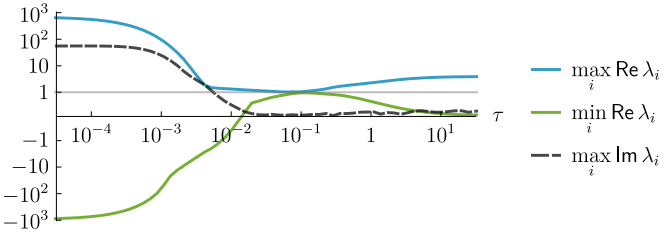

Figure 1. Spectral properties of the multigrid preconditioned system $V\mathscr{A}$ as a function of pressure penalty stabilization parameter; note the quasilogarithmic axis. Results correspond to a two-dimensional Stokes problem in standard form with $p = 2$; similar characteristics are obtained with other p, or in 3D, or with the viscous-stress form of the Stokes equations.

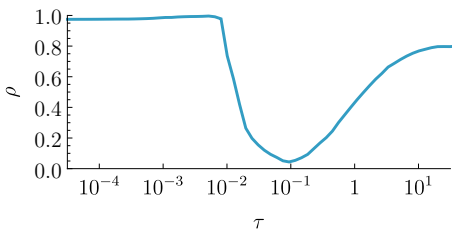

Figure 2. Multigrid efficiency as a function of pressure penalty stabilization parameter. Here, the convergence rate ρ, defined by (9), measures the average reduction factor per iteration in the residual of the V-cycle preconditioned GMRES method.

experiments examining the efficacy of the V-cycle as a standalone iterative method show that the V-cycle ceases to have spectral radius less than 1 at this same point.) Meanwhile, if τ is too large, e.g., $\tau \gtrapprox 1$, then the eigenvalues of $V\mathscr{A}$ begin to diverge away from 1.

In this work, the primary metric used to assess multigrid efficiency is the convergence rate ρ of the multigrid preconditioned GMRES method. This metric correlates with the spectral characteristics described above, and also to the performance of the V-cycle as a standalone iterative solver. Here, ρ is defined as the average residual reduction factor per iteration of the (left) preconditioned GMRES method,

$$\rho = \exp\left(\frac{1}{n}\log\frac{\|V\mathscr{A}x_n - Vb\|_2}{\|V\mathscr{A}x_0 - Vb\|_2}\right), \tag{9}$$

where n is the number of iterations required to reduce the residual by a factor of 10^8 from its starting value. In particular, a right-hand side of $b = 0$ is used, with initial guess x_0 given by a randomly generated $(d+1)$-dimensional vector field. With high probability, the randomly generated field contains modes which are damped slowest by the multigrid method, and thus ρ represents a typical "worst-case" convergence rate. Figure 2 shows the convergence rate ρ as a function of τ for the same example considered in Figure 1. Optimal convergence is attained when $\tau \approx 0.1$, precisely

Stokes form	d	polynomial degree p				
		1	2	3	4	5
standard	2D	0.19	0.10	0.086	0.019	0.031
	3D	0.12	0.088	0.084		
viscous stress	2D	0.14	0.046	0.034	0.0095	0.011
	3D	0.12	0.039	0.040		

Table 1. Optimal values of pressure penalty stabilization parameter, attaining minimal multigrid iteration count.

Stokes form	d	polynomial degree p				
		1	2	3	4	5
standard	2D	(0.15, 0.28)	(0.082, 0.12)	(0.067, 0.11)	(0.013, 0.029)	(0.021, 0.041)
	3D	(0.061, 0.30)	(0.070, 0.11)	(0.064, 0.12)		
viscous stress	2D	(0.091, 0.18)	(0.040, 0.056)	(0.027, 0.043)	(0.0058, 0.021)	(0.0072, 0.020)
	3D	(0.025, 0.19)	(0.031, 0.066)	(0.021, 0.059)		

Table 2. Range of pressure penalty stabilization parameters for which the number of multigrid iterations is at most 12.5% more than optimal.

when the spectrum of $V\mathcal{A}$ is tightly clustered around 1; meanwhile, when $\tau \lessapprox 10^{-2}$, ρ is approximately 1, representing the fact that GMRES is unable to effectively reduce the residual owing to a breakdown of the preconditioner.

Similar behavior to that seen in Figures 1 and 2 is observed for other choices of polynomial degree p, for different grid sizes, for the viscous-stress form of the Stokes equations, and in 3D as well as 2D. In all cases, the convergence rate ρ exhibits a well-defined valley as a function of τ. At present, a formula for the corresponding optimal value of τ is not known. In this work, a simple one-dimensional parameter sweep was used to find the optimal value of τ on successive grid sizes $n \times n(\times n)$ for $n = 4, 8, 16, \ldots$ up to $n = 256$ in 2D and $n = 128$ in 3D; experiments indicated that the optimum essentially converges around $n = 64$ or 128, beyond which $\arg\min_\tau \rho$ is relatively insensitive to the grid size. Table 1 contains the results of the search for a variety of p in 2D and 3D. In addition to the optimal value, a "window" of acceptable τ may also be computed: one can search for all τ such that $\rho(\tau) \leq (\min \rho)^{1-\epsilon}$, where $0 \leq \epsilon < 1$ represents a user-defined threshold for which the number of multigrid iterations increases by a factor of $1/(1-\epsilon)$ above the optimal minimum. For example, with $\epsilon = \frac{1}{9}$, the corresponding range of τ values yields at most 12.5% more iterations than optimal; Table 2 contains the corresponding ranges, and shows that, even if τ is not chosen exactly at the optimum, there is a relatively wide range of values for τ that will nevertheless attain good multigrid efficiency. Finally, numerical experiments indicate that the pressure

penalty parameter has very little influence on the velocity or pressure discretization error (see, for example, the demonstration given in Appendix C.1); this is ideal, as it allows us to concentrate mainly on the impact of τ on multigrid performance. Further comments concerning the selection of τ are given in the concluding remarks.

5. Multigrid efficiency for the time-independent Stokes equations

In the next two sections, multigrid performance is examined for a variety of Stokes problems, in both standard and viscous-stress forms, and with different types of boundary conditions. We also consider test cases with variable viscosity, multiphase problems exhibiting large discontinuities in μ across an embedded interface, and curved geometry problems which use semiunstructured meshes. Our primary focus is on demonstrating effective multigrid performance under the action of mesh refinement. The order of accuracy in the velocity and pressure, in the L^2 and maximum error norms, is also measured and reported. In two dimensions, grid sizes typically range from 4×4 up to 1024×1024, with polynomial degrees $p = 1, 2, \ldots, 5$; in three dimensions, owing to limited computing resources, only $p = 1, 2, 3$ are considered (i.e., trilinear, triquadratic, and tricubic polynomials) on grid sizes up to $128 \times 128 \times 128$ (the largest of these problems has half a billion degrees of freedom and requires 1 TB of memory to store just the block-diagonal component of the block-sparse matrix $\mathcal{A}$). In the remainder of this paper, for every test case, the pressure penalty stabilization parameter is chosen equal to the values reported in Table 1. Multigrid convergence rates are assessed using the average reduction factor in residual per iteration of the GMRES method, i.e., using (9), on a test problem with right-hand side $b = 0$, initial guess defined by a randomly generated $(d + 1)$-dimensional vector field, over as many iterations as necessary to reduce the initial residual by a factor of 10^8.

5.1. ***Periodic boundary conditions.*** We begin with perhaps the simplest Stokes problem, i.e., the Stokes equations in standard form, with $\mu = 1$ and periodic boundary conditions, on the unit square/cube domain $\Omega = (0, 1)^d$. Figure 3 plots

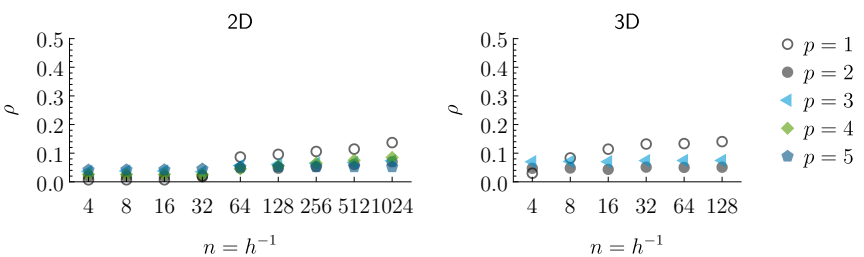

Figure 3. Measured multigrid convergence rates when solving the Stokes equations in standard form, with $\mu = 1$ and periodic boundary conditions.

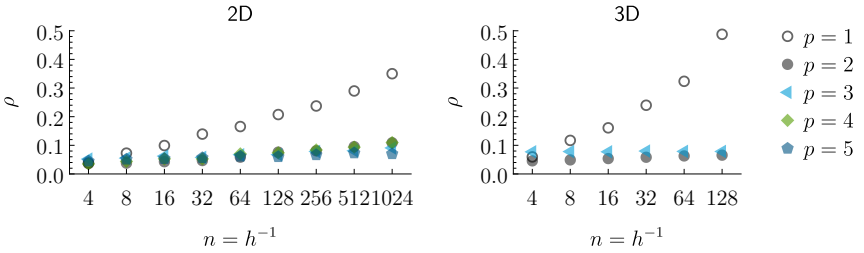

Figure 4. Measured multigrid convergence rates when solving the Stokes equations in standard form, with $\mu = 1$ and velocity Dirichlet boundary conditions.

the measured convergence rate ρ as a function of grid size $n \times n(\times n)$. With the exception of $p = 1$, ideal convergence rates are attained. For example, $\rho \approx 0.05$ corresponds to needing only seven iterations to reduce the residual by a factor of 10^9. When $p = 1$, however, slower convergence is seen — compared to the higher-degree cases, it appears that the elementwise block Gauss–Seidel relaxation method is less effective for a bilinear and trilinear LDG discretization of the Stokes equations. This behavior is consistently observed across all of the presented tests; see also the next two examples. Regarding the order of accuracy, numerical experiments show that both velocity and pressure attain order $p + 1$ accuracy, in both the L^2 and maximum error norms, for all p considered, in 2D and 3D; see Appendix C.

5.2. ***Dirichlet boundary conditions.*** The next test problem is identical to the previous, but with velocity Dirichlet boundary conditions imposed on $\partial\Omega$. Figure 4 plots the measured multigrid convergence rates and shows that, with the exception of $p = 1$, excellent multigrid performance is attained, similar to that of the periodic case in Figure 3. When $p = 1$, we observe a stronger failure of multigrid efficiency, with ρ diverging toward 1 as the mesh is refined. Another difference compared to the periodic case concerns the order of accuracy: numerical experiments show that the velocity attains order $p + 1$, in both the L^2 and maximum error norms, whereas the pressure field attains order $p + \frac{1}{2}$ in the L^2 norm and order p in the maximum norm; see Appendix C. The order reduction in the computed pressure field, as compared to the order $p + 1$ observed in the periodic case, is due to a numerical boundary layer (see, e.g., Figure 5); it is important to note, however, that this numerical boundary layer does not impact the optimal order accuracy of the computed velocity. See also the discussion of Cockburn et al. [14], wherein a priori estimates show that order p accuracy in pressure is to be expected for this LDG discretization of the Stokes equations.

5.3. ***Stress boundary conditions.*** We next test performance of the Stokes multigrid solver when stress boundary conditions are imposed, in which case the pertinent form of the Stokes equations is the viscous-stress form ($\gamma = 1$). Similar to the

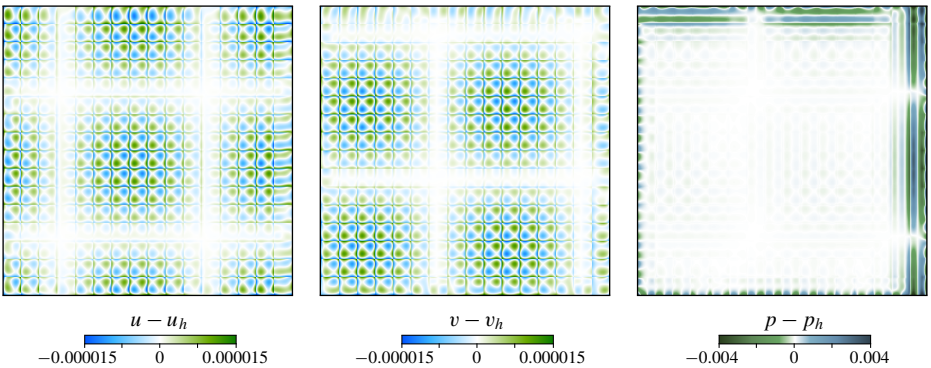

Figure 5. Illustration of the discrete error for the test case considered in Section 5.2, corresponding to a single-phase Stokes problem in standard form with velocity Dirichlet boundary conditions. The error in velocity $\boldsymbol{u} = (u, v)$ and pressure p is shown in the case of a 16×16 Cartesian mesh, for $p = 3$ bicubic polynomials. Note the numerical boundary layer in pressure, which according to grid convergence analyses does not impact the maximum norm optimal order accuracy of the velocity field.

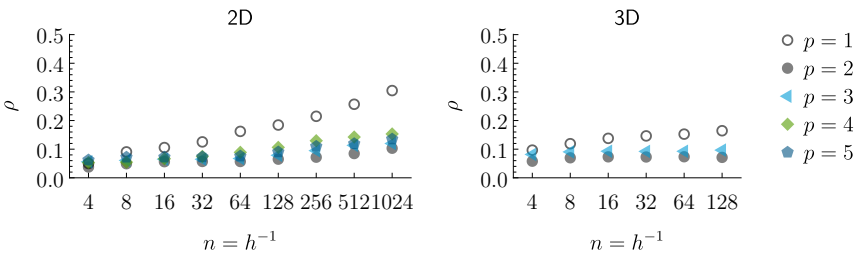

Figure 6. Measured multigrid convergence rates when solving the Stokes equations in viscous-stress form, with $\mu = 1$ and stress boundary conditions.

previous test problems, (2) with $\mu = 1$ is solved on an $n \times n(\times n)$ mesh of a unit square/cube domain $\Omega = (0, 1)^d$, with stress boundary conditions $\boldsymbol{\sigma} \cdot \boldsymbol{n} = \boldsymbol{h}_\partial$ imposed on $\partial\Omega$. Figure 6 plots the measured multigrid convergence rates. Compared to the previous two test problems (which employed the standard form of the Stokes equations), we observe a mild increase in ρ for the cases in which $p > 1$, most visible in 2D. The slight increase in ρ is mainly attributed to the viscous-stress form of the Stokes equations, and not solely to the imposition of stress boundary conditions. (Indeed, if one imposes Neumann-like boundary conditions for the Stokes equations in standard form, convergence rates similar to those in Figures 3 and 4 are obtained.) In Figure 6, for $p = 1$, we once again see less than ideal multigrid efficiency, though with marginal improvements compared to the case of velocity Dirichlet boundary conditions. Since the case of $p = 1$ is generally not of significant practical interest in the context of high-order accurate DG methods, we will focus on degrees $p > 1$ in the remainder of the presented results. Meanwhile,

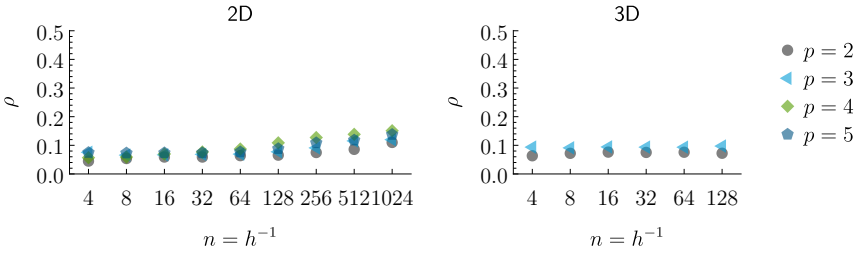

Figure 7. Measured multigrid convergence rates when solving the Stokes equations in viscous-stress form with a nonconstant viscosity function $\mu : \Omega \to \mathbb{R}^+$ given by (10), and stress boundary conditions.

for the current test problem, numerical experiments show that the velocity field attains order $p+1$ accuracy in the maximum norm, while the pressure field attains order $p+\frac{1}{2}$ in the L^2 norm and order p in the maximum norm; see Appendix C. As a general rule, our results indicate that whenever boundary conditions (or interfacial jump conditions) are imposed, the pressure field loses one order of accuracy near the boundary (or interface) owing to a numerical boundary layer, but this never affects the optimal order accuracy of the computed velocity field.

5.4. ***Variable viscosity.*** In the next example, we consider the possibility of a nonconstant viscosity function $\mu : \Omega \to \mathbb{R}^+$, varying throughout the domain. This problem serves three main purposes: (i) to test the application of the local inverse scaling by μ of the pressure penalty stabilization parameter suggested in (8), (ii) to demonstrate whether or not multigrid efficiency is impacted by variable ellipticity coefficient, and (iii) to examine the order of accuracy of the discrete solution in the variable-coefficient case. Specifically, we consider the Stokes equations in viscous-stress form on the unit square/cube domain $\Omega = (0,1)^d$, where $\mu : \Omega \to \mathbb{R}^+$ is given by

$$\mu = \begin{cases} 1 + \frac{1}{2}\sin 4\pi x \sin 4\pi y & \text{in 2D,} \\ 1 + \frac{1}{2}\sin 4\pi x \sin 4\pi y \sin 4\pi z & \text{in 3D,} \end{cases} \tag{10}$$

together with stress boundary conditions on $\partial\Omega$. Measured multigrid convergence rates, plotted in Figure 7, show very similar behavior to the constant-viscosity test problem of the previous example (Figure 6). Meanwhile, numerical experiments examining the order of accuracy (see Appendix C) confirm that the velocity field attains order $p+1$ in the maximum norm, while pressure attains order $p+\frac{1}{2}$ in the L^2 norm and order p in the maximum norm.

