Available online at www.sciencedirect.com

SCIENCE DIRECT®

Journal of Computational Physics 212 (2006) 703–717

JOURNAL OF COMPUTATIONAL PHYSICS

www.elsevier.com/locate/jcp

A fully implicit, nonlinear adaptive grid strategy

L. Chacón *, G. Lapenta

Theoretical Division, Los Alamos National Laboratory, MS 4717, T-15, Los Alamos, NM 87545, United States

Received 8 April 2005; received in revised form 21 June 2005; accepted 22 July 2005
Available online 23 September 2005

Abstract

We propose an efficient, fully implicit, nonlinear solver for the Beltrami grid generation equation. The Beltrami equation is obtained using Harmonic map theory, and therefore the existence and uniqueness of a solution is guaranteed. The nonlinear solver strategy is based on Newton–Krylov methods, preconditioned here with a multigrid-based method for scalability. Numerical experiments performed for both grid adaptation and grid alignment are presented and demonstrate optimal scaling under grid refinement. We therefore conclude that such a fully nonlinear approach is indeed feasible and efficient.

Keywords: Grid generation; Newton–Krylov; Fully implicit methods; Multigrid; Beltrami equation; Harmonic maps; Moving mesh

1. Introduction

In the numerical simulation of complex physical phenomena, the crucial requirement is predictability, i.e., that the simulation results remain faithful to the actual physical processes. Accordingly, the generation and accumulation of numerical error during the simulation is of special concern, since it introduces distortions that fundamentally alter the fidelity of the simulation. Errors resulting from a lack of spatial resolution are particularly deleterious. However, over-resolving is computationally expensive.

Adaptive grids attempt to provide sufficient resolution where needed while minimizing the computational cost of the simulation. Our emphasis is on moving grid methods (also known as r-refinement methods) in logically structured grids, where grid points are able to move to follow the solution (e.g., see [1–13]). The grid positions are determined from a suitable grid evolution equation. While many grid evolution equations have been proposed in the literature [14–17], here we focus on Harmonic maps [3,4,6,9–11], which are desirable because, under certain conditions, they guarantee the existence and uniqueness of the grid mapping.

However, one drawback of Harmonic function theory is that the resulting grid evolution equation is generally very nonlinear and stiff. Furthermore, multiple-time-scale physics models for which spatial

[*] Corresponding author. Tel.: +1 505 665 6973; fax: +1 505 665 7150.
E-mail address: chacon@lanl.gov (L. Chacón).

0021-9991/$ - see front matter © 2005 Elsevier Inc. All rights reserved.
doi:10.1016/j.jcp.2005.07.023

adaptation is necessary are typically very stiff as well. For such systems, implicit temporal schemes are preferred for efficiency, as they allow one to use time steps comparable to the dynamical timescale of interest in the problem at hand [18]. However, when coupled to a grid evolution equation, such large implicit time steps may not be advantageous from an accuracy standpoint unless both grid and physics equations are solved in a coupled manner.

The coupled nonlinear solution of such physics-grid systems represents, however, a formidable numerical challenge. At the heart of the matter is to demonstrate that developing a scalable, efficient nonlinear algorithm to solve the grid evolution equation *alone* is indeed possible. In fact, while the theory and applications of variational grid generation has become increasingly more sophisticated [14–17], the development of scalable and efficient nonlinear solvers for the resulting nonlinear elliptic equations (MMPDEs) has been lagging. Most authors describe some sort of iterative relaxation scheme, combined with some pseudo-transient treatment of the grid governing equation [9,11,13,19], with very slow convergence properties. Recently, modern nonlinear multigrid methods have been applied to elliptic grid generation PDEs for complex geometries with some success [20–22]. However, to the authors' knowledge, no such advanced grid generation solver has been developed in the context of implicit methods for moving meshes in multidimensional geometries, where grid equation and physics model are intimately coupled (it is worth mentioning that implicit methods have been employed in the past to solve the coupled grid-physics equation in 1D moving grids in many physical applications of interest; see e.g., [8,23–26]).

In this paper, we explore a fully implicit, nonlinear solver for the grid equation, as an integral part of an implicit approach to moving meshes. The coupling of this nonlinear grid solver with a suitable physics model is left for future work. We base our strategy on Newton–Krylov methods [27], which are ideally suited for this task owing to their robustness and the possibility of preconditioning. Specifically, we present in this paper a scalable and robust preconditioning strategy for the numerical generation of Harmonic maps, based on multigrid methods. We demonstrate its performance in two distinct applications, namely, grid adaptation to a scalar field and grid alignment to a vector field. However, it is noteworthy that the approach outlined here can readily be used for the more general purpose of grid generation for complex domains [4,14,19], and therefore the results outlined here may also impact that community.

The rest of the discussion in this paper is organized as follows. We introduce the Newton–Krylov solver engine in Section 2. The Harmonic map theory and its applications to grid generation, as well as the specific form of the inverse Beltrami equation (IBE) used in this work, are outlined in Section 3. The spatial discretization of the IBE is detailed in Section 4. The multigrid preconditioning strategy, which is the cornerstone of the algorithm, is discussed in Section 5. Finally, the results of the numerical experiments are presented in Section 6, and we conclude in Section 7.

2. Newton–Krylov methods

We proceed to give a brief introduction to Newton–Krylov methods (NK). The motivated reader can find extensive discussions on this approach elsewhere [27]. Newton–Krylov methods solve the nonlinear system $\mathbf{G}(\mathbf{x}) = \mathbf{0}$ iteratively by solving successive linear systems of the form:

$$\left.\frac{\partial \mathbf{G}}{\partial \mathbf{x}}\right|_k \delta\mathbf{x}_k = -\mathbf{G}(\mathbf{x}_k),$$

with $\mathbf{x}_{k+1} = \mathbf{x}_k + \delta\mathbf{x}_k$. Nonlinear convergence is achieved when

$$\|\mathbf{G}(\mathbf{x}_k)\|_2 < \epsilon_a + \epsilon_r\|\mathbf{G}(\mathbf{x}_0)\|_2 = \epsilon_t, \tag{1}$$

where $\|\cdot\|_2$ is the ℓ_2-norm (Euclidean norm), $\epsilon_a = N \times 10^{-15}$ (with N the total number of degrees of freedom) is an absolute tolerance to avoid converging below roundoff, ϵ_r is the Newton relative convergence tolerance (set to 10^{-4} in this work), and $\mathbf{G}(\mathbf{x}_0)$ is the initial residual.