5.5. ***Curved domain geometry.*** In the last single-phase example of this section, we consider a Stokes problem in a curved domain. Here, and in all others involving curved geometry, we make use of implicitly defined meshes, building on the implicit mesh DG framework developed in prior work [38; 39]. Briefly, an implicitly defined

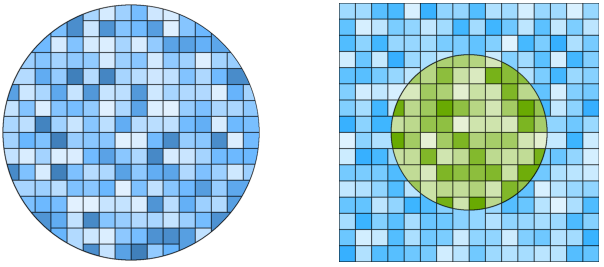

Figure 8. Examples of implicitly defined meshes generated with a background 16×16 Cartesian grid. Left: circular domain, used in the single-phase Stokes problem examined in Figure 9. Right: square domain with an embedded circular interface, used in the two-phase time-dependent Stokes problem considered later in this article, in Figure 12.

mesh uses one or more level set functions, describing the domain geometry and/or embedded interfaces, to cut through the cells of a background quadtree or octree; tiny cut cells are then merged with neighboring cells to create a mesh such that elements adjacent to curved geometry have their shape defined implicitly by the level set functions. In particular, the resulting mesh is interface- and boundary-conforming, thereby sharply representing the implicitly defined geometry. To use an implicitly defined mesh with a DG method, the main task is to compute quadrature schemes for the elements and faces whose geometry is implicitly defined; these are computed using the high-order accurate algorithms detailed in [37; 40], and then used when computing mass matrices, discrete gradient operators, L^2 projections, etc. For details on the implicit mesh DG framework, see [38; 39], and for illustrations demonstrating the associated (implicitly formed) mesh hierarchy underlying the multigrid method, see [20].

In the current test problem, we consider solving the Stokes equations, in standard form, on a unit diameter circle or sphere, $\Omega = \left\{x \in \mathbb{R}^d \;\middle|\; |x| < \frac{1}{2}\right\}$, with $\mu = 1$ and velocity Dirichlet boundary conditions. An example of the corresponding implicitly defined mesh generated by a background 16×16 Cartesian grid is shown in Figure 8, left. Measured multigrid convergence rates for this test case are shown in Figure 9 for polynomial degrees $p = 2$ and $p = 3$. Overall, efficient multigrid convergence rates are witnessed; however, once $n \geq 256$ we observe that ρ increases in value compared to previous test cases; we attribute this to the worsening conditioning of the discrete Stokes operator. In particular, inspection of numerical results shows that the conditioning of $\mathscr{A}$ exceeds 10^8 in these cases. Since the V-cycle is built by coarsening the components of the fine mesh operator $\mathscr{A}$, and since ρ is measured according to how many GMRES iterations it takes to reduce the (preconditioned) residual by a factor 10^8, it follows that ρ may be impacted by this conditioning. Regarding the grid convergence analysis (see Appendix C), numerical results once again confirm that

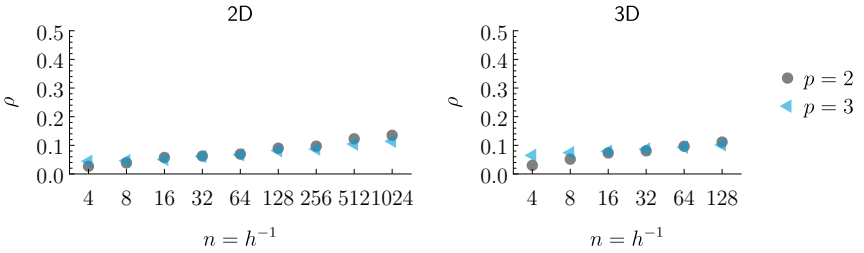

Figure 9. Measured multigrid convergence rates when solving the Stokes equations in standard form, with $\mu = 1$ in a unit diameter spherical domain using implicitly defined meshes together with velocity Dirichlet boundary conditions. Here, n denotes the number of cells in the background uniform Cartesian $n \times n (\times n)$ grid used to build the corresponding implicitly defined mesh; see Figure 8, left.

the velocity field attains order $p+1$ in the maximum norm, whereas the pressure attains order $p+\frac{1}{2}$ in the L^2 norm and order p in the maximum norm.

5.6. ***Multiphase Stokes equations with interfacial jump conditions.*** In this last example for the steady-state Stokes equations, we consider a challenging situation in which the viscosity coefficient μ exhibits jumps several orders in magnitude across an embedded interface. In particular, let $\Omega = (0,1)^d$ be divided into an interior square or cubic phase, $\Omega_1 = \left(\frac{1}{4}, \frac{3}{4}\right)^d$, together with an exterior phase, $\Omega_2 = \Omega \setminus \overline{\Omega_1}$. Four different ratios of viscosity jump across the interface are considered: in every case, the exterior phase has unit viscosity, $\mu_2 = 1$, whereas the interior phase will have one of four values, $\mu_1 \in \{10^{-6}, 10^{-3}, 10^{3}, 10^{6}\}$. We consider the multiphase viscous-stress form of the Stokes equations, (3)–(5) with $\gamma = 1$, for which interfacial jump conditions in velocity and the stress tensor are imposed on $\Gamma = \partial\Omega_1 \cup \partial\Omega_2$. To tackle this problem, we apply two distinct but complementary strategies: viscosity-upwinded weighted fluxes on interfacial mesh faces, and a diagonal scaling to improve the conditioning of the Stokes operator; these are discussed next.

Prior work on LDG methods for elliptic interface problems [41] shows that, in order to obtain ideal multigrid efficiency and solution accuracy, one should apply a biasing strategy for the LDG numerical fluxes on interfacial faces, wherein $\hat{\boldsymbol{u}}$ and $\hat{\boldsymbol{\sigma}}$ (see Appendix A) are biased toward one phase or the other, depending on the local viscosity coefficients. Here, we build on these ideas and extend it the multiphase Stokes case.

The strategy of biasing can be intuitively motivated as follows. Suppose that the interior phase has a vastly smaller viscosity than the exterior phase, i.e., $\mu_1 = \epsilon$ with $0 < \epsilon \ll 1$. Suppose also that the velocity $\boldsymbol{u}$ and its gradient is unit order in magnitude near the interface. From a physical standpoint, the pressure scales as $p \sim \mu U/L$ (where U is a typical velocity scale and L is a typical length scale), and so suppose also that pressure and its gradient is unit order in magnitude in

the exterior phase and scales with ϵ in the interior phase. Then, in the limit of vanishingly small ϵ, the stress jump condition in (4) approximately reduces to phase Ω_2 having a stress-like boundary condition, $\mu_2(\nabla \boldsymbol{u} + \nabla \boldsymbol{u}^T)\boldsymbol{n} - p\boldsymbol{n} \approx \boldsymbol{h}_{12}$ on Γ. Except for the modes associated with the trivial kernel of the corresponding Stokes operator, this is enough to determine the solution $(\boldsymbol{u}_2, p_2)$ in Ω_2 and thus the phase Ω_2 (nearly) decouples from phase Ω_1. Once $(\boldsymbol{u}_2, p_2)$ is found, the remaining jump condition in (4) essentially reduces to a Dirichlet boundary condition for the Stokes problem in phase Ω_1, i.e., $\boldsymbol{u}_1|_\Gamma \approx \boldsymbol{u}_2|_\Gamma - \boldsymbol{g}_{12}$, which is enough to determine $(\boldsymbol{u}_1, p_1)$ up to the modes associated with its kernel. Therefore, in the limit of vanishingly small ϵ, the highly viscous phase essentially "sees" a stress boundary condition on Γ (whose data is nearly independent of the other phase), and the nearly inviscid phase "sees" a Dirichlet boundary condition on Γ (whose data depends on the solution across the interface). It follows, therefore, that it may be advantageous to bias the numerical fluxes of the LDG formulation in the same way, to reflect the physical nature of the interfacial jump conditions in (4).

For elliptic interface problems, a common approach is to bias the fluxes according to a kind of harmonic weighting; in [41], a stronger kind of biasing is advocated, denoted viscosity-upwinded weighting. Here, we adopt the same strategy and refer the reader to [41] for discussion, much of which is directly analogous to the multiphase Stokes case, and to Appendix A with details on how the multivalued numerical flux functions $\hat{\boldsymbol{u}}$ and $\hat{\boldsymbol{\sigma}}$ are chosen for interfacial mesh faces.

In addition to the application of viscosity-upwinded numerical fluxes, one other numerical technique is employed to improve the conditioning of the multiphase Stokes equations. For a single-phase, constant-coefficient Stokes problem with viscosity μ, i.e., $\{-\mu\nabla^2\boldsymbol{u} + \nabla p = \boldsymbol{f},\ -\nabla\cdot\boldsymbol{u} = f\}$, the largest positive eigenvalue of the discretized Stokes operator scales as μ/h^2, whereas all negative eigenvalues scale inversely proportional to μ (and independently of h). Thus, unlike a Poisson problem, whose conditioning is independent of an arbitrary multiple of its ellipticity coefficient, the conditioning of the Stokes problem worsens quadratically in μ as μ is made arbitrarily large. In essence, the two operators of the momentum equations, $\mu\nabla^2\boldsymbol{u}$ and ∇p, are not on equal footing if μ is large and $\boldsymbol{u}$ and p are treated as independent variables. However, as noted earlier, from a physical point of view, the pressure scales as $p \sim \mu U/L$, and so the magnitude of p depends on μ. This apparent ill-conditioning of the Stokes operator can be easily remedied by rescaling one of the variables $\boldsymbol{u}$ or p by μ or $1/\mu$, respectively, effectively recasting the Stokes equations into a unit-viscosity form, $-\nabla^2\tilde{\boldsymbol{u}} + \nabla\tilde{p} = \tilde{\boldsymbol{f}}$.

This issue of ill-conditioning is exaggerated for the multiphase Stokes equations, in which the largest positive eigenvalue scales as $\max_i \mu_i/h^2$. However, as motivated by the single-phase case, a simple rescaling of the solution variables $\boldsymbol{u}$ and p can be used to mitigate the issue (see also [21; 2; 6]). In this work, we achieve

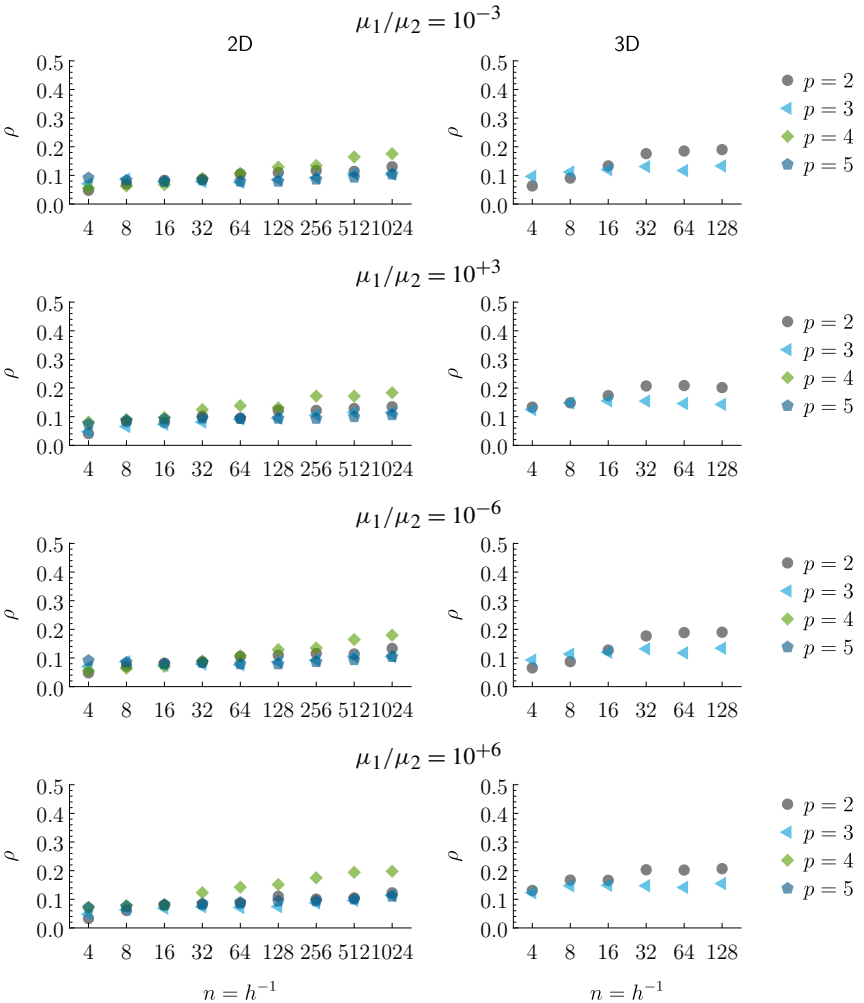

Figure 10. Measured multigrid convergence rates when solving the multiphase Stokes equations in viscous-stress form, wherein $\Omega_1 = \left(\frac{1}{4}, \frac{3}{4}\right)^d$ has viscosity μ_1 and $\Omega_2 = (0,1)^d \setminus \overline{\Omega_1}$ has viscosity μ_2, with the viscosity ratio as indicated and periodic boundary conditions. In each case, 2D results are plotted on the left and 3D results on the right.

this through a diagonal pre- and postscaling of the Stokes operator, i.e., replace the Stokes operator in (6) with

$$\begin{pmatrix} \alpha & 0 \\ 0 & \beta \end{pmatrix} \begin{pmatrix} A & M\mathcal{G} \\ \mathcal{G}^{\mathsf{T}} M & -E \end{pmatrix} \begin{pmatrix} \alpha & 0 \\ 0 & \beta \end{pmatrix}$$

where α and β are diagonal matrices whose entries equal $1/\sqrt{\mu}$ and $\sqrt{\mu}$, respectively. (More precisely, for every velocity field $\boldsymbol{u}$ and pressure field p, we have that $(\alpha \boldsymbol{u})|_E = \mu_E^{-1/2} \boldsymbol{u}|_E$ and $(\beta p)|_E = \mu_E^{1/2} p|_E$ for every element $E \in \mathscr{E}$, where μ_E is

the viscosity on element E.) This pre- and postscaling is built into the multigrid schemes by replacing the Stokes operator with the scaled version on all levels of the hierarchy. Note that it is straightforward to adapt the original linear system to the scaled approach: instead of solving $\mathcal{A}x = b$, one solves for $\tilde{\mathcal{A}}\tilde{x} = \tilde{b}$, where $\tilde{\mathcal{A}}$ is the scaled Stokes operator and $\tilde{b} = \mathrm{diag}(\alpha, \beta)b$, whereupon solving for $\tilde{x}$ gives the original unscaled solution as $x = \mathrm{diag}(\alpha, \beta)\tilde{x}$.

With the application of viscosity-upwinded numerical fluxes and the simple diagonal scaling to improve conditioning, Figure 10 plots the measured multigrid convergence rates when solving the multiphase Stokes equation in viscous-stress form with periodic boundary conditions, for the four different viscosity ratios considered, $\mu_1/\mu_2 \in \{10^{-6}, 10^{-3}, 10^{3}, 10^{6}\}$. Compared to previous test problems, we observe marginally slower convergence rates, representative of the challenging Stokes problems at hand. Nevertheless, good convergence rates are obtained across significant viscosity ratios, i.e., typically seven to ten iterations of multigrid preconditioned GMRES for a 10^8 reduction in the residual norm. Numerical experiments examining the order of accuracy (see Appendix C) show that the velocity attains order $p + 1$ in the maximum norm, while pressure attains order $p + \frac{1}{2}$ in the L^2 norm and order p in the maximum norm. As in other test problems, a numerical boundary layer exists in the pressure field: in the present case, the layer is adjacent to the interface but, as in other test problems, does not impact the optimal order accuracy of the velocity.

6. Multigrid efficiency for the time-dependent Stokes equations

So far, we have focused on the efficacy of the multigrid solver when applied to the steady-state Stokes equations. The time-dependent Stokes equations, however, may pose additional challenges, owing to the competing effects of the temporal derivative and viscous-stress operator [5]. To illustrate, consider the time-dependent Stokes equations in the form

$$\left.\begin{aligned}(\rho_i/\delta)\boldsymbol{u} - \nabla\cdot(\mu_i(\nabla\boldsymbol{u} + \gamma\nabla\boldsymbol{u}^{\mathsf{T}})) + \nabla p &= \boldsymbol{f}\\ -\nabla\cdot\boldsymbol{u} &= f\end{aligned}\right\} \quad \text{in } \Omega_i, \tag{11}$$

where ρ_i is a phase-dependent density and δ is a parameter proportional to the time step Δt of a temporal integration method. Its LDG discretization is a straightforward modification to the corresponding steady-state Stokes equations, and reads

$$\begin{pmatrix}(1/\delta)M_\rho + A & M\mathcal{G}\\ \mathcal{G}^{\mathsf{T}}M & -E\end{pmatrix}\begin{pmatrix}\boldsymbol{u}_h\\ p_h\end{pmatrix} = \begin{pmatrix}\boldsymbol{b_u}\\ b_p\end{pmatrix}, \tag{12}$$

where M_ρ is a ρ-weighted mass matrix, with all other operators unchanged.