Such linear systems are solved iteratively with Krylov methods, which only require matrix–vector products to proceed. Because the linear system matrix is a Jacobian matrix, such matrix–vector products can be implemented Jacobian-free using the Gateaux derivative:

$$\left.\frac{\partial \mathbf{G}}{\partial \mathbf{x}}\right|_k \mathbf{v} = \lim_{\epsilon\to 0} \frac{\mathbf{G}(\mathbf{x}_k + \epsilon\mathbf{v}) - \mathbf{G}(\mathbf{x}_k)}{\epsilon}, \tag{2}$$

where in practice a small but finite ϵ is employed [27]. Thus, the evaluation of the Jacobian–vector product only requires the function evaluation $\mathbf{G}(\mathbf{x}_k + \epsilon\mathbf{v})$, and there is no need to form or store the Jacobian matrix.

An inexact Newton method [28] is used to adjust the convergence tolerance of the Krylov method at every Newton iteration according to the size of the current Newton residual, as follows:

$$\|J_k \delta\mathbf{x}_k + \mathbf{G}(\mathbf{x}_k)\|_2 < \zeta_k \|\mathbf{G}(\mathbf{x}_k)\|_2, \tag{3}$$

where ζ_k is the inexact Newton parameter and $J_k = \frac{\partial \mathbf{G}}{\partial \mathbf{x}}|_k$ is the Jacobian matrix. Thus, the convergence tolerance of the Krylov method is loose when the Newton state vector $\mathbf{x}_k$ is far from the nonlinear solution, but tightens as $\mathbf{x}_k$ approaches the solution. Hence, the linear solver works the hardest when the Newton state vector is closest to the nonlinear root. Superlinear convergence rates of the inexact Newton method are possible if the sequence of ζ_k is chosen properly [27]. Here, we employ the same prescription as in [29]:

$$\begin{aligned}
\zeta_k^A &= \gamma\left(\frac{\|\mathbf{G}(\mathbf{x}_k)\|_2}{\|\mathbf{G}(\mathbf{x}_{k-1})\|_2}\right)^{\alpha}, \\
\zeta_k^B &= \min[\zeta_{\max}, \max(\zeta_k^A, \gamma\zeta_{k-1}^{\alpha})], \\
\zeta_k &= \min\left[\zeta_{\max}, \max\left(\zeta_k^B, \gamma\frac{\epsilon_t}{\|\mathbf{G}(\mathbf{x}_k)\|_2}\right)\right],
\end{aligned}$$

with $\alpha = 1.5$, $\gamma = 0.9$, and $\zeta_{\max} = 0.8$. The convergence tolerance ϵ_t is defined in Eq. (1). In this prescription, the first step ensures superlinear convergence (for $\alpha > 1$), the second avoids volatile decreases in ζ_k, and the last avoids oversolving in the last Newton iteration. We also use a quadratic line-search backtracking algorithm [27] for added robustness of the nonlinear solver.

A further advantage of Krylov methods is that they can be preconditioned by considering the alternate (but equivalent) systems $J_k P_k^{-1} P_k \delta\mathbf{x}_k = -\mathbf{G}_k$ (right preconditioning) or $P_k^{-1} J_k \delta\mathbf{x}_k = -P_k^{-1}\mathbf{G}_k$ (left preconditioning). Such preconditioning step can be straightforwardly and efficiently implemented in the Krylov algorithm as two consecutive matrix–vector products.

A crucial feature of preconditioning is that, while it can substantially improve the convergence properties of the Krylov iteration if $P_k^{-1} \approx J_k^{-1}$, it does not alter the solution of the Jacobian system upon convergence (because the solution $\delta\mathbf{x}_k$ of the preconditioned system is the same as that of the original system). Therefore, one can explore suitable approximations in the preconditioner for efficiency purposes without compromising the accuracy of the final result.

This paper concentrates on the development of a suitable (scalable, robust) preconditioning strategy for the solution of the grid generation equation. Right preconditioning is favored here, because the right-hand side of the Jacobian system $\mathbf{G}_k$ (which is the measure of nonlinear convergence) is unaffected by the preconditioner. In practice, right preconditioning is implemented in two steps, namely, a linear solve $J_k P_k^{-1}\mathbf{y} = -\mathbf{G}_k$, and a posteriori application of the preconditioner to find $\delta\mathbf{x}_k = P_k^{-1}\mathbf{y}$. We use GMRES as the Krylov method of choice for its robustness in non-symmetric, indefinite systems, in anticipation that the grid generation equation may be coupled with a physical system of interest. However, given the effectiveness of the preconditioner strategy developed here (Section 6), the choice of the Krylov method is not very relevant from the efficiency standpoint.

The form of the grid generation equation of choice is the subject of the next section.

3. Grid generation using Harmonic maps

Harmonic maps as a suitable grid generation strategy were first proposed by Dvinsky [3], and later explored by many authors [3,4,6,9–11]. The fundamental advantage that Harmonic map theory features over other grid generation strategies is that it guarantees, under certain conditions and in two dimensions, the existence and uniqueness of a mapping solution to the grid generator equations (see [30,31] for a discussion of the singularity of Harmonic maps in three dimensions). Furthermore, Harmonic grid generation has been shown to be parametrization-independent [19], namely, that the resulting grid in physical space is independent of the choice of parametrization for the physical domain. The implication is that one can solve for the grid mapping between the logical and the parametric space, without involving the physical domain, and that the result will be independent of the parametrization. We proceed to give a brief introduction to Harmonic map theory. The interested reader should refer to [3,4,14,19] for further details.

Let us first define Harmonic maps [32]. Let P and L be Riemannian manifolds with boundaries ∂P and ∂L and local coordinates $\mathbf{x} = (x^1, x^2, x^3)$ and $\xi = (\xi^1, \xi^2, \xi^3)$ respectively [for clarity, one could consider P as a region in physical space, and L as a region in logical (or natural) space]. Let p^{ij} be the contravariant metric tensor in the coordinates x, and l_{mn} the covariant metric tensor in the coordinates ξ. The energy density for a map $\mathbf{x} : L \mapsto P$ is defined as:

$$e(\xi, \mathbf{x}) = \frac{1}{2} p^{ij}(\mathbf{x}) l_{\alpha\beta}(\xi) \frac{\partial \xi^\alpha}{\partial x^i} \frac{\partial \xi^\beta}{\partial x^j}, \tag{4}$$

where summation is intended on repeated indices unless otherwise specified. A total energy $E(\xi)$ is defined from the energy density as:

$$E(\xi) = \frac{1}{2} \int_P \mathrm{d}\mathbf{x}\, \sqrt{p} e(\xi, \mathbf{x}) = \frac{1}{2} \int_P \mathrm{d}\mathbf{x}\, \sqrt{p} p^{ij}(\mathbf{x}) l_{\alpha\beta}(\xi) \frac{\partial \xi^\alpha}{\partial x^i} \frac{\partial \xi^\beta}{\partial x^j}, \tag{5}$$

where $p = \det\{p_{ij}\} = 1/\det\{p^{ij}\}$. The map $\mathbf{x}$ is said to be *Harmonic* if it is of class C^2, and is a critical point of E [32,3]. The corresponding Euler–Lagrange equation that defines the extremal of the energy (and thus the Harmonic map) is

$$\nabla \cdot \left(\sqrt{p} \overleftrightarrow{p} \cdot \nabla \xi^\delta \right) + \Gamma^\delta_{\alpha\beta} \nabla \xi^\alpha \cdot (\sqrt{p} \overleftrightarrow{p}) \cdot \nabla \xi^\beta = 0, \tag{6}$$

where $\Gamma^\delta_{\alpha\beta}$ is the Christoffel symbol of the second kind, given by

$$\Gamma^\delta_{\alpha\beta} = \frac{1}{2} l^{\delta n} \left[\frac{\partial l_{n\alpha}}{\partial x^\beta} + \frac{\partial l_{n\beta}}{\partial x^\alpha} - \frac{\partial l_{\alpha\beta}}{\partial x^n} \right].$$

Equation (6) is of the Laplace–Beltrami type.