When the viscous operator dominates (i.e., a small effective Reynolds number such that $\mu\delta/\rho$ is sufficiently large, depending on mesh resolution), the dominant

operator is the steady-state Stokes equations, with a small ρ/δ-weighted identity shift added to the viscous operator; in this case, one may expect a good steady-state Stokes solver to be effective. On the other hand, in the case when viscous effects are weak (i.e., a large effective Reynolds number such that $\mu\delta/\rho$ is small or the mesh is unable to resolve viscous effects), then (11) essentially reduces to a Helmholtz–Hodge projection problem (having strong connections to Chorin's projection method for solving the incompressible Navier–Stokes equations [10; 24; 31]). In the latter case, a solver designed specifically for the steady-state Stokes equations may deteriorate.

In this work, a simple strategy is employed to automatically account for these two competing effects, resulting in fast multigrid solvers across a full range of Reynolds numbers. In essence, the strategy chooses the pressure penalty stabilization parameter τ_p in (7) according to the expected scaling of the maximal eigenvalue of the operator $(\rho/\delta)\mathbb{I} - \nabla\cdot(\mu(\nabla + \gamma\nabla^{\mathsf{T}}))$. In the steady-state case, e.g., $\rho = 0$, the penalty parameter should scale as $\tau_p \sim h/\mu$. However, in the degenerate time-dependent Stokes case, in which $\mu = 0$, (12) reduces to an LDG method, written in flux form, for computing the solution of a Poisson problem with operator $\nabla\cdot((\delta/\rho)\nabla p)$; the appropriate scaling of the pressure penalty stabilization parameter in this case is then $\tau_p \sim \delta/(\rho h)$ [15; 41]. Both of these scaling statements may be summarized as follows: the penalty parameter should scale such that $\tau \sim (h\Lambda)^{-1}$, where Λ is the maximal eigenvalue of either the discretized operator $-\nabla\cdot(\mu(\nabla + \gamma\nabla^{\mathsf{T}}))$ (in the former case) or $(\rho/\delta)\mathbb{I}$ (in the latter case). This leads to a simple, but effective, idea to treat the general case: let $\tau \sim (h\Lambda)^{-1}$ where Λ is the sum of the expected scalings of the maximal eigenvalues of the two operators. Using this idea, the definition of the pressure penalty parameter in (7)–(8) is replaced with the relation

$$\tau_p = \left(\frac{h\rho}{\tau_0\delta} + \frac{\mu}{\tau h}\right)^{-1}. \tag{13}$$

Note that (13) reduces to the correct penalty parameter for the steady-state Stokes case when $\rho = 0$, i.e., $\tau_p = \tau h/\mu$, where τ is the multigrid-optimal parameter given in Table 1; in the other extreme, when $\mu = 0$ (or δ is vanishingly small), (13) reduces to $\tau = \tau_0\delta/(h\rho)$, appropriate for a scalar Poisson problem with ellipticity coefficient δ/ρ, where τ_0 is a user-defined constant prefactor. In the general situation, i.e., between these two extremes, it is important to note the scaling of (13) may change across the multigrid hierarchy — on a very fine mesh, h is small so that the second term may dominate; on a coarse mesh, however, h is large and the first term may dominate. This should be taken into account when building the multigrid method — in a purely geometric approach, wherein $\mathcal{A}_h$ is explicitly built on every level, (13) can be utilized directly; for the operator coarsening strategy, a simple modification to existing schemes is discussed in Appendix B.

The penalization choice in (13) blends across the two extremes of the time-dependent Stokes equations: a steady-state Stokes problem at one extreme, and a pure Poisson problem (written in flux form) at the other. Across a full range of Reynolds numbers, convergence results confirm optimal order of accuracy and that this choice of τ neither saturates the discretization error nor underpenalizes. As in the case of steady-state Stokes problems, we found that a V-cycle with three pre- and postsmoothing steps (used throughout this work) resulted in approximately optimal convergence speed, independent of the effective Reynolds number; occasionally a different combination of pre- and postsmoothing steps (at least two and at most four) can be slightly faster, but as mentioned earlier, the precise optimal value depends on a host of implementation, hardware, and problem-dependent factors.

6.1. ***Single-phase time-dependent Stokes equations.*** To demonstrate multigrid performance on time-dependent single-phase Stokes problems, we consider two effective Reynolds numbers $\mathrm{Re} = \rho U L/\mu$: $\mathrm{Re} \approx 100$, representing a viscous-dominated case (but where the time-derivative operator nevertheless influences performance characteristics) and $\mathrm{Re} \approx 10\,000$ (wherein the time-derivative operator definitively dominates). In both cases, the velocity and length scales are unitary, $U = 1$ and $L = 1$; density is set to $\rho = 1$, while viscosity satisfies $\mu = 10^{-2}$ in the former case and $\mu = 10^{-4}$ in the latter. In addition, we set $\delta = 0.1h$, where h is the element size on the finest-level mesh, representing a typical scenario of applying the time-dependent Stokes equations in a temporal integration method with CFL about 0.1. As before, multigrid efficiency is quantified through the average convergence rate ρ; Figure 11 plots the results (note the magnified vertical axis).[6] In Figure 11, top, we observe an upward trend in ρ as the mesh is refined; this corresponds to the fact that as the mesh is refined, eventually the viscous operator will definitively dominate and multigrid convergence rates similar to the steady-state Stokes equations will be attained. In the weakly viscous case with $\mathrm{Re} \approx 10\,000$, Figure 11, bottom, shows exceptionally fast multigrid convergence rates, with $\rho \approx 0.01$, corresponding to needing only four GMRES iterations to achieve a factor 10^8 reduction in the initial residual. In the context of solving the incompressible Navier–Stokes equations, these results suggest that a fast nonstationary Stokes solver may outperform a fast projection method solver; further remarks on this topic are given in the conclusions. Meanwhile, grid convergence experiments examining the order of accuracy (see Appendix C) show a departure from the typical results seen elsewhere in this work. In the weakly viscous case with $\mathrm{Re} \approx 10\,000$, the

[6] In the results of this section, we have reincluded the case of $p = 1$, mainly to serve as a point of interest: specifically, for steady-state Stokes problems, we noted in Section 5 that suboptimal multigrid efficiency may occur when $p = 1$; for unsteady Stokes problems, and depending on the effective Reynolds number, ideal multigrid convergence can be restored in this case, as seen in Figures 11 and 12.

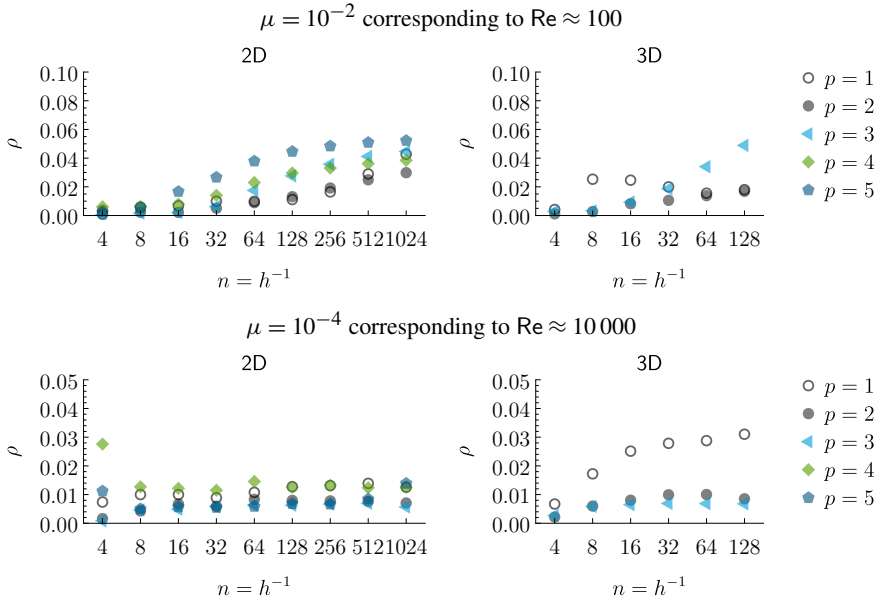

Figure 11. Measured multigrid convergence rates when solving the time-dependent, single-phase Stokes equations in standard form with μ as indicated and $\delta = 0.1h$, together with Neumann boundary conditions.

velocity is order $p+1$ in the maximum norm, and so is the pressure, despite the presence of boundary conditions. This apparent "superconvergence" in pressure is attributed to the property that, for very large Reynolds numbers, the time-dependent Stokes equations nearly reduce to a Helmholtz–Hodge projection, where one may naturally expect to attain optimal order accuracy in the pressure field; see, e.g., [38]. On the other hand, in the viscous-dominated case with Re ≈ 100, the pressure reduces to order $p + \frac{1}{2}$ in the L^2 norm and order p in the maximum norm, while the velocity maintains order $p+1$ in the maximum norm. This is perhaps expected, based on the order of accuracy results reported elsewhere in this work, together with the property that the viscous-dominated case essentially represents a mildly perturbed stationary Stokes problem.

6.2. ***Multiphase time-dependent Stokes equations.*** In our last two examples, we combine the challenging aspects of a multiphase Stokes problem together with nonstationary effects and consider a problem in which both the density ρ and viscosity μ have discontinuities several orders in magnitude across an embedded interface. The specific parameters considered correspond to two scenarios: a water bubble surrounded by gas, and a gas bubble surrounded by water. Specifically, $\rho_{\text{water}} = 1$, $\rho_{\text{gas}} = 0.001$, $\mu_{\text{water}} = 1$, and $\mu_{\text{gas}} = 0.0002$ (approximately accurate values for water and air at ambient temperature in CGS units). The radius of the

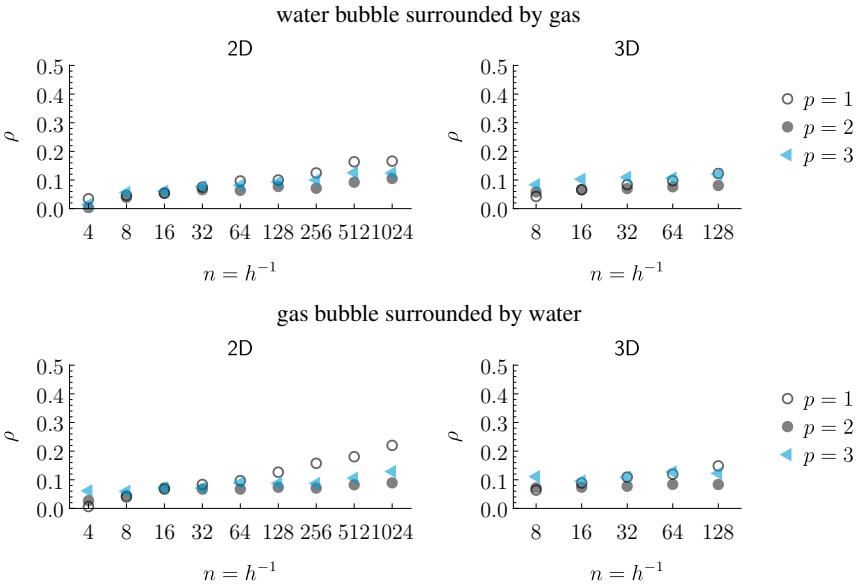

Figure 12. Measured multigrid convergence rates when solving the time-dependent multiphase Stokes equations in viscous-stress form, with $\rho_{\text{water}} = 1$, $\rho_{\text{gas}} = 0.001$, $\mu_{\text{water}} = 1$, $\mu_{\text{gas}} = 0.0002$, and $\delta = 0.1h$, together with velocity Dirichlet boundary conditions. Here, n denotes the number of cells in the background uniform Cartesian $n \times n(\times n)$ grid used to build the corresponding implicitly defined mesh; see Figure 8, right.

bubble is 0.3, centered in a unit square/cubic domain, $\Omega = (0, 1)^d$; (11) with $\gamma = 1$ and (4)–(5) are solved with velocity Dirichlet boundary conditions and stress jump conditions across the gas-water interface. As in the test case on curved domain geometry, a level set function describing the interface geometry is used together with an implicit mesh DG framework to create a semiunstructured, interface-conforming mesh; an example is shown in Figure 8, right. As before, the time step is set equal to $\delta = 0.1h$, where h is the typical element size on the finest mesh, representing the application of a time-stepping method with CFL number about 0.1.

This example combines three distinct but complementary strategies developed in prior examples: (i) viscosity-upwinded numerical fluxes, to robustly and accurately handle the large jump in viscosity across the interface, (ii) diagonal pre- and postscaling of the Stokes operator, to remove the unnecessary ill-conditioning caused by viscosity coefficients differing by several orders of magnitude, and (iii) a pressure penalty stabilization parameter controlled by (13), to automatically adjust penalization behavior between the two extremes of the time-dependent Stokes equations. As such, this example serves to demonstrate a variety of subtleties in efficiently solving the multiphase time-dependent Stokes equations, but as shown in the results plotted in Figure 12, combined together, one can attain highly efficient

multigrid solvers. For the particular scales chosen in this problem, the water-bubble and gas-bubble problems correspond to a unit-order Reynolds number. As such, grid convergence analyses confirm the expectation that the velocity attains order $p+1$ in the maximum norm, and pressure attains order $p+\frac{1}{2}$ in the L^2 norm and order p in the maximum norm.

7. Concluding remarks

In this paper, we devised efficient geometric multigrid solvers for the Stokes equations discretized by local discontinuous Galerkin (LDG) methods. The approach follows standard geometric multigrid concepts, utilizing a V-cycle and a simple block Gauss–Seidel relaxation method free from under-relaxation parameters. With a suitably chosen pressure penalty stabilization parameter, a wide array of tests showed that the Stokes multigrid solver can match the speed of classical geometric multigrid methods for Poisson problems [20; 41]. For example, typical convergence rates ranged from $\rho \approx 0.05$ to 0.1, corresponding to needing about six to eight iterations for a 10^8 reduction of the residual; fewer iterations are required for unsteady time-dependent Stokes equations. To implement the multigrid algorithm, one possibility is to explicitly build the mesh and associated LDG operators on every level of the hierarchy; an alternative method, not requiring the explicit formation of coarse meshes, can be implemented based on the operator coarsening algorithms detailed in Appendix B. In addition, we also extended the LDG methods of [14] to variable-viscosity and multiphase problems exhibiting interfacial stress jump conditions; across all test problems, grid convergence analyses demonstrated order $p+1$ accuracy in the maximum norm for the computed velocity field, and at least order p accuracy in the maximum norm for pressure.

Key findings of this work include the following aspects:

- Multigrid efficiency depends on an appropriate choice of the pressure penalty stabilization parameter τ_p. In general, for the steady-state Stokes equations, τ_p scales linearly with the local element size h and inversely proportional to the local viscosity, leaving the end user to define the prefactor τ in the formula $\tau_p = \tau h/\mu$. Table 1 provides values of τ resulting in optimal multigrid convergence rates, showing that τ depends on the polynomial degree of the DG space, the spatial dimension, and whether the standard form or viscous-stress form of the Stokes equations is being solved. Table 2 shows that an associated range of values exists for τ within which multigrid performance is very close to optimal. (See also [28] by Kanschat, in which the performance of an approximate Schur-form block preconditioner for LDG was also noted to depend crucially on the pressure penalty parameter, although the scaling of τ determined therein is very different from the results of the present work.)

- In the case of the time-dependent Stokes equations, (13) implements an effective strategy to automatically adjust the penalty parameter based on the competing effects of the viscous and time-derivative operators.

- For multiphase problems, in which the viscosity may exhibit jumps several orders in magnitude across an embedded interface, one should bias the numerical fluxes (here implemented with viscosity-upwinded weighting) to reflect the physical nature of the interfacial stress and velocity jump conditions. In addition, it can be advantageous to employ a diagonal preconditioning of the Stokes operator to correct the potential adverse effects large viscosities may have on the spectral characteristics of the saddle point problem.

A formula for the optimal pressure penalty parameter τ is not known at this time. In this work, a simple one-dimensional parameter sweep was used to find τ for particular choices of p, but a theoretical result would be ideal. Here, it may be possible to use analytical tools such as convergence criteria for block Gauss–Seidel methods [18], local mode analyses [43; 22], or local Fourier analyses [25; 19] to help determine a formula. On a similar note, it may also be possible to derive analytical proofs of the convergence of the overall multigrid method; see also, e.g., [42; 30; 17; 29; 27].

Other areas of study include the following. In this work, we mainly considered structured meshes (such as Cartesian grids) and semiunstructured, nonconforming implicitly defined meshes resulting from cell-merging procedures. Efficacy of the multigrid Stokes solver on fully unstructured meshes is also worthy of examination — for highly anisotropic meshes, one may need to group elements into clusters for the block Gauss–Seidel method to be effective [44; 35; 49]. Another possibility is to accelerate the Gauss–Seidel method through low-degree Chebyshev iterations, as was noted by Farrell et al. [19] for Vanka-type smoothers. Meanwhile, a wide variety of work on developing multigrid methods for the Stokes equations reports that W-cycles can be more effective than V-cycles, or even variable V-cycles which change the smoothing counts between levels of the hierarchy; see, e.g., [22; 17; 19; 29; 1]. Although a V-cycle with fixed presmoothing and postsmoothing steps was found highly effective in this work, these alternative strategies could prove useful for different applications. In the presented results, we also mentioned that the block Gauss–Seidel method is less effective when $p = 1$, i.e., for a bilinear or trilinear LDG discretization of the Stokes equations; although $p = 1$ is rarely of interest for DG methods, one possibility here is to use a $p > 1$ method as a preconditioner for the $p = 1$ system, or perhaps cluster elements into larger blocks for the Gauss–Seidel method. Meanwhile, the results of this paper could also be used to inform the design of algebraic multigrid methods [32].