Existence and uniqueness of Harmonic maps are established by the following theorems [33,34]:

Hamilton–Schoen–Yau (HSY) Theorem. *Let P with metric p_{ij} and L with metric l_{ij} be two Riemannian manifolds with boundaries ∂P and ∂L, and let $\phi : P \mapsto L$ be a diffeomorphism. For any map $f : P \mapsto L$ such that $f|_{\partial\ P} = \phi|_{\partial P}$, we define $E(f) = \int_P \|\mathrm{d}f\|^2 \mathrm{d}\mathbf{x}$. The mapping f is Harmonic if it is an extremal of E.*

Theorem. *If the curvature of L is non-positive, and ∂L is convex (with respect to the metric $l_{\alpha\beta}$) then there exists a unique Harmonic map $f : P \mapsto L$ such that f is an homotopy equivalent to ϕ. In other words, one can deform f to ϕ by constructing a continuous family of maps $g_t : P \mapsto L$, $t \in\ [0, 1]$, such that $g_0(\mathbf{x}) = \phi(\mathbf{x})$, $g_1(\mathbf{x}) = f(\mathbf{x})$, and $g_t(\mathbf{x}) = \phi(\mathbf{x})$, $\forall \mathbf{x} \in \partial P$, $t \in [0, 1]$.*

In particular, a solution is guaranteed if one chooses a "flat" Euclidean space for the logical space. In this case, $\Gamma^\delta_{\alpha\beta} = 0$, and the Beltrami grid-generator equation reads:

$$\nabla \cdot \left(\sqrt{p} \overleftrightarrow{p} \cdot \nabla \xi^\delta \right) = 0; \quad \delta = 1, \ldots, n. \tag{7}$$

This equation provides a solution for the mapping $\xi(\mathbf{x})$. However, in most applications of interest, the mapping $\mathbf{x}(\xi)$ is of interest. This requires inverting Eq. (7), and is the subject of the discussion in the next section.

The fundamental interest of Harmonic functional theory in regards to grid generation is that, as Dvinsky points out [3], there is freedom in the selection of the physical space metric tensor, p^{ij}. The only conditions to be required of p^{ij} is that it be symmetric positive definite. In fact, one can choose suitable forms of such tensor to adapt to particular features in the domain [3], to adapt to a given scalar function (e.g., a measure of the numerical error) [4], or to align the grid with a given vector field [4]. In the numerical results section (Section 6), we will demonstrate the performance of the solver for both adaptation and alignment.

3.1. Inverse Beltrami equation (IBE)

In reviewing the literature, one finds as many equivalent formulations of the inverse Beltrami equation – to solve for $\mathbf{x}(\xi)$ – as practitioners. However, some of them are remarkably complex and ill-conditioned from the numerical standpoint (e.g., see [3,4,14]). Here, we proceed to derive a very simple form of the inverse Beltrami equation that numerical experiments will prove very convenient from the standpoint of multigrid methods.

We first realize that Eq. (7) is just the divergence of a vector in physical space. It is well-known that such divergence can always be written in logical space as [14]:

$$\nabla \cdot \mathbf{A} = \frac{1}{J}\frac{\partial(JA^i)}{\partial \xi^i},$$

where $A^i = \mathbf{A}\cdot\nabla\xi^i$ and J is the Jacobian factor of the transformation $\mathbf{x}(\xi)$. Using this formula with $\mathbf{A} = \sqrt{p}\overleftrightarrow{p}\cdot\nabla\xi^\delta$, we find trivially

$$\nabla\cdot\left(\sqrt{p}\overleftrightarrow{p}\cdot\nabla\xi^\delta\right) = \frac{1}{J}\frac{\partial}{\partial\xi^i}\left(J\sqrt{p}\nabla\xi^i\cdot\overleftrightarrow{p}\cdot\nabla\xi^\delta\right).$$

Defining $\hat{p}^{i\delta} = J\nabla\xi^i\cdot(\sqrt{p}\overleftrightarrow{p})\cdot\nabla\xi^\delta$, we finally arrive at the compact inverse equation

$$\frac{\partial\hat{p}^{i\delta}}{\partial\xi^i} = 0, \quad \delta = 1,\ldots,n. \tag{8}$$

Similarly compact formulations can be found in [19,9,10]. To finalize the derivation of the IBE, we note that the normal vectors $\nabla\xi^k$ can be found from the tangential vectors $\partial\mathbf{x}/\partial\xi^i$ as [14]:

$$\nabla\xi^k = \frac{\epsilon_{ijk}}{2J}\frac{\partial\mathbf{x}}{\partial\xi^i}\times\frac{\partial\mathbf{x}}{\partial\xi^j}, \tag{9}$$

where ϵ_{ijk} is the Levi–Civita antisymmetric tensor. If sources of the form described in Eq. (6) are to be considered, the IBE generalizes straightforwardly as

$$\frac{\partial\hat{p}^{i\delta}}{\partial\xi^i} + \Gamma^\delta_{\alpha\beta}\hat{p}^{\alpha\beta} = 0, \quad \delta = 1,\ldots,n.$$

In order to complete the derivation of the IBE, we need to specify suitable boundary conditions. Following [4], the natural boundary condition on a surface bounded by ξ^i that is compatible with the minimization of the grid energy variational principle (Eq. (5)) is

$$\hat{p}^{ij} = J\sqrt{p}\nabla\xi^i\cdot\overleftrightarrow{p}\cdot\nabla\xi^j = 0, \quad i \neq j. \tag{10}$$

Obtaining an inverse formulation of this boundary condition is straightforward using Eq. (9). In the case where $\overleftrightarrow{p}$ is diagonal (as in pure adaptivity applications), the natural boundary condition simplifies to $g^{ij} = \nabla\xi^i\cdot\nabla\xi^j = 0$, $i \neq j$, where g^{ij} is the contravariant metric tensor between the logical and the physical domains. In this case, the inversion is trivial, since $g^{ij} = 0$ implies $g_{ij} = \frac{\partial\mathbf{x}}{\partial\xi^i}\cdot\frac{\partial\mathbf{x}}{\partial\xi^j} = 0,\ i \neq j$, because the latter is the inverse of the former (i.e., if off-diagonal terms in g^{ij} are zero, then so are those of g_{ij}). At this point, the formulation of the IBE is complete. We now proceed to discuss its discrete representation.