Finally, we remark that the measured multigrid convergence rates for the time-dependent Stokes equations are particularly encouraging, indicating a strong potential for developing fast solvers for the general incompressible Navier–Stokes equations. In particular, preliminary work indicates that a fast Stokes solver could outperform the well-known and widely applied projection method of Chorin [10; 24] and could be integrated into arbitrary-order time-stepping methods for Navier–Stokes [31]; this will be further investigated in future work.

Appendix A: Local discontinuous Galerkin methods for the multiphase Stokes equations

In this section, an LDG framework is derived for the variable-viscosity multiphase Stokes equations. The construction partly follows the schemes set out by Cockburn et al. [14] but with some differences, including (i) the formulation is derived in a way which makes the role of the discrete gradient operator and its adjoint more visible (relevant to the operator coarsening multigrid schemes presented in Appendix B), (ii) the penalty stabilization operators are separated out from the numerical fluxes, and (iii) the formulation is extended to treat variable-viscosity and multiphase Stokes problems. For reference, the governing equations are repeated here: we seek to compute a velocity field $\boldsymbol{u} : \Omega \to \mathbb{R}^d$ and pressure field $p : \Omega \to \mathbb{R}$ such that

$$\left.\begin{aligned} -\nabla \cdot (\mu_i(\nabla \boldsymbol{u} + \gamma \nabla \boldsymbol{u}^{\mathsf{T}})) + \nabla p &= \boldsymbol{f} \\ -\nabla \cdot \boldsymbol{u} &= f \end{aligned}\right\} \quad \text{in } \Omega_i, \tag{14}$$

subject to the interfacial jump conditions

$$\left.\begin{aligned} [\![\boldsymbol{u}]\!] &= \boldsymbol{g}_{ij} \\ [\![\mu(\nabla \boldsymbol{u} + \gamma \nabla \boldsymbol{u}^{\mathsf{T}})\boldsymbol{n} - p\boldsymbol{n}]\!] &= \boldsymbol{h}_{ij} \end{aligned}\right\} \quad \text{on } \Gamma_{ij} \tag{15}$$

and boundary conditions

$$\begin{aligned} \boldsymbol{u} &= \boldsymbol{g}_\partial \quad \text{on } \Gamma_D, \\ \mu(\nabla \boldsymbol{u} + \gamma \nabla \boldsymbol{u}^{\mathsf{T}})\boldsymbol{n} - p\boldsymbol{n} &= \boldsymbol{h}_\partial \quad \text{on } \Gamma_N. \end{aligned} \tag{16}$$

We begin with some preliminary set up and notation. As stated in the main article, we consider meshes arising from Cartesian grids or, for domains or interfaces with curved geometry, semiunstructured quadtree/octree-based implicitly defined meshes. In this setting, it is natural to adopt a tensor product piecewise polynomial space. Let $\mathscr{E} = \bigcup_i E_i$ denote the set of mesh elements, let $p \geq 1$ be an integer, and define $\mathfrak{Q}_p(E)$ as the space of tensor product polynomials of (one-dimensional) degree p on element E. We assume in this work that the mesh is interface-conforming; i.e., if there is an interface separating the domain into two or more phases, then the interface does not cut through any element. Then, regarding the faces of the mesh, we denote *intraphase faces* as those shared by two elements in the same phase,

interphase faces as those shared by two elements in differing phases (therefore situated on Γ_{ij} for some i, j), and *boundary faces* as those situated on $\partial\Omega$. Each face has an associated normal vector $\boldsymbol{n}$; on intraphase faces, which are always flat and lie in a particular coordinate plan, $\boldsymbol{n}$ is defined to point "left to right"; e.g., for vertical faces in 2D, $\boldsymbol{n} = \hat{\boldsymbol{x}}$. Interphase faces adopt the same normal vector as the interface Γ_{ij} on which they coincide, defined to point from the phase with smallest phase index i into the phase with largest index, $j > i$. Boundary faces adopt the natural outwards-pointing normal to the domain boundary. The notation $[\![\cdot]\!]$ denotes the jump of a quantity across an interface or face and is defined consistent with its orientation; in particular, $[\![u]\!] := u^- - u^+$ where $u^{\pm}(x) = \lim_{\epsilon\to 0^+} u(x \pm \epsilon \boldsymbol{n})$ denotes the left and right trace values u^- and u^+, respectively. In addition, define Γ_0 as the set of all points belonging to intraphase faces, and for an element $E \in \mathscr{E}$, define $\chi(E)$ to be the phase of that element, such that $E \subseteq \Omega_{\chi(E)}$.

In the first step of the LDG formulation, a discrete approximation to $\nabla \boldsymbol{u}$ is defined through a "strong-weak" form.[7] Given $\boldsymbol{u} \in V_h$, $\boldsymbol{\eta} \in V_h^{d\times d}$ is defined such that

$$\int_E \boldsymbol{\eta} : \boldsymbol{\omega} = \int_E \nabla \boldsymbol{u} : \boldsymbol{\omega} + \int_{\partial E} (\hat{\boldsymbol{u}}_{\chi(E)} - \boldsymbol{u}) \cdot \boldsymbol{\omega} \cdot \boldsymbol{n} \tag{17}$$

holds for every element $E \in \mathscr{E}$ and every test function $\boldsymbol{\omega} \in V_h^{d\times d}$, where $\hat{\boldsymbol{u}}_\chi$ is a numerical flux function defined as

$$\hat{\boldsymbol{u}}_\chi := \begin{cases} \boldsymbol{u}^- & \text{on any intraphase face,} \\ \lambda \boldsymbol{u}^- + (1-\lambda)(\boldsymbol{u}^+ + \boldsymbol{g}_{\chi i}) & \text{on } \Gamma_{\chi i} \text{ if } \chi < i, \\ \lambda(\boldsymbol{u}^- - \boldsymbol{g}_{i\chi}) + (1-\lambda)\boldsymbol{u}^+ & \text{on } \Gamma_{i\chi} \text{ if } \chi > i, \\ \boldsymbol{u}^- & \text{on } \Gamma_N, \\ \boldsymbol{g}_\partial & \text{on } \Gamma_D. \end{cases} \tag{18}$$

(See Figure 13 for a schematic illustration.) Note that the flux is multivalued on interphase faces — on these faces, the interfacial jump condition $[\![\boldsymbol{u}]\!] = \boldsymbol{g}_{ij}$ on Γ_{ij} in (15) is taken into account as follows: when an element "reaches across" the interface to evaluate the trace of $\boldsymbol{u}$ on the other side, the trace value is compensated by the jump data to correctly account for the intended discontinuity in the solution. Note

[7] The *strong-weak form* states that $\boldsymbol{\eta}$ must satisfy $\int_E \boldsymbol{\eta} : \boldsymbol{\omega} = \int_E \nabla \boldsymbol{u} : \boldsymbol{\omega} + \int_{\partial E} (\hat{\boldsymbol{u}} - \boldsymbol{u}) \cdot \boldsymbol{\omega} \cdot \boldsymbol{n}$ for all test functions $\boldsymbol{\omega}$, whereas the *weak form* states that $\boldsymbol{\eta}$ must satisfy $\int_E \boldsymbol{\eta} : \boldsymbol{\omega} = -\int_E \boldsymbol{u} \cdot (\nabla \cdot \boldsymbol{\omega}) + \int_{\partial E} \hat{\boldsymbol{u}} \cdot \boldsymbol{\omega} \cdot \boldsymbol{n}$. The two forms are equivalent whenever the associated quadrature scheme exactly preserves the identity of integration by parts. This is often the case for many implementations, including on quadrilateral, prismatic, or simplicial elements. However, this property may not hold when approximate quadrature schemes are used, e.g., for implicitly defined meshes which use high-order accurate quadrature schemes (wherein integration by parts only holds up to a high-order truncation error). In the latter situation, to ensure symmetry of the final discretized Stokes operator, it is necessary to use the strong-weak form to define $\nabla \boldsymbol{u}$ and the weak form to defined $\nabla \cdot \boldsymbol{\sigma}$ (see (22)), or vice versa. For further discussion as it relates to the analogous case of elliptic interface problems, see [38].

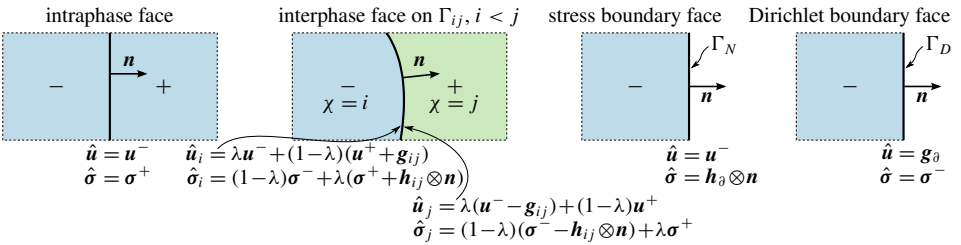

Figure 13. Schematic of the numerical flux functions $\hat{\boldsymbol{u}}$ and $\hat{\boldsymbol{\sigma}}$ defined by (18) and (23). Except for interphase faces, the flux is single-valued; on interphase faces, the flux is multivalued so as to incorporate the interfacial jump conditions $[\![\boldsymbol{u}]\!]=\boldsymbol{g}_{ij}$ and $[\![\boldsymbol{\sigma}\cdot\boldsymbol{n}]\!]=\boldsymbol{h}_{ij}$ on Γ_{ij}, $i<j$. Plus and minus signs denote the elemental values on the right and left of the face, respectively; e.g., for a point x on the face, $\boldsymbol{u}^\pm(x)=\lim_{\epsilon\to 0^+}\boldsymbol{u}(x\pm\epsilon\boldsymbol{n})$.

also that interfacial fluxes are weighted through a convex combination parameter $\lambda\in[0,1]$, which can vary from face to face. (For example, if $\lambda=0$, then the numerical flux is sourced solely from the right element's trace $\boldsymbol{u}^+$; if $\lambda=1$, it is sourced solely from the left element's trace $\boldsymbol{u}^-$.) The purpose of λ is to implement the strategy of viscosity-upwinded numerical fluxes, as detailed in Appendix A.2. Upon summing (17) over every element of the mesh, one has

$$\begin{aligned}\int_\Omega \boldsymbol{\eta}:\boldsymbol{\omega} &= \sum_{E\in\mathscr{E}}\int_E \nabla\boldsymbol{u}:\boldsymbol{\omega}+\int_{\Gamma_0}[(\hat{\boldsymbol{u}}-\boldsymbol{u}^-)\cdot\boldsymbol{\omega}^-\cdot\boldsymbol{n}-(\hat{\boldsymbol{u}}-\boldsymbol{u}^+)\cdot\boldsymbol{\omega}^+\cdot\boldsymbol{n}]\\
&\quad+\sum_{j>i}\int_{\Gamma_{ij}}[(\hat{\boldsymbol{u}}_i-\boldsymbol{u}^-)\cdot\boldsymbol{\omega}^-\cdot\boldsymbol{n}-(\hat{\boldsymbol{u}}_j-\boldsymbol{u}^+)\cdot\boldsymbol{\omega}^+\cdot\boldsymbol{n}]\\
&\quad+\int_{\Gamma_D}(\hat{\boldsymbol{u}}-\boldsymbol{u}^-)\cdot\boldsymbol{\omega}^-\cdot\boldsymbol{n}+\int_{\Gamma_N}(\hat{\boldsymbol{u}}-\boldsymbol{u}^-)\cdot\boldsymbol{\omega}^-\cdot\boldsymbol{n}\\
&= \sum_{E\in\mathscr{E}}\int_E \nabla\boldsymbol{u}:\boldsymbol{\omega}-\int_{\Gamma_0}(\boldsymbol{u}^--\boldsymbol{u}^+)\cdot\boldsymbol{\omega}^+\cdot\boldsymbol{n}\\
&\quad-\sum_{j>i}\int_{\Gamma_{ij}}[(1-\lambda)(\boldsymbol{u}^--\boldsymbol{u}^+)\cdot\boldsymbol{\omega}^-\cdot\boldsymbol{n}+\lambda(\boldsymbol{u}^--\boldsymbol{u}^+)\cdot\boldsymbol{\omega}^+\cdot\boldsymbol{n}]\\
&\quad+\sum_{j>i}\int_{\Gamma_{ij}}(1-\lambda)\boldsymbol{g}_{ij}\cdot\boldsymbol{\omega}^-\cdot\boldsymbol{n}+\lambda\boldsymbol{g}_{ij}\cdot\boldsymbol{\omega}^+\cdot\boldsymbol{n}\\
&\quad+\int_{\Gamma_D}(\boldsymbol{g}_\partial-\boldsymbol{u}^-)\cdot\boldsymbol{\omega}^-\cdot\boldsymbol{n},\end{aligned}\tag{19}$$

which motivates the definition of the following operators:

- Let $\nabla_h: V_h\to V_h^d$ be the broken gradient operator, and let $L: V_h\to V_h^d$ be the lifting operator, such that

$$\int_\Omega(\nabla_h u)\cdot\boldsymbol{v}=\sum_{E\in\mathscr{E}}\int_E\nabla u\cdot\boldsymbol{v}$$

and

$$\int_\Omega (Lu)\cdot \boldsymbol{v} = \sum_{j>i} \int_{\Gamma_{ij}} (1-\lambda)(u^+ - u^-)\boldsymbol{v}^- \cdot \boldsymbol{n} + \lambda(u^+ - u^-)\boldsymbol{v}^+ \cdot \boldsymbol{n} + \int_{\Gamma_0} (u^+ - u^-)\boldsymbol{v}^+ \cdot \boldsymbol{n} - \int_{\Gamma_D} u^- \boldsymbol{v}^- \cdot \boldsymbol{n}$$

hold for every $\boldsymbol{v} \in V_h^d$.

- Define $J_g \in V_h^{d\times d}$ such that

$$\int_\Omega J_g : \boldsymbol{\omega} = \int_{\Gamma_D} \boldsymbol{g}_\partial \cdot \boldsymbol{\omega}^- \cdot \boldsymbol{n} + \sum_{j>i} \int_{\Gamma_{ij}} (1-\lambda)\boldsymbol{g}_{ij} \cdot \boldsymbol{\omega}^- \cdot \boldsymbol{n} + \lambda \boldsymbol{g}_{ij} \cdot \boldsymbol{\omega}^+ \cdot \boldsymbol{n}$$

holds for every $\boldsymbol{\omega} \in V_h^{d\times d}$.

With these definitions, (19) is equivalent to the statement that

$$\boldsymbol{\eta}_{ij} = G_j u_i + J_{g,ij},$$

where $G : V_h \to V_h^d$ is the *discrete gradient operator*, $G := \nabla_h + L$, having components $G = (G_1, \ldots, G_d)$, $\boldsymbol{\eta}_{ij}$ denotes the (i, j)-th component of $\boldsymbol{\eta}$, and u_i denotes the i-th component of $\boldsymbol{u}$. To complete this step of the LDG construction, we define $\boldsymbol{\tau}$ as the natural discretization of $\nabla \boldsymbol{u} + \gamma \nabla \boldsymbol{u}^\mathsf{T}$:

$$\boldsymbol{\tau} := \boldsymbol{\eta} + \gamma \boldsymbol{\eta}^\mathsf{T}, \quad \text{i.e.,} \quad \tau_{ij} := G_j u_i + \gamma G_i u_j + J_{g,ij} + \gamma J_{g,ji}.$$

In the second step of the LDG formulation, a discrete approximation of $\boldsymbol{\sigma} = \mu\boldsymbol{\tau} - p\mathbb{I}$ is defined. In essence, this is implemented via an L^2 projection of $\mu\boldsymbol{\tau}$ onto $V_h^{d\times d}$. We define $\boldsymbol{\sigma} \in V_h^{d\times d}$ as the unique piecewise polynomial function such that

$$\int_E \boldsymbol{\sigma} : \boldsymbol{\omega} = \int_E (\mu\boldsymbol{\tau} - p\mathbb{I}) : \boldsymbol{\omega} \tag{20}$$

holds for every element $E \in \mathscr{E}$ and every test function $\boldsymbol{\omega} \in V_h^{d\times d}$. In the case that μ is piecewise constant, computing this L^2 projection is a simple matter of multiplying $\boldsymbol{\tau}$ by a scalar and subtracting the discrete pressure $p \in V_h$. In the general case, it is straightforward to show that a μ-weighted L^2 projection is equivalent to multiplication by the block-diagonal matrix $M^{-1}M_\mu$, where M_μ is the μ-weighted mass matrix such that $u^\mathsf{T} M_\mu v = \int_\Omega u\mu v$ holds for all $u, v \in V_h$. In this work, M_μ is computed with sufficiently high-order accurate quadrature schemes, typically Gaussian quadrature schemes. Using this relation, (20) is equivalent to

$$\sigma_{ij} := M^{-1}M_\mu \tau_{ij} - p\delta_{ij} = M^{-1}M_\mu(G_j u_i + \gamma G_i u_j + J_{g,ij} + \gamma J_{g,ji}) - p\delta_{ij}. \tag{21}$$