4. Spatial discretization of IBE

The conservative form of the IBE proposed in Eq. (8) is remarkably compact and concise. However, the interest is not only academic, since the conservative formulation lends itself to a compact discretization in multi-dimensions that features the right properties for a classical multigrid treatment, namely, the existence of off-the-shelf smoothing techniques based on point Jacobi or Gauss–Seidel iterative methods. This is demonstrated by numerical experiment later in this section.

In what follows, we specialize the discussion to two dimensions, with $\mathbf{x} = [x(\xi^1,\xi^2), y(\xi^1,\xi^2)]$. Taking equation $\delta = 1$ in Eq. (8) as an example, the IBE reads

$$\frac{\partial\hat{p}^{11}}{\partial\xi^1}+\frac{\partial\hat{p}^{21}}{\partial\xi^2}=0,$$

where $\hat{p}^{ij}=J\nabla\xi^i\cdot\overset{\leftrightarrow}{P}\cdot\nabla\xi^j$, $\overset{\leftrightarrow}{P}(\mathbf{x})=\sqrt{p}\,\overset{\leftrightarrow}{p}$, $J=(\frac{\partial x}{\partial\xi^1}\frac{\partial y}{\partial\xi^2}-\frac{\partial x}{\partial\xi^2}\frac{\partial y}{\partial\xi^1})$, and, following Eq. (9):

$$\nabla\xi^1=\frac{1}{J}\left(\frac{\partial y}{\partial\xi^2},-\frac{\partial x}{\partial\xi^2}\right);\quad\nabla\xi^2=\frac{1}{J}\left(-\frac{\partial y}{\partial\xi^1},\frac{\partial x}{\partial\xi^1}\right).$$

Therefore

$$\hat{p}^{11}=-\frac{1}{J}\left[P^{11}\left(\frac{\partial y}{\partial\xi^2}\right)^2-2P^{12}\frac{\partial x}{\partial\xi^2}\frac{\partial y}{\partial\xi^2}+P^{22}\left(\frac{\partial x}{\partial\xi^2}\right)^2\right],$$

$$\hat{p}^{12}=-\frac{1}{J}\left[P^{11}\frac{\partial y}{\partial\xi^2}\frac{\partial y}{\partial\xi^1}-P^{12}\left(\frac{\partial x}{\partial\xi^1}\frac{\partial y}{\partial\xi^2}+\frac{\partial x}{\partial\xi^2}\frac{\partial y}{\partial\xi^1}\right)+P^{22}\frac{\partial x}{\partial\xi^2}\frac{\partial x}{\partial\xi^1}\right],$$

$$\hat{p}^{22}=-\frac{1}{J}\left[P^{11}\left(\frac{\partial y}{\partial\xi^1}\right)^2-2P^{12}\frac{\partial x}{\partial\xi^1}\frac{\partial y}{\partial\xi^2}+P^{22}\left(\frac{\partial x}{\partial\xi^1}\right)^2\right].$$

A finite volume discretization of the IBE reads

$$\frac{\hat{p}^{11}_{i+1/2,j}-\hat{p}^{11}_{i-1/2,j}}{\Delta\xi^1}+\frac{\hat{p}^{12}_{i,j+1/2}-\hat{p}^{12}_{i,j-1/2}}{\Delta\xi^2}=0,$$

where, as an example

$$\hat{p}^{11}_{i+1/2,j}=-\frac{1}{J_{i+1/2,j}}\left[P^{11}\left(\frac{\partial y}{\partial\xi^2}\right)^2-2P^{12}\frac{\partial x}{\partial\xi^2}\frac{\partial y}{\partial\xi^2}+P^{22}\left(\frac{\partial x}{\partial\xi^2}\right)^2\right]_{i+1/2,j}$$

and so on. The final step is to discretize terms of the form $\frac{\partial}{\partial\xi^1}\Big|_{i+1/2,j}$ and $\frac{\partial}{\partial\xi^2}\Big|_{i+1/2,j}$, and similarly at the face $j+1/2$. This is done in the usual finite-volume manner, which we exemplify below:

$$\left.\frac{\partial x}{\partial\xi^1}\right|_{i+1/2,j}=\frac{x_{i+1,j}-x_{i,j}}{\Delta\xi^1},$$

$$\left.\frac{\partial x}{\partial\xi^2}\right|_{i+1/2,j}=\frac{1}{2}\left[\frac{x_{i+1,j+1}-x_{i+1,j-1}}{2\Delta\xi^2}+\frac{x_{i,j+1}-x_{i,j-1}}{2\Delta\xi^2}\right].$$

The discretization is now complete, and results in a compact 9-point stencil. Such compact stencil support is essential to ensure good numerical behavior of the ensuing algebraic equations. In particular, it results in a diagonally dominant formulation, which is advantageous in a multigrid setting due to the effectiveness of off-the-shelf smoothers such as weighed Jacobi or Gauss–Seidel [35].

We demonstrate that this is the case for the proposed discretization of the IBE by finding the spectral radius of the *damped* Jacobi iteration matrix $S_k(\omega)=(I-\omega D_k^{-1}J_k)$, where J_k is the Jacobian resulting from the discretization above (for the double-vortex grid-alignment problem in Section 6.2), D_k is its block-diagonal (with 2×2 blocks corresponding to the two coupled grid equations in 2D), and ω is the damping parameter. The undamped iteration matrix $S_k(\omega=1)$ governs the rate of decay of the error in the solution of the system $J_k\mathbf{x}=\mathbf{b}$ when a stationary Jacobi iteration of the form $D_k\mathbf{x}^{m+1}=(D_k-J_k)\mathbf{x}^m+\mathbf{b}$ is employed [35]. If $\rho[S_k(\omega=1)]<1$, the Jacobi iteration is convergent; furthermore, the iterative method can be turned into a proper smoother by choosing $\omega<1$ [35]. Fig. 1 depicts the eigenvalue spectrum $|\sigma|$ of $S_0(\omega)$ for several $\omega\leqslant1$, demonstrating that $\rho[S_0(\omega)]<1$, and hence implying that the discretization above has the smoothing property.

The implementation details of a multigrid preconditioner that exploits such smoothing property is the subject of the next section.

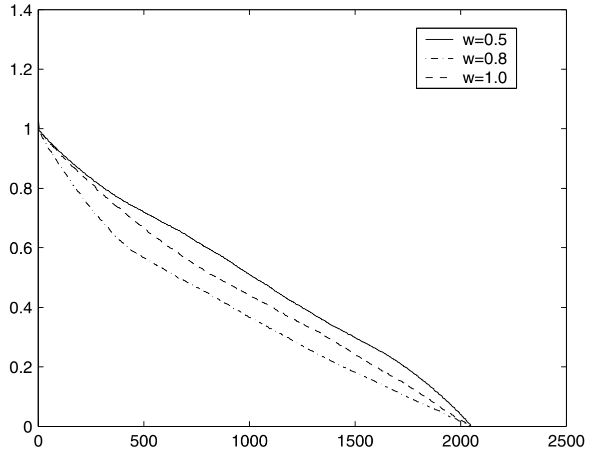

Fig. 1. Plot of the eigenvalues of the stationary iteration matrix $S_0(\omega) = (I - \omega D_0^{-1} J_0)$ corresponding to the initial Newton iterate $k = 0$ for the double-vortex grid-alignment problem (Section 6.2).

5. Preconditioning strategy

While multigrid methods are well-known for being very sensitive to the nature of the problem at hand (mostly due to the difficulty in finding adequate smoothers), they have been shown in many instances [29,36–39] to be ideally suited for the preconditioner stage of a Krylov method. In fact, fairly crude multigrid implementations have succeeded in delivering optimal Krylov convergence rates under grid refinement [29,36–39]. The fundamental aspect of a successful multigrid method is the existence of an effective smoother. In many applications, this is a statement of diagonal dominance, i.e., if the discrete formulation results in a diagonal-dominant matrix, simple stationary iterative techniques such as weighed Jacobi or Gauss–Seidel will be successful as smoothers [35]. It is important to note that our concern here is to capture enough of the MG technology to obtain a successful preconditioner, with success measured in terms of the scalability of the outer Krylov iteration. Accordingly, we are not bound by traditional MG theory, and suitable approximations that simplify the MG implementation (such as using low-order restriction and prolongation interpolations) are allowed [29,36–39].

The preconditioner step aims to cheaply approximate the inverse of the Jacobian matrix. This implies that the IBE equation above [Eq. (8)] must be linearized before MG is to be applied. Given the description of the discretization in Section 4, it is clear that this is no simple task. However, one can trivially do this using the Gateaux derivative in Eq. (2) every time the MG algorithm requires forming the residual. We term this a matrix-light MG implementation, because only the diagonal (or block diagonal in system of PDEs) of the Jacobian needs to be formed and stored, for smoothing purposes. Such diagonal is found in a Jacobian-free fashion using the Gateaux derivative, taking advantage of efficient colored-grid methods [40,41]. In structured grids, these methods exploit the node decoupling in discrete stencils characteristic of finite-volume/finite-difference schemes. Decoupled nodes are assigned the same color (e.g., red-black coloring in a 5-point stencil discretization of the Laplacian operator). Matrix elements corresponding to same-color nodes can be found in a single matrix–vector product. In this manner, only as many matrix–vector products as colors in a grid are needed to form the full diagonal. In 2D, four colors are needed to decouple a 9-point stencil, which in turn implies that only four Jacobian–vector products are required to form the complete diagonal. This approach has the fundamental advantages of reusing the nonlinear function evaluation, and avoiding its explicit linearization (which in turn allows one to reuse the algorithm for arbitrary adaptivity tensors $\overleftrightarrow{p}$, regardless of their complexity). The coarse operators are formed by re-discretization, instead of using a Galerkin procedure. This allows further reuse of the nonlinear function evaluation for all levels in the MG algorithm. It should be noted that the IBE system of equations in Eq. (8), with the boundary conditions as expressed in Eq. (10), is tightly coupled. Therefore, one necessarily needs to solve them coupled, which implies that a system MG solver is required.

In our MG implementation, we employ a single $V(5,5)$ cycle, smoothed with weighed block Jacobi with a weight factor $\omega = 0.8$. Piece-wise constant interpolation is used in the restriction and prolongation steps [36–38]. The coarsest grid level is a 16×16 grid. The smoother is also employed as the coarse grid solver.

A fundamental aspect of the MG implementation is the coarsening of the adaptivity tensor, $p^{ij}(\mathbf{x})$. This is particularly relevant in instances where p^{ij} is localized in space, which is common when p^{ij} contains an estimate of numerical truncation error [42–44], or is specified for the grid to adapt to a localized feature [3]. In such cases, a simple evaluation of p^{ij} at the coarse locations is not sufficient, as the coarse grids may miss the feature entirely. This is dealt with here as follows. In coarser grids, we define an average adaptivity tensor $\langle p^{ij} \rangle$, in a finite volume sense, as follows:

$$\langle p^{ij} \rangle_{\Omega_c} = \frac{1}{V_c} \int_{\Omega_c} \mathrm{d}V \, p^{ij}(\mathbf{x}) = \frac{1}{V_c} \int_{\Omega_c} \mathrm{d}\xi_1 \, \mathrm{d}\xi_2 \, J[\mathbf{x}(\xi_1, \xi_2)] p^{ij}[\mathbf{x}(\xi_1, \xi_2)],$$

where Ω_c is a given coarse logical control volume, and $V_c = \int_{\Omega_c} J \, \mathrm{d}\xi_1 \, \mathrm{d}\xi_2$ is its physical volume. While this averaging procedure tends to decrease the magnitude of the tensor as the grid is coarsened, it successfully transfers information of localized features in the fine grid to coarser meshes, thereby producing an effective MG algorithm. Numerical proof is provided in Section 6.

The integral is performed numerically by first finding $\mathbf{x}(\xi_1, \xi_2)$ at the vertices of the logical control volume using the current fine-grid solution of the mapping, and then reconstructing the mapping within such volume by a bilinear interpolation of $\mathbf{x}(\xi_1, \xi_2)$. We call such reconstructed solution $\hat{\mathbf{x}}(\xi_1, \xi_2)$. Finally, the integral in Ω_c is discretized as

$$\int_{\Omega_c} \mathrm{d}\xi_1 \, \mathrm{d}\xi_2 \, J p^{ij}[\mathbf{x}(\xi_1, \xi_2)] \approx \sum_{n,m=1}^{n_p} \delta\xi_1 \delta\xi_2 [J p^{ij}]_{\hat{\mathbf{x}}_{mn}},$$

where $\hat{\mathbf{x}}_{mn} = \hat{\mathbf{x}}(\xi_{1m}, \xi_{2n})$, $\xi_{i\ l} = \xi_{i\ min} + (l-1)\delta\xi_i$, $\delta\xi_i = \Delta_c \xi_i / n_p$, $\Delta_c \xi_i$ is the coarse grid logical spacing in the i-direction, and n_p is the number of points to consider for the integral inside Ω_c. Typically, we let n_p increase as the grid gets coarser, to capture local features of p^{ij} in coarse grids: $n_p = 1$ for the finest grid (i.e., a single-point evaluation), and $n_p = n_p + 1$ per coarsening step.