In the third step, we consider a weak formulation for computing the divergence of $\boldsymbol{\sigma}$. This proceeds similarly to defining the discrete gradient of $\boldsymbol{u}$, except numerical

fluxes act in the opposite direction. For simplicity of presentation, the following numerical flux for $\boldsymbol{\sigma}$ is matrix-valued; however, only the normal component of the flux is used. Given $\boldsymbol{\sigma} \in V_h^{d\times d}$, define $\boldsymbol{w} \in V_h^d$ as the discrete divergence of $\boldsymbol{\sigma}$ such that

$$\int_E \boldsymbol{w}\cdot\boldsymbol{v} = -\int_E \boldsymbol{\sigma} : \nabla \boldsymbol{v} + \int_{\partial E} \boldsymbol{v}\cdot\hat{\boldsymbol{\sigma}}_{\chi(E)}\cdot\boldsymbol{n} \tag{22}$$

holds for every test function $\boldsymbol{v} \in V_h^d$ and every element $E \in \mathscr{E}$. Here, the numerical flux is defined by (see also Figure 13)

$$\hat{\boldsymbol{\sigma}}_\chi := \begin{cases} \boldsymbol{\sigma}^+ & \text{on any intraphase face,} \\ (1-\lambda)\boldsymbol{\sigma}^- + \lambda(\boldsymbol{\sigma}^+ + \boldsymbol{h}_{\chi i}\otimes\boldsymbol{n}) & \text{on } \Gamma_{\chi i} \text{ if } \chi < i, \\ (1-\lambda)(\boldsymbol{\sigma}^- - \boldsymbol{h}_{i\chi}\otimes\boldsymbol{n}) + \lambda\boldsymbol{\sigma}^+ & \text{on } \Gamma_{i\chi} \text{ if } \chi > i, \\ \boldsymbol{h}_\partial \otimes \boldsymbol{n} & \text{on } \Gamma_N, \\ \boldsymbol{\sigma}^- & \text{on } \Gamma_D. \end{cases} \tag{23}$$

As in the numerical flux for $\hat{\boldsymbol{u}}$, the interfacial jump condition $[\![\boldsymbol{\sigma}\cdot\boldsymbol{n}]\!] = \boldsymbol{h}_{ij}$ on Γ_{ij} in (15) is taken into account via a multivalued interfacial flux, such that whenever an element reaches across the interface, the neighboring element's trace is compensated by $\boldsymbol{h}_{ij}$ to correctly put it in the context of the source element. Summing (22) over every element, one has, for every $\boldsymbol{v} \in V_h^d$,

$$\begin{aligned} \int_\Omega \boldsymbol{w}\cdot\boldsymbol{v} &= -\sum_{E\in\mathscr{E}}\int_E \boldsymbol{\sigma}:\nabla\boldsymbol{v} + \int_{\Gamma_0}(\boldsymbol{v}^- - \boldsymbol{v}^+)\cdot\hat{\boldsymbol{\sigma}}\cdot\boldsymbol{n} + \sum_{j>i}\int_{\Gamma_{ij}}(\boldsymbol{v}^-\cdot\hat{\boldsymbol{\sigma}}_i - \boldsymbol{v}^+\cdot\hat{\boldsymbol{\sigma}}_j)\cdot\boldsymbol{n} \\ &\quad + \int_{\Gamma_D}\boldsymbol{v}^-\cdot\hat{\boldsymbol{\sigma}}\cdot\boldsymbol{n} + \int_{\Gamma_N}\boldsymbol{v}^-\cdot\hat{\boldsymbol{\sigma}}\cdot\boldsymbol{n} \\ &= -\sum_{E\in\mathscr{E}}\int_E \boldsymbol{\sigma}:\nabla\boldsymbol{v} + \int_{\Gamma_0}(\boldsymbol{v}^- - \boldsymbol{v}^+)\cdot\boldsymbol{\sigma}^+\cdot\boldsymbol{n} \\ &\quad + \sum_{j>i}\int_{\Gamma_{ij}}(1-\lambda)(\boldsymbol{v}^- - \boldsymbol{v}^+)\cdot\boldsymbol{\sigma}^-\cdot\boldsymbol{n} + \lambda(\boldsymbol{v}^- - \boldsymbol{v}^+)\cdot\boldsymbol{\sigma}^+\cdot\boldsymbol{n} \\ &\quad + \sum_{j>i}\int_{\Gamma_{ij}}\lambda\boldsymbol{v}^-\cdot\boldsymbol{h}_{ij} + (1-\lambda)\boldsymbol{v}^+\cdot\boldsymbol{h}_{ij} \\ &\quad + \int_{\Gamma_D}\boldsymbol{v}^-\cdot\boldsymbol{\sigma}^-\cdot\boldsymbol{n} + \int_{\Gamma_N}\boldsymbol{v}^-\cdot\boldsymbol{h}_\partial. \end{aligned} \tag{24}$$

Similar to the operator J_g defined above, let $J_h \in V_h^d$ be such that

$$\int_\Omega J_h\cdot\boldsymbol{v} = \int_{\Gamma_N}\boldsymbol{v}^-\cdot\boldsymbol{h}_\partial + \sum_{j>i}\int_{\Gamma_{ij}}\lambda\boldsymbol{v}^-\cdot\boldsymbol{h}_{ij} + (1-\lambda)\boldsymbol{v}^+\cdot\boldsymbol{h}_{ij}$$

holds for every $\boldsymbol{v} \in V_h^d$. Then, upon using the lifting operator defined earlier, (24) is equivalent to the statement that, for every $\boldsymbol{v} \in V_h^d$,

$$(\boldsymbol{w}, \boldsymbol{v}) = -\sum_{i=1}^{d} (\boldsymbol{\sigma}_i, \nabla_h v_i) - (\boldsymbol{\sigma}_i, L v_i) + (J_h, \boldsymbol{v}) = -\sum_{i=1}^{d} (\boldsymbol{\sigma}_i, G v_i) + (J_h, \boldsymbol{v}),$$

where v_i denotes the i-th component of $\boldsymbol{v}$ and $\boldsymbol{\sigma}_i$ denotes the i-th row of the matrix $\boldsymbol{\sigma}$. Transferring the G operator onto $\boldsymbol{\sigma}_i$ via the adjoint, it follows that $w_i = -\sum_{j=1}^{d} M^{-1} G_j^{\mathsf{T}} M \sigma_{ij} + J_{h,i}$, where w_i is the i-th component of $\boldsymbol{w}$. Combining with (21), we have

$$w_i = -\sum_{j=1}^{d} M^{-1} G_j^{\mathsf{T}} M_\mu (G_j u_i + \gamma G_i u_j) + M^{-1} G_i^{\mathsf{T}} M p - \sum_{j=1}^{d} M^{-1} G_j^{\mathsf{T}} M_\mu (J_{g,ij} + \gamma J_{g,ji}) + J_{h,i}. \quad (25)$$

This is the weak statement that $\boldsymbol{w}$ is equal to the discrete divergence of $\boldsymbol{\sigma}$, itself a discrete approximation to $\mu(\nabla \boldsymbol{u} + \gamma \nabla \boldsymbol{u}^{\mathsf{T}}) - p\mathbb{I}$, taking into account velocity Dirichlet boundary data, stress boundary data, and interfacial jump condition data (if any). One may recognize the first term of (25) as implementing $\nabla \cdot (\mu(\nabla \boldsymbol{u} + \gamma \nabla \boldsymbol{u}^{\mathsf{T}}))$ and the second term as implementing $-\nabla p$, while the remaining terms represent the contribution of any boundary or interfacial jump data.

We now turn to the LDG discretization of the divergence constraint of the Stokes equations. Given $\boldsymbol{u} \in V_h^d$, define $w \in V_h$ as the discrete divergence of $\boldsymbol{u}$ via the strong-weak form, such that

$$\int_E w v = \int_E v \nabla \cdot \boldsymbol{u} + \int_{\partial E} v (\hat{\boldsymbol{u}}_{\chi(E)} - \boldsymbol{u}) \cdot \boldsymbol{n} \quad (26)$$

holds for every element $E \in \mathscr{E}$ and every $v \in V_h$. Here, the same numerical flux for $\boldsymbol{u}$ as was used to define its gradient (see (18)) is employed, but only the normal component is seen by (26). A similar derivation as before reveals that w is essentially equal to the trace of $\boldsymbol{\eta}$, i.e.,

$$w = \sum_{i=1}^{d} G_i u_i + J_{g \cdot n},$$

with $J_{g \cdot n} \in V_h$ such that $\int_\Omega J_{g \cdot n} v = \int_{\Gamma_D} v^- \boldsymbol{g}_\partial \cdot \boldsymbol{n} + \sum_{j>i} \int_{\Gamma_{ij}} ((1-\lambda) v^- + \lambda v^+) \boldsymbol{g}_{ij} \cdot \boldsymbol{n}$ holds for every $v \in V_h$.

Lastly, in what is essentially the final step of the LDG formulation for the Stokes equations (14)–(16), penalty stabilization terms are added to ensure the well-posedness of the discrete problem [14; 3; 26]. These terms weakly impose solution continuity between neighboring element polynomials in the same phase, weakly

impose Dirichlet boundary conditions, and weakly enforce interfacial jump conditions (if any). Regarding the stabilization parameters for velocity, we classify them according to three types: boundary ($\tau_{\boldsymbol{u},\partial}$), intraphase ($\tau_{\boldsymbol{u},0}$), and interphase ($\tau_{\boldsymbol{u},ij}$). Let $\mathscr{E}_{\boldsymbol{u},g} : V_h^d \to V_h^d$ be the affine operator such that, for each $\boldsymbol{u} \in V_h$,

$$\int_\Omega \mathscr{E}_{\boldsymbol{u},g}(\boldsymbol{u}) \cdot \boldsymbol{v} = \int_{\Gamma_0} \tau_{\boldsymbol{u},0} [\![\boldsymbol{u}]\!] \cdot [\![\boldsymbol{v}]\!] + \sum_{j>i} \int_{\Gamma_{ij}} \tau_{\boldsymbol{u},ij} ([\![\boldsymbol{u}]\!] - \boldsymbol{g}_{ij}) \cdot [\![\boldsymbol{v}]\!] + \int_{\Gamma_D} \tau_{\boldsymbol{u},\partial} (\boldsymbol{u}^- - \boldsymbol{g}_\partial) \cdot \boldsymbol{v}^- \quad (27)$$

holds for every test function $\boldsymbol{v} \in V_h^d$. Note that $\mathscr{E}_{\boldsymbol{u},g}(\boldsymbol{u}) = (\mathscr{E}_{\boldsymbol{u},0} u_1, \ldots, \mathscr{E}_{\boldsymbol{u},0} u_d) + \mathscr{E}_{\boldsymbol{u},g}(0)$, where $\mathscr{E}_{\boldsymbol{u},0}$ represents the linear part of the operator acting on the vector field components, defined analogously to (27) with homogeneous jump and boundary data. Concerning the stabilization parameter τ_p for pressure, define the linear operator $\mathscr{E}_p : V_h \to V_h$ such that, for each $p \in V_h$,

$$\int_\Omega \mathscr{E}_p(p) v = \int_{\Gamma_0} \tau_p [\![p]\!] [\![v]\!]$$

holds for every test function $v \in V_h$. Following the formulation presented in [14], subject to a suitable specification of the parameter values to be discussed shortly, these operators are added and subtracted to the discretization of the Stokes momentum equations and divergence constraint, respectively. Specifically, the discretized multiphase Stokes problem (14)–(16) consists of finding a velocity field $\boldsymbol{u} \in V_h^d$ and pressure field $p \in V_h$ such that

$$\sum_{j=1}^d M^{-1} G_j^\mathsf{T} M_\mu (G_j u_i + \gamma G_i u_j) + \mathscr{E}_{\boldsymbol{u},0} u_i - M^{-1} G_i^\mathsf{T} M p = \mathbb{P}_{V_h^d}(f_i) - \sum_{j=1}^d M^{-1} G_j^\mathsf{T} (J_{g,ij} + \gamma J_{g,ji}) + J_{h,i} - \mathscr{E}_{\boldsymbol{u},g,i}(0) \quad (28)$$

holds for each $i = 1, \ldots, d$, where f_i denotes the i-th component of $\boldsymbol{f}$, subject to the divergence constraint

$$-\sum_{i=1}^d G_i u_i - \mathscr{E}_p p = \mathbb{P}_{V_h}(f) + J_{g\cdot n}. \quad (29)$$

Multiplying both (28) and (29) by M, taken together these equations may be succinctly written in block form as

$$\begin{pmatrix} A & M\mathcal{G} \\ \mathcal{G}^\mathsf{T} M & -E \end{pmatrix} \begin{pmatrix} \boldsymbol{u} \\ p \end{pmatrix} = \begin{pmatrix} \boldsymbol{b}_{\boldsymbol{u}} \\ b_p \end{pmatrix} \quad (30)$$

where the block operators are as follows:

- A implements the viscous part of the Stokes momentum equations, and can be written in $d \times d$ block form corresponding to its action on the d components of $\boldsymbol{u}$, where the (i, j)-th block is given by

$$A_{ij} = \delta_{ij}\left(\sum_{k=1}^{d} G_k^{\mathsf{T}} M_\mu G_k\right) + \gamma G_j^{\mathsf{T}} M_\mu G_i + \delta_{ij}\widetilde{E}, \tag{31}$$

  where $\widetilde{E} := M\mathscr{E}_{\boldsymbol{u},0}$ is the penalty stabilization matrix associated with velocity stabilization.

- $\mathscr{G} = (\mathscr{G}_1, \ldots, \mathscr{G}_d)$ is a discrete gradient operator, closely related to the adjoint of G, whose components are given by

$$\mathscr{G}_i = -M^{-1} G_i^{\mathsf{T}} M.$$

- E is the penalty stabilization matrix associated with pressure, defined by $E := M\mathscr{E}_p$.

- Lastly, $(\boldsymbol{b_u}, b_p)$ collects the entire influence of the right-hand-side data, $(\boldsymbol{f}, f)$, together with Dirichlet, stress, and interfacial jump source data, and corresponds to the multiplication by M of the right-hand sides of (28) and (29).

A.1. ***Specification of penalty parameters.*** As described, four different kinds of penalty parameters need specification for the LDG formulation of the Stokes equations — three for velocity on the boundary, intraphase, and interphase faces of the mesh, and one for pressure stabilization. Remarks concerning their general specification and particular choices made in this work are provided here.

- In general, strictly positive parameters are sufficient to ensure well-posedness of the final linear system, i.e., to ensure it has the expected trivial kernel of the continuum Stokes operator, and to ensure the inf-sup conditions hold [14; 11; 13].[8] However, this is not a necessary condition. For example, on a regular Cartesian mesh, with purely one-sided intraphase numerical fluxes for $\hat{\boldsymbol{u}}$ and $\hat{\boldsymbol{\sigma}}$ that "upwind" in the opposite direction, as used in this work, one can set the intraphase penalty parameter for velocity to zero, $\tau_{\boldsymbol{u},i} = 0$; see, e.g., [12] in the case of LDG for scalar Poisson problems. On the other hand, a choice of penalty parameter which is too large can impact discretization accuracy and overall conditioning of the final linear operator as well as multigrid solver efficiency.

- If Γ_D is nonempty, then $\tau_{\boldsymbol{u},\partial}$ must be positive to ensure well-posedness.

[8] Subtleties may arise in the multiphase case owing to the nonpenalization of pressure jumps across the interface, depending on mesh topology and the choice of numerical fluxes $\hat{\boldsymbol{u}}$ and $\hat{\boldsymbol{\sigma}}$. Discussion is deferred to future work.

- Although no specific lower bound on parameter values is required for the LDG system to be well-posed, for consistent discretization behavior (including invariance with respect to stretching the mesh as well as preserving spectral characteristics as the mesh is refined), velocity penalty parameters should scale proportionally to h^{-1} and pressure penalty parameters should scale proportionally to h. For anisotropic meshes, one can be more precise and require that penalty parameters on a particular mesh face scale appropriately with the measure of the face divided by the measure of the elements on either side [14], but this extra kind of precision is not pursued here.
- To ensure correct scaling with respect to ellipticity coefficient, penalty parameters should also scale with viscosity. For velocity penalty parameters, this implies $\tau_{\boldsymbol{u},\partial} \sim \mu^-$ and $\tau_{\boldsymbol{u},0} \sim \mu$, where μ is the local value of the viscosity on the mesh face in question, and for interphase velocity penalty parameters, τ_{ij} should scale with the smaller of the two viscosities on either side of the interface [41]. Meanwhile, the pressure penalty parameter should scale inversely proportionally to the local viscosity to preserve the spectral characteristics of the Stokes operator; see, e.g., [11].
- One can also choose to scale τ with the polynomial degree, and this can be important for the study of DG methods utilizing very high-degree polynomials; e.g., one could scale according to $\tau_{\boldsymbol{u}} \sim p^2$. However, in this work, only moderate-order polynomials are used and a linear scaling in p is applied, as defined next.

The specification of the pressure penalty stabilization parameter is one of the main subjects of this work and is discussed in the main article. Regarding the remaining penalty parameters, unless otherwise specified, the following values have been used throughout this work:

- On faces associated with the imposition of velocity Dirichlet boundary conditions, $\tau_{\boldsymbol{u},\partial} = 10p\mu^-/h$, where p is the (one-dimensional) polynomial degree, μ^- is the local viscosity of the face, and h is the typical element size.
- For intraphase penalty parameters, $\tau_{\boldsymbol{u},0} = 0$ on all Cartesian mesh examples, and $\tau_{\boldsymbol{u},0} = 0.5p\mu/h$ on all examples using semiunstructured meshes, where μ is the local viscosity of the face.
- For interphase penalty parameters, $\tau_{\boldsymbol{u},ij} = Cp\min(\mu^-, \mu^+)/h$, where $\mu^\pm$ is the local viscosity of the two phases on either side of the interfacial mesh face. Here, $C = 3$ in the case of Cartesian grid meshes, whereas $C = 8$ in the case of implicitly defined meshes, which benefit from slightly increased penalty stabilization owing to their marginally less uniform variety of element shapes next to the interface.

- Lastly, regarding the time-dependent Stokes equations and their generalized form of the pressure penalty parameter τ_p (see (33)), $\tau_0 = 0.5p$ is the prefactor used for the limiting case of vanishing viscosity.

These values have been chosen based on the dual goals of obtaining high-order discretization accuracy as well as good multigrid performance, and follow typical values employed in prior work [41].

A.2. ***Viscosity-upwinded numerical fluxes.*** As discussed in the main body of the article, viscosity-upwinded numerical fluxes [41] are utilized for the multiphase Stokes problems. This corresponds to defining λ in the numerical flux functions $\hat{\boldsymbol{u}}$, (18), and $\hat{\boldsymbol{\sigma}}$, (23), for interfacial mesh faces as (see also Figure 13)

$$\lambda = \begin{cases} 0 & \text{if } \mu^- < \mu^+, \\ 0.5 & \text{if } \mu^- = \mu^+, \\ 1 & \text{if } \mu^- > \mu^+. \end{cases} \tag{32}$$

Accordingly, $\hat{\boldsymbol{u}}$ is biased to the more viscous phase, and $\hat{\boldsymbol{\sigma}}$ is biased to the less viscous phase, with interfacial jump data $\boldsymbol{g}_{ij}$ and $\boldsymbol{h}_{ij}$ incorporated appropriately.

Appendix B: Operator coarsening geometric multigrid methods

In this section, we describe an operator coarsening strategy which allows one to compute the discrete Stokes operator $\mathscr{A}_h$ on every level of the mesh hierarchy, without having to explicitly build the mesh. This approach is equivalent in function to a purely geometric multigrid method, but provides a variety of convenient benefits, including that (i) elements and faces do not need to be enumerated on coarse meshes, (ii) construction of quadrature schemes for coarse mesh elements or faces can be avoided, (iii) the viscosity coefficient μ is automatically coarsened down the mesh hierarchy in a manner consistent with performing repeated L^2 projections, and (iv) LDG operators, such as the discrete gradient and penalty stabilization operators, are built automatically such that the chosen numerical flux functions of the finest mesh are inherited consistently by the coarse meshes. The technique is also amenable to simple block-sparse linear algebra routines, providing an opportunity to optimize the implementation. These schemes were originally derived for LDG methods applied to elliptic problems in [20] and then extended to variable-coefficient elliptic interface problems with large viscosity jumps in [41]; here, they are extended to variable-viscosity multiphase Stokes problems.

The operator coarsening approach is similar to the "RAT" paradigm often seen in multigrid methods, where the coarse mesh operator is defined by the fine mesh operator, pre- and postmultiplied by the restriction ("R") and interpolation ("T") operators, respectively. Such an approach is often applied to the primary elliptic operator itself, e.g., the discrete Laplacian operator. However, as shown in [20], this

approach leads to breakdown of multigrid performance for LDG methods. Instead, a more appropriate strategy is to apply "RAT" to the individual discrete gradient and divergence operators underlying the LDG method. In [20], it is proven this is equivalent to a purely geometric method, and it is straightforward to extend the methods therein to prove the same is true for the LDG formulation of the Stokes equations described above in Appendix A. We summarize this construction here and omit the proof.

Given a general operator $A : V_h \to V_h$ defined on a fine mesh, its coarsened counterpart $\mathscr{C}(A) : V_{2h} \to V_{2h}$ on a coarse mesh is defined variationally. Specifically, $\mathscr{C}(A)$ is defined such that

$$(\mathscr{C}(A)u, v)_{V_{2h}} = (AI_{2h}^h u, I_{2h}^h v)_{V_h}$$

for all $u, v \in V_{2h}$; here $(\cdot, \cdot)$ denotes the standard inner product, and V_{2h} denotes the piecewise polynomial space associated with the coarse mesh. Equivalently, as a matrix acting on coefficient vectors in the chosen basis,

$$\mathscr{C}(A) = R_h^{2h} A I_{2h}^h = M_{2h}^{-1} (I_{2h}^h)^\mathsf{T} M_h A I_{2h}^h$$

where M_h and M_{2h} are the mass matrices of the two meshes.

B.1. ***Time-independent Stokes equations.*** We derive the operator coarsening strategy for the Stokes system (30), as follows:

(1) The mass matrix of the coarse mesh is given by $M_{2h} = (I_{2h}^h)^\mathsf{T} M_h I_{2h}^h$ [20]. If $M_{\mu,h}$ is a μ-weighted mass matrix on the fine mesh, its coarsened counterpart is given by $M_{\mu,2h} = (I_{2h}^h)^\mathsf{T} M_{\mu,h} I_{2h}^h$ [41].

(2) The coarsened operators making up the Stokes operator are given by $\mathscr{C}(G_h)$, $\mathscr{C}(\mathscr{G}_h)$, $\frac{1}{2} M_{2h} \mathscr{C}(M_h^{-1} \widetilde{E}_h)$, and $2 M_{2h} \mathscr{C}(M_h^{-1} E_h)$ for the discrete gradient operator, the adjoint form of the discrete gradient operator, the velocity penalty stabilization operator, and the pressure penalty stabilization operator, respectively. In particular, note the $\frac{1}{2}$ and 2 prefactors in the coarsened penalty operators — these correspond to the observation that velocity penalty parameters scale proportionally to h^{-1} and pressure penalty parameters scale proportionally to h, and that one should preserve this scaling across the full multigrid hierarchy to attain ideal multigrid performance [41].

(3) Lastly, the discrete Stokes operator on the coarse mesh is formed by computing the viscous operator in (31) using the coarsened discrete gradient operators, and then building the overall operator using the form given in (30).

Algorithm 2 defines the overall operator coarsening scheme, to be applied recursively down the mesh hierarchy. We note that in this algorithm, both of the discrete gradient

1: $M_{2h} := (I_{2h}^h)^\mathsf{T} M_h I_{2h}^h$
2: $M_{\mu,2h} := (I_{2h}^h)^\mathsf{T} M_{\mu,h} I_{2h}^h$
3: $G_{2h} := M_{2h}^{-1}(I_{2h}^h)^\mathsf{T} M_h G_h I_{2h}^h$
4: $\mathcal{G}_{2h} := M_{2h}^{-1}(I_{2h}^h)^\mathsf{T} M_h \mathcal{G}_h I_{2h}^h$; (equivalently, $\mathcal{G}_{2h,i} := -M_{2h}^{-1} G_{2h,i}^\mathsf{T} M_{2h}$)
5: $\widetilde{E}_{2h} := \frac{1}{2}(I_{2h}^h)^\mathsf{T} \widetilde{E}_h I_{2h}^h$
6: $E_{2h} := 2(I_{2h}^h)^\mathsf{T} E_h I_{2h}^h$
7: Build the coarsened viscous operator in $d \times d$ block form, with (i, j)-th block given by

$$A_{2h,ij} := \delta_{ij}\left(\sum_{k=1}^{d} G_{2h,k}^\mathsf{T} M_{\mu,2h} G_{2h,k}\right) + \gamma G_{2h,j}^\mathsf{T} M_{\mu,2h} G_{2h,i} + \delta_{ij}\widetilde{E}_{2h}$$

8: Form the coarsened Stokes operator

$$\mathcal{A}_{2h} := \begin{pmatrix} A_{2h} & M_{2h}\mathcal{G}_{2h} \\ \mathcal{G}_{2h}^\mathsf{T} M_{2h} & -E_{2h} \end{pmatrix}$$

Algorithm 2. Construction of coarse mesh operators for the time-independent multiphase Stokes equations, given fine mesh operators M_h, $M_{\mu,h}$, G_h, $\mathcal{G}_h$, $\widetilde{E}_h$, E_h.

operators G and $\mathcal{G}$ are coarsened; this is to assist in overall clarity; however, it also suffices to coarsen just one.

B.2. ***Time-dependent Stokes equations.*** As discussed in the main article, a modification to the pressure penalty parameter scaling is appropriate for the time-dependent Stokes equations. In this setting, the relation $\tau_p = \tau h/\mu$ is replaced with

$$\tau_p = \left(\frac{h\rho}{\tau_0 \delta} + \frac{\mu}{\tau h}\right)^{-1}. \tag{33}$$

To incorporate this scaling into the operator coarsening strategy described above, the prefactor of 2 in the coarsening of the pressure stabilization operator, $E_{2h} = 2M_{2h}\mathcal{C}(M_h^{-1}E_h)$, should be appropriately modified to reflect the property that the scaling in (33) may change from $\tau_p \sim h$ to $\tau_p \sim 1/h$ down the multigrid hierarchy. In this work, this is implemented through the coarsening of *two* stabilization operators for pressure, which are then combined into a net result via a harmonic weighting. Specifically, on the finest mesh we define E_μ and E_ρ such that

$$u^T E_\mu v = \int_{\Gamma_0} \frac{\tau h}{\mu} [\![u]\!][\![v]\!] \quad \text{and} \quad u^T E_\rho v = \int_{\Gamma_0} \frac{\tau_0 \delta}{h\rho} [\![u]\!][\![v]\!]$$

hold for every $u, v \in V_h$. Then, E_μ is coarsened using a prefactor of 2, and E_ρ is coarsened using a prefactor of $\frac{1}{2}$. The two block-sparse operators are then combined using a heuristic based on the Frobenius norm of each block. Specifically, if $E_{\mu,ij}$ denotes the (i, j)-th block of E_μ (similarly for E_ρ), where individual blocks

1: $M_{2h} := (I_{2h}^h)^\mathsf{T} M_h I_{2h}^h$
2: $M_{\mu,2h} := (I_{2h}^h)^\mathsf{T} M_{\mu,h} I_{2h}^h$
3: $M_{\rho,2h} := (I_{2h}^h)^\mathsf{T} M_{\rho,h} I_{2h}^h$
4: $G_{2h} := M_{2h}^{-1} (I_{2h}^h)^\mathsf{T} M_h G_h I_{2h}^h$
5: $\mathcal{G}_{2h} := M_{2h}^{-1} (I_{2h}^h)^\mathsf{T} M_h \mathcal{G}_h I_{2h}^h$; (equivalently, $\mathcal{G}_{2h,i} := -M_{2h}^{-1} G_{2h,i}^\mathsf{T} M_{2h}$)
6: $\widetilde{E}_{2h} := \frac{1}{2} (I_{2h}^h)^\mathsf{T} \widetilde{E}_h I_{2h}^h$
7: $E_{\mu,2h} := 2 (I_{2h}^h)^\mathsf{T} E_{\mu,h} I_{2h}^h$
8: $E_{\rho,2h} := \frac{1}{2} (I_{2h}^h)^\mathsf{T} E_{\rho,h} I_{2h}^h$
9: Combine $E_{\mu,2h}$ and $E_{\rho,2h}$ into one effective pressure stabilization operator, E_{2h}, via (34)
10: Build the coarsened viscous operator in $d \times d$ block form, with (i, j)-th block given by

$$A_{2h,ij} := \delta_{ij} \left(\sum_{k=1}^{d} G_{2h,k}^\mathsf{T} M_{\mu,2h} G_{2h,k} \right) + \gamma G_{2h,j}^\mathsf{T} M_{\mu,2h} G_{2h,i} + \delta_{ij} \widetilde{E}_{2h}$$

11: Form the coarsened Stokes operator,

$$\mathcal{A}_{2h} := \begin{pmatrix} (1/\delta) M_{\rho,2h} + A_{2h} & M_{2h} \mathcal{G}_{2h} \\ \mathcal{G}_{2h}^\mathsf{T} M_{2h} & -E_{2h} \end{pmatrix}$$

Algorithm 3. Construction of coarse mesh operators for the time-dependent multiphase Stokes equations, given fine mesh operators M_h, $M_{\mu,h}$, $M_{\rho,h}$, G_h, $\mathcal{G}_h$, $\widetilde{E}_h$, $E_{\mu,h}$, $E_{\rho,h}$.

correspond to individual elements of the mesh, then we define E such that

$$E_{ij} = \left(\frac{\|E_{\rho,ij}\|_F}{\|E_{\mu,ij}\|_F + \|E_{\rho,ij}\|_F} \right)^2 E_{\mu,ij} + \left(\frac{\|E_{\mu,ij}\|_F}{\|E_{\mu,ij}\|_F + \|E_{\rho,ij}\|_F} \right)^2 E_{\rho,ij}. \quad (34)$$

This formula is based on the identity $(1/x + 1/y)^{-1} = (x^2 y + y^2 x)/(x + y)^2$ and is simply a heuristic means of recombining E_μ and E_ρ into one operator based on the form of (33); other methods of deriving an operator coarsening strategy for the pressure stabilization operator are likely possible. Algorithm 3 summarizes the operator coarsening strategy for this generalization to the time-dependent Stokes equations, which, as before, is to be applied recursively to define all necessary operators down the mesh hierarchy.

Appendix C: Grid convergence analyses

Throughout this work, grid convergence analyses have been used to measure the discretization order of accuracy. In every test case, the same exact solution is used to define the source data $\boldsymbol{f}$, f, interfacial jump data $\boldsymbol{g}_{ij}$, $\boldsymbol{h}_{ij}$, and boundary data $\boldsymbol{g}_\partial$, $\boldsymbol{h}_\partial$ in the governing equations. This exact solution is based on a simple smooth

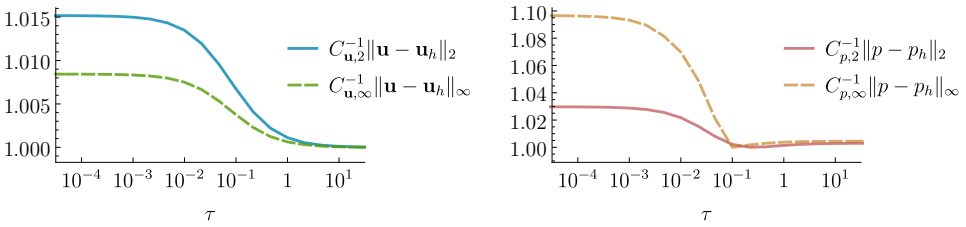

Figure 14. Discretization accuracy as a function of pressure penalty stabilization parameter. Here, the errors in velocity (left) and pressure (right) are normalized by their minimum attained value.

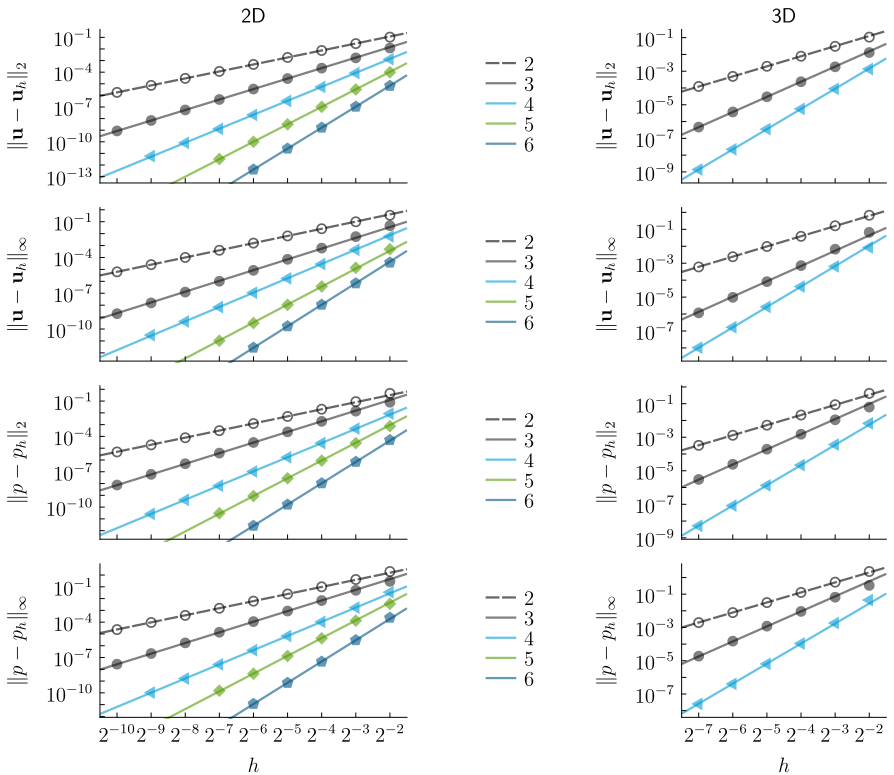

Figure 15. Grid convergence analysis for the single-phase Stokes problem in standard form, with $\mu = 1$ and periodic boundary conditions (see Section 5.1). Here, h denotes the mesh element size and the lines of indicated slope illustrate the asymptotic convergence rate in the corresponding error norm, e.g., a slope of 4 indicates fourth-order accuracy. Polynomial degrees are symbolized by ○, ●, ◀, ◆, ⬟ for $p = 1, 2, 3, 4, 5$, respectively.

sinusoidal wave, for both velocity and pressure, with a translation depending on the component (and phase in the multiphase case) to avoid any coincidental alignment with the mesh or cancellation in jump data. It is defined by the velocity field