Such average adaptivity tensor $\langle p^{ij} \rangle$ is employed instead of the original p^{ij} in the discrete equations for the evaluation of residuals in coarse grid levels. The performance of the MG preconditioner is discussed in the following section.

6. Numerical results

We proceed to demonstrate the algorithm in two distinct applications: grid adaptivity and grid alignment. The former demonstrates the effectiveness of the algorithm in solving the IBE for adaptation to a given scalar function, while the latter demonstrates the effectiveness in aligning the grid with a given vector field.

We choose rectangular domains for both the logical and the physical domains. This is not a fundamental limitation, since, as stated in Section 3, Harmonic grid generation can be applied to a suitable parametrization of the physical domain, and such parametric space is rectangular in many cases of interest (e.g., (r, θ) for polar domains, (r, θ, Φ) for spherical domains, etc.). For the examples below, the initial guess for the nonlinear iteration is the identity mapping (i.e., a Cartesian grid).

6.1. Grid adaptivity

Grid adaptation to a scalar monitor function $w(\mathbf{x}, t)$ is achieved by setting $p^{ij} = \delta^{ij}/w(\mathbf{x},t)$, where δ^{ij} is the Kronecker delta. In this special case, the Harmonic function theory results in Winslow's variable diffusion method [45], and the IBE reads

$$\frac{\partial}{\partial \xi_i}\left(\frac{Jg^{i\delta}}{w}\right) = 0, \quad \delta = 1,2. \tag{11}$$

The monitor function may be used to adapt to a feature [3], to numerical error [42–44], or to generate a geometry [4]. Equation (11) will concentrate the grid in regions, where w is large, and rarefy the grid in regions, where w is small [4].

Equation (11) is an equidistribution principle. Assuming $g^{ij} \ll g^{ii} \sim \mathcal{O}(1)$, then it follows that the IBE will preserve the ratio w/J approximately constant. This, in turn, implies that the ratio of the maximum to the minimum Jacobian (cell volume) values, $J_{\max}/J_{\min}$ is approximately determined by $w_{\max}/w_{\min}$. This consideration is important in order to construct well-behaved scalar monitor functions.

The MG-preconditioned Newton–Krylov solver is tested with a Gaussian annulus $w(r) = 1 + b \exp[-|r^2 - r_0^2|/\sigma^2]$, with $r_0 = 0.4$, $\sigma = 0.1$, and r the cylindrical radius, with origin at the center of the Cartesian domain $0 \leqslant x, y \leqslant 1$. This monitor function mimics a sharp front. By the argument in the previous paragraph, it is clear that the coefficient b approximately determines the maximum to minimum volume (Jacobian) ratio. Here, $b = 9$ is chosen to limit the Jacobian ratio to $J_{\max}/J_{\min} \approx 10$.

The scalability of the solver under grid refinement is demonstrated in Table 1, where it is shown that the CPU time scales linearly (optimally) with the number of mesh points N (i.e., by a factor of 4 per grid refinement), and that both the number of nonlinear and linear (GMRES) iterations scale as $O(N^0)$. The number of GMRES iterations reported is the total number, not the number per nonlinear iteration. The number of linear iterations is smaller than that of nonlinear iterations owing to the fact that we employ our MG preconditioner to provide an initial guess for GMRES solve, and often the initial guess is good enough to satisfy the inexact Newton linear solve convergence tolerance, ζ_k [Eq. (3)]. In fact, this suggests that one could use the MG preconditioner as a solver in each Newton step, without mediation of GMRES. However, we keep the GMRES step in anticipation of using the grid generation strategy outlined here in a moving-grid context, where the grid equation will be coupled to a physical model of interest. A sample 128×128 grid obtained for the Gaussian annulus is shown in Fig. 2(a), where the front is clearly visible. The Jacobian J of the mapping $\mathbf{x}(\xi)$ is depicted in logical space in Fig. 2(b); its maximum and minimum values indicate that $J_{\max}/J_{\min} \approx 10$, as expected.

Table 1
Grid convergence study of Eq. (11) for the Gaussian annulus error function

Grid	Newton iterations	Total GMRES iterations	CPU(s)
32 × 32	9	1	8
64 × 64	10	1	35
128 × 128	10	2	150

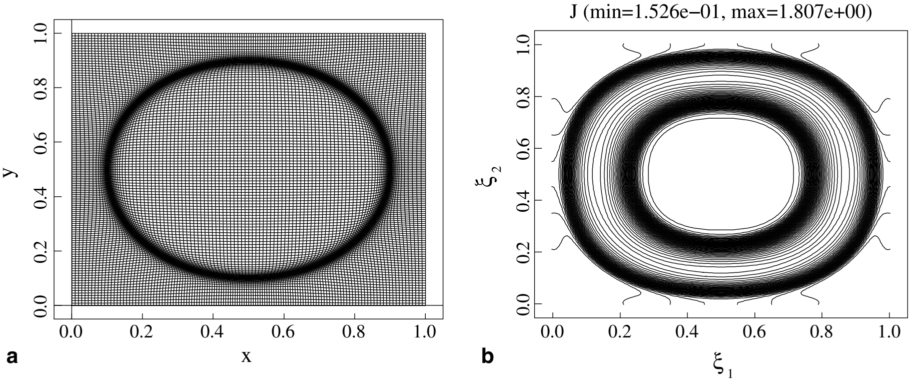

Fig. 2. Two-dimensional 128 × 128 grid obtained with Winslow's variable diffusion method for the Gaussian annulus error function: (a) actual grid and (b) Jacobian J of the mapping.

6.2. Grid alignment

Grid alignment to a vector field $\mathbf{A}$ is of interest when minimizing the numerical error in directional derivative terms of the form $\mathbf{A}\cdot\nabla$ is of the essence. This is the case in strongly anisotropic problems, such as flow-dominated regimes [7] (where advection along the velocity field is the dominant transport mechanism) and strongly magnetized plasmas [4] (where energy transport occurs preferentially along the magnetic field $\mathbf{B}$ [46]). In such scenarios, the numerical treatment of the directional derivatives is delicate, since even small numerical truncation errors in the parallel (to the vector field) derivative pollutes the perpendicular dynamics.

Grid alignment can help in such problems by transforming an otherwise multidimensional representation into an essentially one-dimensional one, where the discretization along the parallel direction can be performed accurately. Previous authors have used various grid generation algorithms to achieve this goal [2,7], including Harmonic maps [4,10]. In what follows, we propose a well-behaved formulation of the adaptivity tensor suitable for alignment, and demonstrate the good performance of the fully implicit algorithm with a challenging vector field.