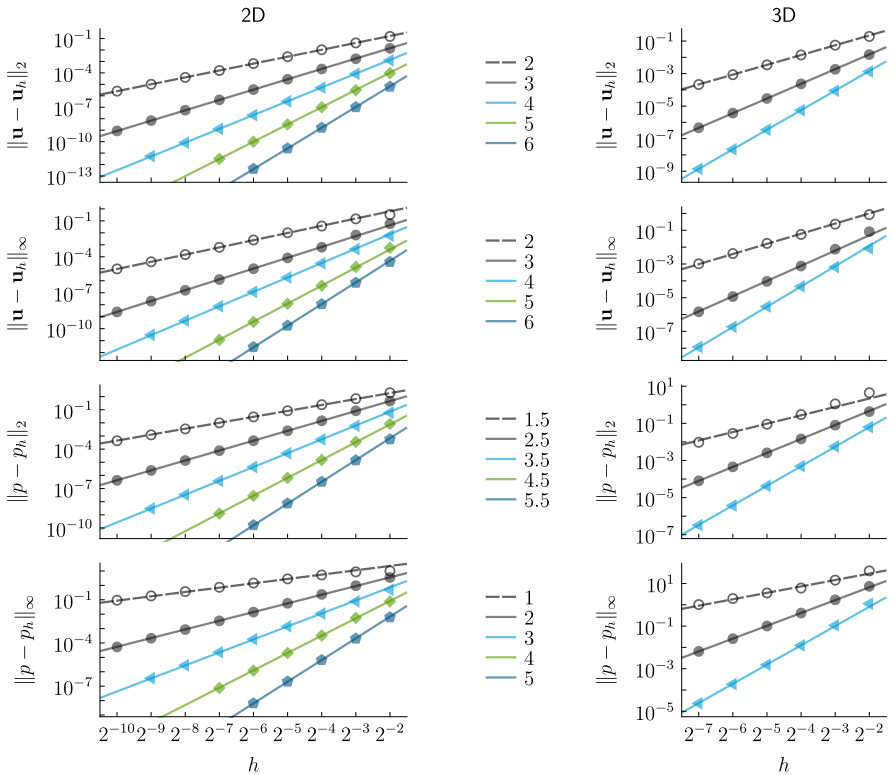

Figure 16. Grid convergence analysis for the single-phase Stokes problem in standard form, with $\mu = 1$ and velocity Dirichlet boundary conditions (see Section 5.2). (Results for the test problem considered in Section 5.3, i.e., with stress boundary conditions, have similar characteristics.) Here, h denotes the mesh element size and the lines of indicated slope illustrate the asymptotic convergence rate in the corresponding error norm. Polynomial degrees are symbolized by ○, ●, ◀, ◆, ⬟ for $p = 1, 2, 3, 4, 5$, respectively.

$\boldsymbol{u} : \Omega \to \mathbb{R}^d$, $\boldsymbol{u} = (u_1, \ldots, u_d)$, and pressure field $p : \Omega \to \mathbb{R}$, where

$$u_i(x) = \prod_{j=1}^{d} \sin 2\pi(x_j - 0.2i - 0.25(\chi - 1)),$$
$$p(x) = \mu_\chi \prod_{j=1}^{d} \sin 2\pi(x_j + 0.2 - 0.25(\chi - 1)),$$

where the term $0.2i$ represents a component-dependent shift, χ is the phase (i.e., $x \in \Omega_\chi$), and μ_χ is a typical viscosity coefficient of phase χ.

Whether the continuum (steady-state) Stokes problem is posed with velocity Dirichlet boundary conditions, or stress boundary conditions, or periodic boundary

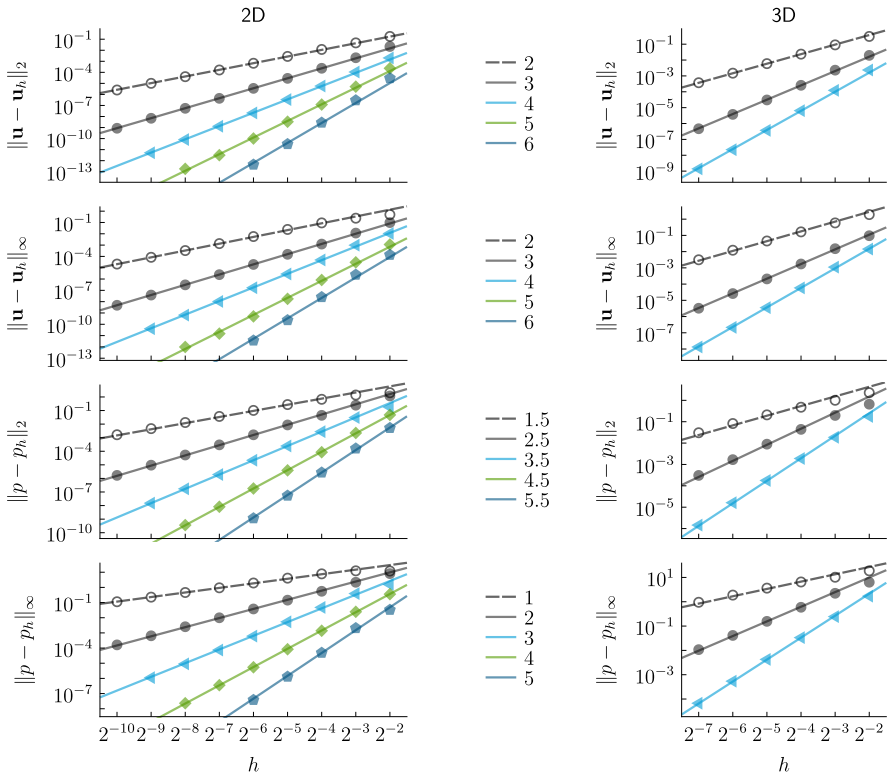

Figure 17. Grid convergence analysis for the single-phase Stokes problem in viscous-stress form, with nonconstant variable viscosity and stress boundary conditions (see Section 5.4). Here, h denotes the mesh element size and the lines of indicated slope illustrate the asymptotic convergence rate in the corresponding error norm. Polynomial degrees are symbolized by ○, ●, ◀, ◆, ⬟ for $p = 1, 2, 3, 4, 5$, respectively.

conditions, there is always an associated kernel of dimension at least one.[9] This kernel is referred to as the "trivial kernel", for it consists of constant-valued velocity fields and/or constant pressure fields, and possibly additional modes in the case of the viscous-stress form with stress boundary conditions (e.g., the velocity field $(x, y) \mapsto (-y, x)$, for which $\nabla \boldsymbol{u} + \nabla \boldsymbol{u}^{\mathsf{T}}$ is zero). Since the Stokes operator is symmetric, it follows that the right-hand-side data $(\boldsymbol{b}_{\boldsymbol{u}}, b_p)$ must be orthogonal to the kernel; this is always the case for the method of manufactured solution applied here. However, the continuum solution is only unique up to modes in the kernel, and the discrete solution computed via the multigrid preconditioned GMRES method may contain arbitrary modes of the corresponding discrete kernel. To appropriately measure the discrete error, these modes are therefore disregarded. In particular, we

[9]The dimension of the kernel may be smaller for time-dependent Stokes problems; for example, if Neumann or stress boundary conditions are applied, the kernel is zero-dimensional.

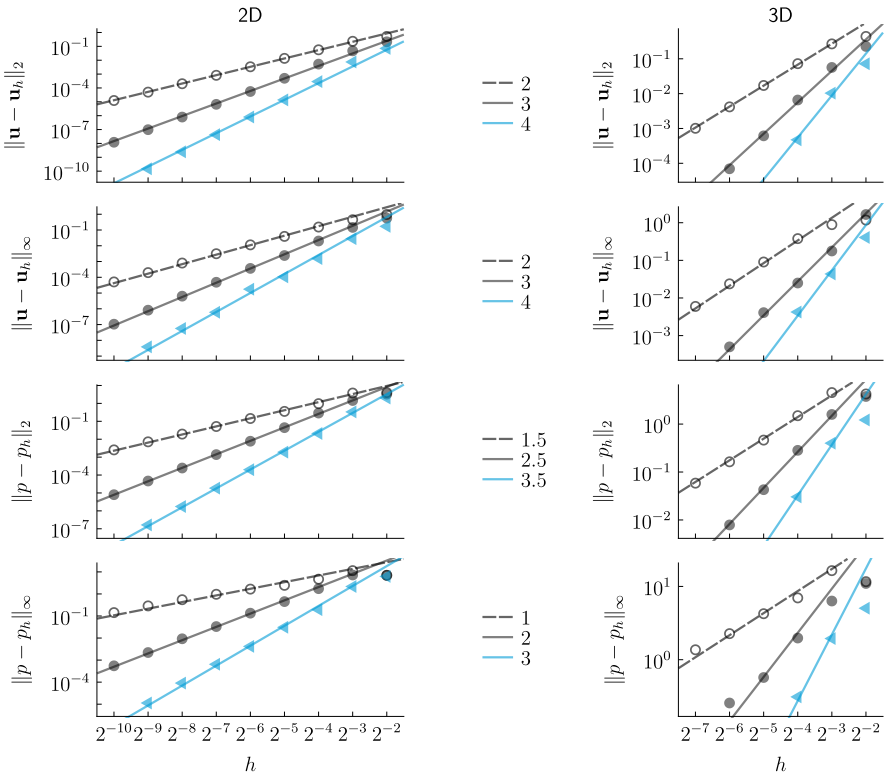

Figure 18. Grid convergence analysis for the single-phase Stokes problem in standard form, with $\mu = 1$ and velocity Dirichlet boundary conditions, in a unit diameter spherical domain using implicitly defined meshes (see Section 5.5). Here, h denotes the typical mesh element size and the lines of indicated slope illustrate the asymptotic convergence rate in the corresponding error norm. Polynomial degrees are symbolized by $\circ$, $\bullet$, $\blacktriangleleft$ for $p = 1, 2, 3$, respectively.

compute the discrete error $(\boldsymbol{u}-\boldsymbol{u}_h,\, p-p_h)$ and nullify any components in the kernel through a simple Gram–Schmidt process applied to a basis of the kernel, known ahead of time. The resulting discrete error $(\boldsymbol{u}-\boldsymbol{u}_h,\, p-p_h)$ is then measured in the L^2 norm and the maximum norm, and is reported in the following collection of graphs, Figures 15–22. In each case, data points represent the measured error, and the lines of indicated slope are plotted to illustrate the asymptotic convergence rate. In some cases, the discrete error is saturated by numerical conditioning associated with double-precision arithmetic, forcing the cessation of high-order convergence; these data points are excluded from the graphs.

C.1. ***Impact of pressure penalty parameter on discretization error.*** To supplement the discussion in Section 4, shown here is a test case examining the impact of the pressure penalty parameter τ on discretization error. We consider a single-phase

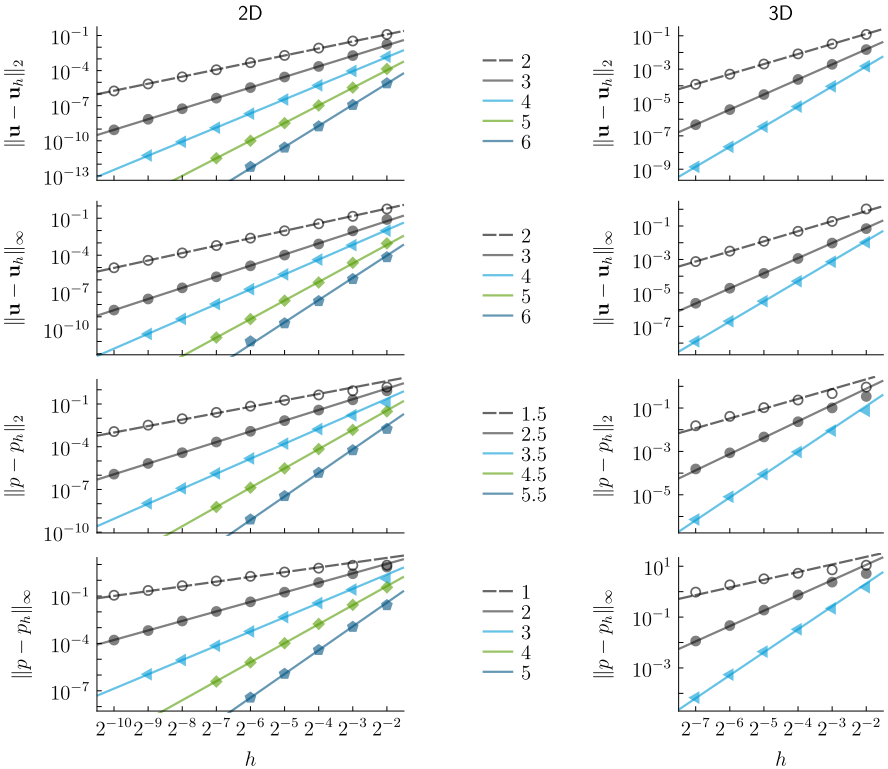

Figure 19. Grid convergence analysis for the multiphase Stokes problem in viscous-stress form, in which $\Omega_1 = \left(\frac{1}{4}, \frac{3}{4}\right)^d$ has viscosity $\mu_1 = 10^{-6}$, and $\Omega_2 = (0,1)^d \setminus \overline{\Omega_1}$ has viscosity $\mu_2 = 1$, with stress boundary conditions (see Section 5.6). (Results for the other viscosity ratios considered, i.e., $\mu_1 \in \{10^{-3}, 10^{+3}, 10^{+6}\}$, have similar characteristics.) Here, h denotes the mesh element size and the lines of indicated slope illustrate the asymptotic convergence rate in the corresponding error norm. Polynomial degrees are symbolized by ○, ●, ◀, ◆, ⬟ for $p = 1, 2, 3, 4, 5$, respectively.

Stokes problem in standard form, with $\mu = 1$ and periodic boundary conditions, on a uniform Cartesian grid with $p = 2$ biquadratic elements. In the following tests, we consider extreme values of τ and utilize a direct solver instead of multigrid (thereby eliminating possible issues of multigrid nonconvergence associated with extreme values of τ); as such, a coarser grid of 16×16 is used. With the same range of τ as used in the discussion of Section 4, Figure 14 shows the discrete error in the velocity and pressure in the L^2 and maximum norms; in each case the error is normalized by the minimum attained value, e.g., $C_{\boldsymbol{u},2} = \min_\tau \|\boldsymbol{u} - \boldsymbol{u}_h\|_2$, and similarly for the other quantities shown. (The order of magnitude of these errors can be inferred from the 2D results of Figure 15.) According to Figure 14, note that τ has relatively little influence on the error in both velocity and pressure, with maximal discretization errors at most 10% greater than optimal. Although only a

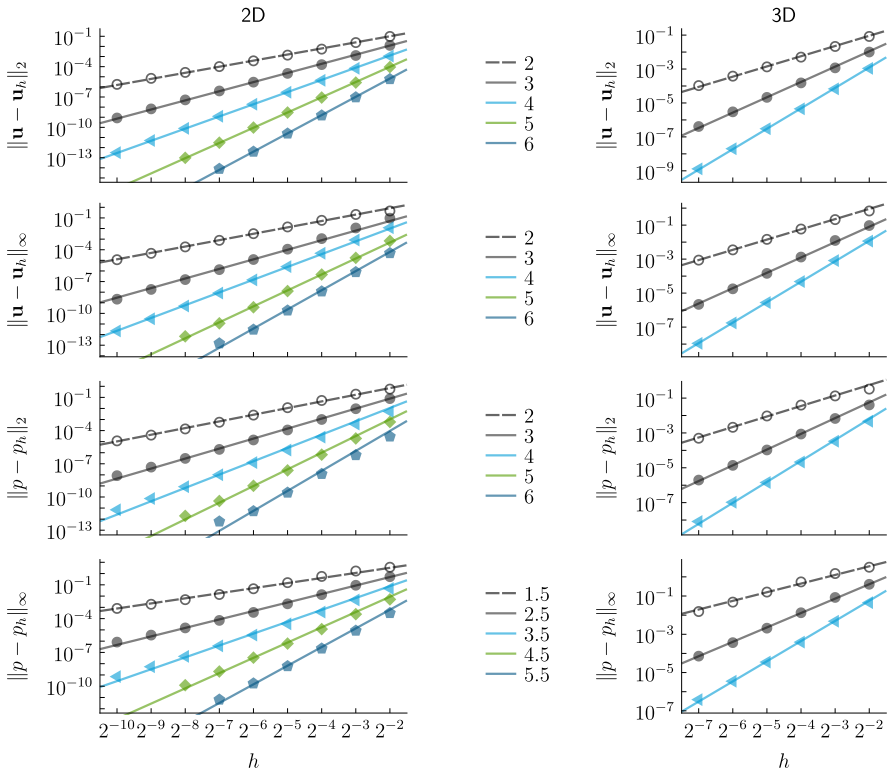

Figure 20. Grid convergence analysis for the time-dependent single-phase Stokes problem in standard form, with $\mu = 10^{-2}$ and $\delta = 0.1h$, corresponding to $\mathrm{Re} \approx 100$ (see Section 6.1). Here, h denotes the mesh element size and the lines of indicated slope illustrate the asymptotic convergence rate in the corresponding error norm. Polynomial degrees are symbolized by ○, ●, ◀, ◆, ⬟ for $p = 1, 2, 3, 4, 5$, respectively.

minor improvement, note also that the best error in pressure is attained when τ is approximately equal to or larger than the optimal τ values found in Table 1. Overall, this kind of behavior has been observed across many of the examples considered in this work, leading to the conclusion that, at least for relatively well-resolved Stokes problems, the discretization error is largely insensitive to the value of τ, thereby allowing this parameter to be tuned for excellent multigrid performance.

Acknowledgements

This research was supported by the Applied Mathematics Program of the U.S. DOE Office of Advanced Scientific Computing Research under contract number DE-AC02-05CH11231. Some computations used resources of the National Energy Research Scientific Computing Center (NERSC), a U.S. DOE Office of Science User Facility operated under contract number DE-AC02-05CH11231.

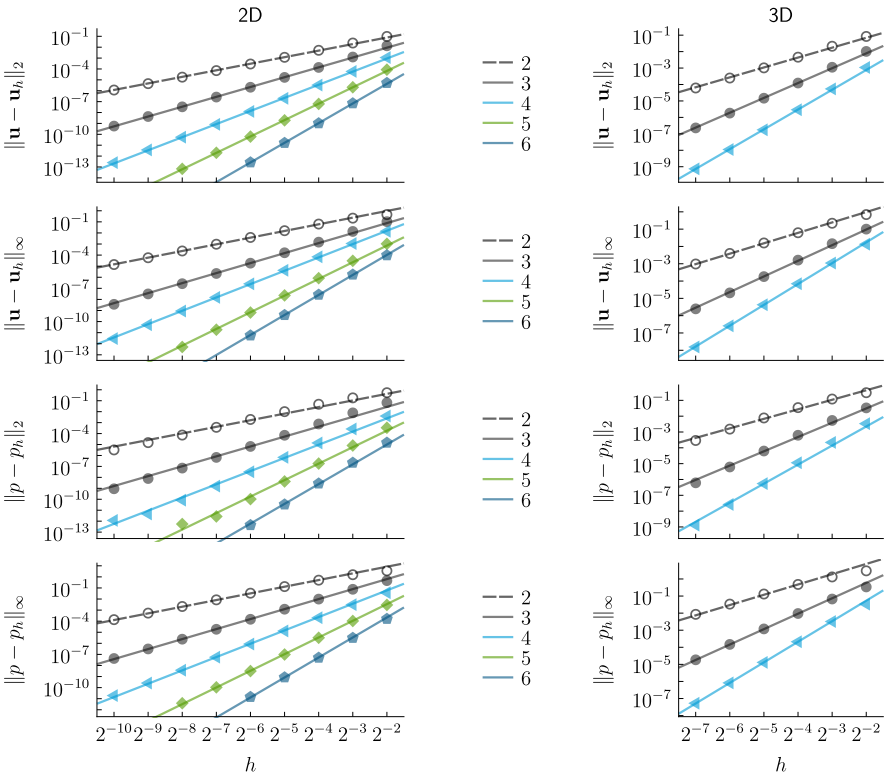

Figure 21. Grid convergence analysis for the time-dependent single-phase Stokes problem in standard form, with $\mu = 10^{-4}$ and $\delta = 0.1h$, corresponding to $\mathrm{Re} \approx 10\,000$ (see Section 6.1). Here, h denotes the mesh element size and the lines of indicated slope illustrate the asymptotic convergence rate in the corresponding error norm. Polynomial degrees are symbolized by ○, ●, ◀, ◆, ⬟ for $p = 1, 2, 3, 4, 5$, respectively.

References

[1] J. H. Adler, T. R. Benson, and S. P. MacLachlan, *Preconditioning a mass-conserving discontinuous Galerkin discretization of the Stokes equations*, Numer. Linear Algebra Appl. **24** (2017), no. 3, e2047. MR Zbl

[2] B. Aksoylu and Z. Unlu, *Robust preconditioners for the high-contrast Stokes equation*, J. Comput. Appl. Math. **259** (2014), no. part B, 944–954. MR Zbl

[3] D. N. Arnold, F. Brezzi, B. Cockburn, and L. D. Marini, *Unified analysis of discontinuous Galerkin methods for elliptic problems*, SIAM J. Numer. Anal. **39** (2002), no. 5, 1749–1779. MR

[4] P. Bauer, V. Klement, T. Oberhuber, and V. Žabka, *Implementation of the Vanka-type multigrid solver for the finite element approximation of the Navier–Stokes equations on GPU*, Comput. Phys. Commun. **200** (2016), 50–56. MR

[5] M. Benzi, G. H. Golub, and J. Liesen, *Numerical solution of saddle point problems*, Acta Numer. **14** (2005), 1–137. MR

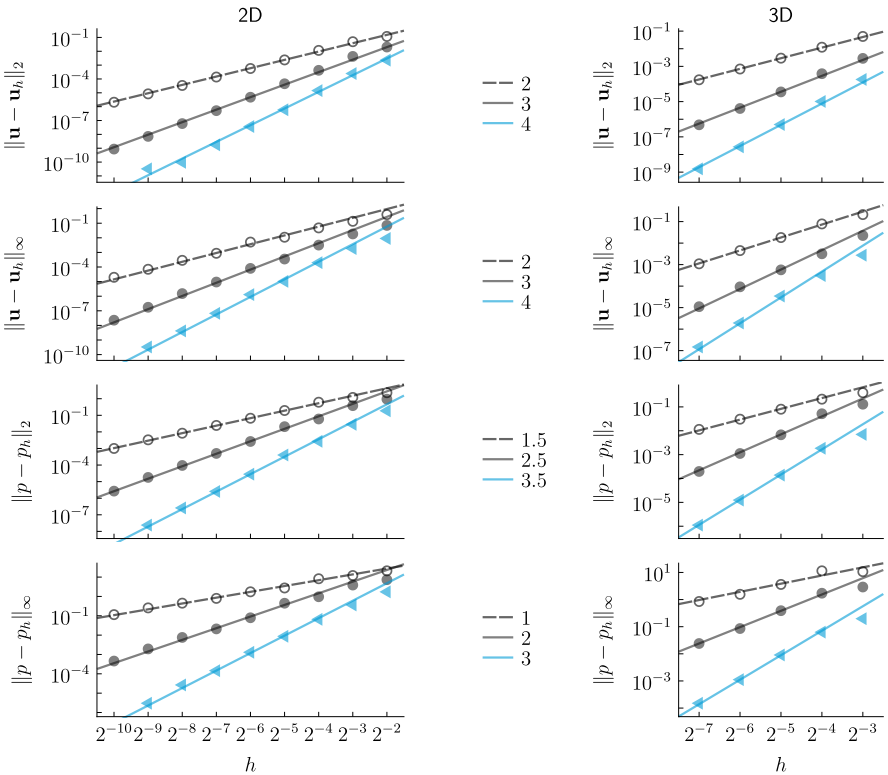

Figure 22. Grid convergence analysis for the time-dependent multiphase Stokes problem in viscous-stress form, for a water bubble surrounded by gas, with $\rho_{\text{water}} = 1$, $\rho_{\text{gas}} = 0.001$, $\mu_{\text{water}} = 1$, $\mu_{\text{gas}} = 0.0002$, and $\delta = 0.1h$, together with velocity Dirichlet boundary conditions (see Section 6.2). (Results for the opposite case, i.e., a gas bubble surrounded by water, have similar characteristics.) Here, h denotes the typical mesh element size and the lines of indicated slope illustrate the asymptotic convergence rate in the corresponding error norm. Polynomial degrees are symbolized by $\circ$, $\bullet$, $\blacktriangleleft$ for $p = 1, 2, 3$, respectively.

[6] D. Borzacchiello, E. Leriche, B. Blottière, and J. Guillet, *Box-relaxation based multigrid solvers for the variable viscosity Stokes problem*, Comput. Fluids **156** (2017), 515–525. MR Zbl

[7] A. Brandt and O. E. Livne, *Multigrid techniques: 1984 guide with applications to fluid dynamics*, Classics in Applied Mathematics, no. 67, Society for Industrial and Applied Mathematics, Philadelphia, 2011. MR

[8] W. L. Briggs, V. E. Henson, and S. F. McCormick, *A multigrid tutorial*, 2nd ed., Society for Industrial and Applied Mathematics, Philadelphia, 2000. MR Zbl

[9] L. Chen, *Multigrid methods for saddle point systems using constrained smoothers*, Comput. Math. Appl. **70** (2015), no. 12, 2854–2866. MR

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

[11] B. Cockburn, G. Kanschat, and D. Schötzau, *LDG methods for Stokes flow problems*, Numerical mathematics and advanced applications (F. Brezzi, A. Buffa, S. Corsaro, and A. Murli, eds.), Springer, 2003, pp. 755–764. MR Zbl

[12] B. Cockburn and B. Dong, *An analysis of the minimal dissipation local discontinuous Galerkin method for convection-diffusion problems*, J. Sci. Comput. **32** (2007), no. 2, 233–262. MR Zbl

[13] B. Cockburn, G. Kanschat, and D. Schötzau, *A locally conservative LDG method for the incompressible Navier–Stokes equations*, Math. Comp. **74** (2005), no. 251, 1067–1095. MR Zbl

[14] B. Cockburn, G. Kanschat, D. Schötzau, and C. Schwab, *Local discontinuous Galerkin methods for the Stokes system*, SIAM J. Numer. Anal. **40** (2002), no. 1, 319–343. MR Zbl

[15] B. Cockburn and C.-W. Shu, *The local discontinuous Galerkin method for time-dependent convection-diffusion systems*, SIAM J. Numer. Anal. **35** (1998), no. 6, 2440–2463. MR Zbl

[16] C. Coley, J. Benzaken, and J. A. Evans, *A geometric multigrid method for isogeometric compatible discretizations of the generalized Stokes and Oseen problems*, Numer. Linear Algebra Appl. **25** (2018), no. 3, e2145. MR Zbl

[17] D. Drzisga, L. John, U. Rüde, B. Wohlmuth, and W. Zulehner, *On the analysis of block smoothers for saddle point problems*, SIAM J. Matrix Anal. Appl. **39** (2018), no. 2, 932–960. MR Zbl

[18] L. Elsner and V. Mehrmann, *Convergence of block iterative methods for linear systems arising in the numerical solution of Euler equations*, Numer. Math. **59** (1991), no. 6, 541–559. MR Zbl

[19] P. E. Farrell, Y. He, and S. P. MacLachlan, *A local Fourier analysis of additive Vanka relaxation for the Stokes equations*, Numer. Linear Algebra Appl. (2020), e2306.

[20] D. Fortunato, C. H. Rycroft, and R. Saye, *Efficient operator-coarsening multigrid schemes for local discontinuous Galerkin methods*, SIAM J. Sci. Comput. **41** (2019), no. 6, A3913–A3937. MR Zbl

[21] M. Furuichi, D. A. May, and P. J. Tackley, *Development of a Stokes flow solver robust to large viscosity jumps using a Schur complement approach with mixed precision arithmetic*, J. Comput. Phys. **230** (2011), no. 24, 8835–8851. MR Zbl

[22] F. J. Gaspar, Y. Notay, C. W. Oosterlee, and C. Rodrigo, *A simple and efficient segregated smoother for the discrete Stokes equations*, SIAM J. Sci. Comput. **36** (2014), no. 3, A1187–A1206. MR Zbl

[23] B. Gmeiner, M. Huber, L. John, U. Rüde, and B. Wohlmuth, *A quantitative performance study for Stokes solvers at the extreme scale*, J. Comput. Sci. **17** (2016), no. part 3, 509–521. MR

[24] J. L. Guermond, P. Minev, and J. Shen, *An overview of projection methods for incompressible flows*, Comput. Methods Appl. Mech. Engrg. **195** (2006), no. 44-47, 6011–6045. MR Zbl

[25] Y. He and S. P. MacLachlan, *Local Fourier analysis of block-structured multigrid relaxation schemes for the Stokes equations*, Numer. Linear Algebra Appl. **25** (2018), no. 3, e2147. MR Zbl

[26] J. S. Hesthaven and T. Warburton, *Nodal discontinuous Galerkin methods: algorithms, analysis, and applications*, Texts in Applied Mathematics, no. 54, Springer, 2008. MR Zbl

[27] Q. Hong, J. Kraus, J. Xu, and L. Zikatanov, *A robust multigrid method for discontinuous Galerkin discretizations of Stokes and linear elasticity equations*, Numer. Math. **132** (2016), no. 1, 23–49. MR Zbl

[28] G. Kanschat, *Block preconditioners for LDG discretizations of linear incompressible flow problems*, J. Sci. Comput. **22** (2005), 371–384. MR Zbl

[29] G. Kanschat and Y. Mao, *Multigrid methods for H^{div}-conforming discontinuous Galerkin methods for the Stokes equations*, J. Numer. Math. **23** (2015), no. 1, 51–66. MR Zbl

[30] S. Manservisi, *Numerical analysis of Vanka-type solvers for steady Stokes and Navier–Stokes flows*, SIAM J. Numer. Anal. **44** (2006), no. 5, 2025–2056. MR Zbl

[31] M. L. Minion and R. I. Saye, *Higher-order temporal integration for the incompressible Navier–Stokes equations in bounded domains*, J. Comput. Phys. **375** (2018), 797–822. MR Zbl

[32] Y. Notay, *Algebraic multigrid for Stokes equations*, SIAM J. Sci. Comput. **39** (2017), no. 5, S88–S111. MR Zbl

[33] M. A. Olshanskii and E. E. Tyrtyshnikov, *Iterative methods for linear systems: theory and applications*, Society for Industrial and Applied Mathematics, Philadelphia, 2014. MR Zbl

[34] C. W. Oosterlee and F. J. Gaspar, *Multigrid relaxation methods for systems of saddle point type*, Appl. Numer. Math. **58** (2008), no. 12, 1933–1950. MR Zbl

[35] C. W. Oosterlee and P. Wesseling, *A robust multigrid method for a discretization of the incompressible Navier–Stokes equations in general coordinates*, Impact Comput. Sci. Engrg. **5** (1993), no. 2, 128–151. MR Zbl

[36] ———, *Steady incompressible flow around objects in general coordinates with a multigrid solution method*, Numer. Methods Partial Differential Equations **10** (1994), no. 3, 295–308. MR Zbl

[37] R. I. Saye, *High-order quadrature methods for implicitly defined surfaces and volumes in hyperrectangles*, SIAM J. Sci. Comput. **37** (2015), no. 2, A993–A1019. MR Zbl

[38] ———, *Implicit mesh discontinuous Galerkin methods and interfacial gauge methods for high-order accurate interface dynamics, with applications to surface tension dynamics, rigid body fluid-structure interaction, and free surface flow, I*, J. Comput. Phys. **344** (2017), 647–682. MR Zbl

[39] ———, *Implicit mesh discontinuous Galerkin methods and interfacial gauge methods for high-order accurate interface dynamics, with applications to surface tension dynamics, rigid body fluid-structure interaction, and free surface flow, II*, J. Comput. Phys. **344** (2017), 683–723. MR Zbl

[40] ———, *Algoim: algorithms for implicitly defined geometry, level set methods, and Voronoi implicit interface methods*, 2019, software package.

[41] ———, *Efficient multigrid solution of elliptic interface problems using viscosity-upwinded local discontinuous Galerkin methods*, Commun. Appl. Math. Comput. Sci. **14** (2019), no. 2, 247–283. MR Zbl

[42] J. Schöberl and W. Zulehner, *On Schwarz-type smoothers for saddle point problems*, Numer. Math. **95** (2003), no. 2, 377–399. MR Zbl

[43] S. Sivaloganathan, *The use of local mode analysis in the design and comparison of multigrid methods*, Comput. Phys. Commun. **65** (1991), no. 1–3, 246–252. Zbl

[44] M. C. Thompson and J. H. Ferziger, *An adaptive multigrid technique for the incompressible Navier–Stokes equations*, J. Comput. Phys. **82** (1989), no. 1, 94–121. Zbl

[45] S. P. Vanka, *Block-implicit multigrid solution of Navier–Stokes equations in primitive variables*, J. Comput. Phys. **65** (1986), no. 1, 138–158. MR Zbl

[46] ———, *A calculation procedure for three-dimensional steady recirculating flows using multigrid methods*, Comput. Methods Appl. Mech. Engrg. **55** (1986), no. 3, 321–338. Zbl

[47] P. Wesseling, *Principles of computational fluid dynamics*, Springer Series in Computational Mathematics, no. 29, Springer, 2001. MR Zbl

[48] P. Wesseling and C. W. Oosterlee, *Geometric multigrid with applications to computational fluid dynamics*, J. Comput. Appl. Math. **128** (2001), no. 1–2, 311–334. MR Zbl

[49] H. Wobker and S. Turek, *Numerical studies of Vanka-type smoothers in computational solid mechanics*, Adv. Appl. Math. Mech. **1** (2009), no. 1, 29–55. MR

Received January 31, 2020. Revised July 20, 2020.

ROBERT I. SAYE: rsaye@lbl.gov
Mathematics Group, Lawrence Berkeley National Laboratory, Berkeley, CA, United States

Guidelines for Authors

Authors may submit manuscripts in PDF format on-line at the Submission page at msp.org/camcos.

Originality. Submission of a manuscript acknowledges that the manuscript is original and 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 LaTeX but submissions in other varieties of TeX, 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 BibTeX 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. 15 no. 2 2020

Simulating single-coil MRI from the responses of multiple coils 115
Mark Tygert and Jure Zbontar

On a computationally scalable sparse formulation of the multidimensional and nonstationary maximum entropy principle 129
Illia Horenko, Ganna Marchenko and Patrick Gagliardini

Fast multigrid solution of high-order accurate multiphase Stokes problems 147
Robert I. Saye