6.2.1. Formulation of the alignment tensor

In the logical representation, $\mathbf{A}\cdot\nabla = A^i\partial/\partial\xi^i$, with $A^i = \mathbf{A}\cdot\nabla\xi^i$ the contravariant component of $\mathbf{A}$. Assume that ξ^1 is the coordinate chosen for alignment. Then, one-dimensionalizing the problem in 2D is equivalent to maximizing A^1 (which is in turn equivalent to minimizing $\mathbf{A}\times\nabla\xi^1$) while minimizing A^2. This suggests the following grid energy density (with the integral energy principle to be formed according to Eq. (5)):

$$e(\xi,\mathbf{x}) = \tfrac{1}{2}[(\mathbf{A}\times\nabla\xi^1)^2 + (\mathbf{A}\cdot\nabla\xi^2)^2]. \tag{12}$$

This is equivalent to that proposed by Brackbill in [4]:

$$e(\xi,\mathbf{x}) = \tfrac{1}{2}[(\mathbf{A}\times\nabla\xi^1)^2 + (\mathbf{B}\times\nabla\xi^2)^2],$$

when noting that, in [4], $\mathbf{A} = \hat{\mathbf{z}}\times\nabla\psi$ and $\mathbf{B} = \nabla\psi$, and therefore $(\mathbf{B}\times\nabla\xi^2)^2 = (\mathbf{A}\cdot\nabla\xi^2)^2$. When we cast Eq. (12) in the form of Eq. (4), we find:

$$\overleftrightarrow{p}_\perp = A^2\overleftrightarrow{I} - \mathbf{AA}, \quad \overleftrightarrow{p}_\parallel = \mathbf{AA},$$

where $\overleftrightarrow{p}_\perp$ corresponds to the ξ^1 equation, and $\overleftrightarrow{p}_\parallel$ to the ξ^2 equation. Note that $\overleftrightarrow{p}_\perp + \overleftrightarrow{p}_\parallel = A^2\overleftrightarrow{I}$, where $A = |\mathbf{A}|$.

However, the previous approach has a fundamental limitation, namely, that one has to choose the alignment coordinate a priori. This is not advantageous when the topology of the vector field $\mathbf{A}$ is complex, since the convenience of a given coordinate for alignment is region-dependent (and regularity constraints on the grid result in "transition" regions where no coordinate is perfectly aligned). It is best to follow [10], where it is shown that the eigenvectors of $\overleftrightarrow{p}$ can be used to determine the alignment directions. Accordingly, we pose a grid energy density that attempts to minimize all contravariant components of $\mathbf{A}$, in the following fashion:

$$e(\xi) = \tfrac{1}{2}[(\mathbf{A}\cdot\nabla\xi^1)^2 + (\mathbf{A}\cdot\nabla\xi^2)^2]. \tag{13}$$

The resulting alignment metric $\overleftrightarrow{p} = \mathbf{AA}$ has the alignment vector $\mathbf{A}$ and its perpendicular vector $\mathbf{A}_\perp$ as eigenvectors, but it does not constrain which component A^i to minimize (the algorithm will select this in a manner compatible with the local vector topology and the boundary conditions). However, as it stands, the adaptivity metric $\overleftrightarrow{p} = \mathbf{AA}$ is unsuitable for two reasons. Firstly, the tensor $\mathbf{AA}$ is singular everywhere (its determinant is zero). And secondly, in complex topologies, $\mathbf{A}$ will have null points (where $A = 0$), in the neighborhood of which A changes direction sharply. The latter is also a robustness issue, as it will most likely cause convergence difficulties as the solver tries to accommodate the topological complexity around null points.

These two issues can be resolved by considering the following regularized adaptivity tensor:

$$\overleftrightarrow{p} = \alpha\overleftrightarrow{I} + (1-\alpha)\left(\frac{A}{A_0}\right)^2[(1-\epsilon)\mathbf{aa} + \epsilon(I - \mathbf{aa})], \tag{14}$$

where $\mathbf{a} = \mathbf{A}/A$, A_0 is a normalization constant, $\alpha \in [0,1]$ regularizes the metric at null points, and $\epsilon \in [0,1]$ determines the relative contribution of $\overleftrightarrow{p}_\perp$ and $\overleftrightarrow{p}_\parallel$.The eigenvalues of $\overleftrightarrow{p}$ are $\lambda_\perp = \alpha + (1-\alpha)\epsilon\left(\frac{A}{A_0}\right)^2 = \alpha\left[1+\left(\frac{A}{A^*}\right)^2\right]$ with eigenvector $A_\perp$, and $\lambda_\parallel = \alpha + (1-\alpha)(1-\epsilon)\left(\frac{A}{A_0}\right)^2 = \alpha\left[1+\frac{(1-\epsilon)}{\epsilon}\left(\frac{A}{A^*}\right)^2\right]$ with eigenvector $\mathbf{A}$. The parameter $A^* = A_0\sqrt{\frac{\alpha}{(1-\alpha)\epsilon}}$ provides a threshold for the magnitude of $\mathbf{A}$ that separates neighborhoods of null points from the rest of the domain.Note that, by construction, $\lambda_{\perp,\parallel} \to \alpha$ when $A \to 0$, which prevents problems around null points. It should be noted that, according to Eq.(5), Eq.(14) is affected by $\sqrt{p} = 1/\sqrt{\det[\overleftrightarrow{p}]}$ to form a proper Harmonic map. In 2D, this correction results in $\det[\sqrt{p}\overleftrightarrow{p}] = 1$, which in turn implies that the eigenvalues of $\overleftrightarrow{P} = \sqrt{p}\overleftrightarrow{p}$ are $\lambda_1 = \sqrt{\lambda_\parallel/\lambda_\perp}$ and $\lambda_2 = 1/\lambda_1$, where

$$\lambda_1 = \sqrt{\frac{1+\frac{(1-\epsilon)}{\epsilon}\left(\frac{A}{A^*}\right)^2}{1+\left(\frac{A}{A^*}\right)^2}}. \tag{15}$$

Note that α only affects the value of the threshold A^*, and that ϵ determines most of the properties of the grid (in practice, as discussed below, ϵ provides an "alignment limit," the equivalent of a refinement limit for alignment). We now consider two limits for this equation. In the limit $A = 0$ (null points), $\lambda_{1,2} = 1$, i.e., the identity tensor is recovered, and no alignment is enforced – as desired. In the opposite limit $A \gg A^*$, $\lambda_1 \to \sqrt{\frac{1-\epsilon}{\epsilon}}$. Following [10], the behavior of the grid in this limit and in the transition region (where $A \sim A^*$) depends on the rate of change of λ_1^2 along its principal direction, $\mathrm{d}\lambda_1^2/\mathrm{d}A$, which for constant ϵ reads

$$\frac{\mathrm{d}\lambda_1^2}{\mathrm{d}A} = \frac{2A}{A_0^2}\left(\frac{1-\alpha}{\alpha}\right)\frac{(1-2\epsilon)}{\left[1+\left(\frac{A}{A^*}\right)^2\right]^2}. \tag{16}$$

The behavior of the grid in transition regions (where $A \sim A^*$) will depend on the magnitude and sign of the derivative in Eq. (16). The magnitude is essentially determined by A_0 and α. The sign, however, depends on whether $\epsilon \lessgtr 1/2$ (recall that the identity is recovered for $\epsilon = 1/2$, which enforces no alignment), and will determine whether the grid will concentrate or rarefy in these regions [10] (concentration is expected for $\mathrm{d}\lambda_1/\mathrm{d}A < 0$, and rarefaction otherwise). The parameter ϵ also determines which alignment tensor, $\overleftrightarrow{p}_\parallel$ or $\overleftrightarrow{p}_\perp$, is weighed favorably in Eq. (14), and hence one should expect alignment with $\mathbf{A}$ for $\epsilon < 1/2$, and with $\mathbf{A}_\perp$ for $\epsilon > 1/2$. Significant alignment is only achieved for ϵ values well separated from the 1/2 threshold, as the identity tensor is recovered when $\epsilon = 1/2$. In practice, one should choose $\epsilon \sim 1/4$ or 3/4 to achieve some measure of alignment without sacrificing resolution. In this study, unless otherwise specified, we employ $\alpha = 0.05$, $A_0 = 0.5\max[|\mathbf{A}|]$, and $\epsilon = 0.22$.

If grid adaptation is desired in addition to alignment, Eq. (16) suggests that one could use arbitrary functions $\epsilon(A) \in [0,1]$ to achieve a desired grid adaptation pattern. Alternatively, one could envision a generalized adaptivity tensor:

$$\overleftrightarrow{p} = \alpha\overleftrightarrow{I} + (1-\alpha)f(A)[(1-\epsilon)\mathbf{aa} + \epsilon(I-\mathbf{aa})],$$

where $f(A \to 0) \to 0$ for robustness in the neighborhood of null points. Exploring this issue any further, however, is beyond the scope of the present study.

6.2.2. Alignment test case: double-vortex equilibrium

We test the alignment capabilities of the alignment tensor formulated above, as well as the performance of the nonlinear solver, with an ideal 2D MHD equilibrium, given by the poloidal flux function [47]:

$$\psi(r,\theta) = \begin{cases} \frac{2}{kJ_0(k)}J_1(kr)\cos(\theta), & r < 1, \\ \left[r - \frac{1}{r}\right]\cos(\theta), & r \geqslant 1, \end{cases}$$

with r, θ the cylindrical coordinates, J_0 and J_1 are Bessel functions of the first kind, and $J_1(k) = 0$. The domain considered is Cartesian, with $-2 < x, y < 2$. The magnetic field, which is the alignment vector field, is obtained from the poloidal flux as $\mathbf{B} = \mathbf{z} \times \nabla\psi$. The double vortex problem has been chosen for its topological complexity, since it features both X-points and O-points (as is evident in Fig. 3).

The solver performance on this problem is presented in Table 2. Overall, the algorithm behaves similarly to the adaptive grid case. There is some increase of the nonlinear iterations with mesh refinement due to the very nonlinear nature of the equations in the alignment case, but the number of linear iterations remains constant. The CPU scaling remains essentially optimal, with a multiplication factor of 4–5 per grid refinement. We should point out, however, that the behavior of the nonlinear solver is strongly dependent on the value of ϵ, and significant slowdown of the Newton convergence is observed for $\epsilon \lesssim 0.15$.

A sample 128 × 128 grid obtained with the double-vortex configuration is depicted in Fig. 4. Notice that the mesh is essentially uniform except in the neighborhood of the null point (O-point), were it is significantly rarefied (as expected for $\epsilon < 1/2$, since $\mathrm{d}\lambda_1^2/\mathrm{d}A > 0$). To measure the quality of the grid alignment, we define the alignment figure-of-merit η as:

$$\eta = \frac{(B^1)^2}{(B^1)^2 + (B^2)^2}.$$

This figure of merit quantitatively measures how one-dimensional $\mathbf{B} \cdot \nabla$ has become owing to the grid alignment: the problem is effectively one dimensional when $\eta \sim 1$ or $\eta \sim 0$. Fig. 5 shows the regions delimited by $\eta < 0.1$ and $\eta > 0.9$ before and after alignment. The transition region $0.1 < \eta < 0.9$ has substantially shrunk in the aligned case, demonstrating that the aligning algorithm successfully improves on a uniform grid.

The effect of the parameter ϵ on the solution is demonstrated in Fig. 6, where solutions for $\epsilon = 0.17$, 0.25, 0.45, 0.75 are plotted. Clearly, the grid aligns with $\mathbf{A}$ for $\epsilon < 1/2$, with $\mathbf{A}_\perp$ for $\epsilon > 1/2$, and no alignment occurs for $\epsilon \sim 1/2$. Notice the lack of resolution around the O-point regions for $\epsilon = 0.15$, and the grid concentration in the same regions for $\epsilon = 0.75$, as expected from the analysis in the previous section.

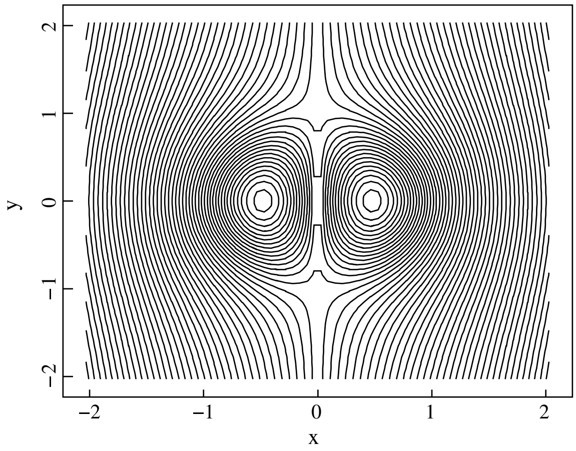

Fig. 3. Plot of the double-vortex poloidal flux function (see text).

Table 2
Grid convergence study of the alignment IBE for the double-vortex equilibrium

Grid	Newton iterations	Total GMRES iterations	CPU(s)
32 × 32	10	1	11
64 × 64	11	1	50
128 × 128	15	1	273

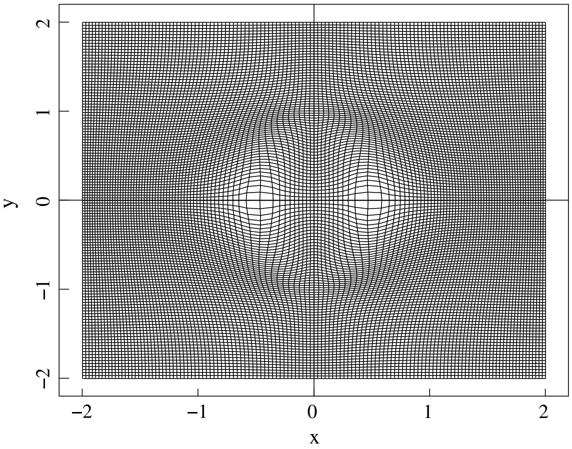

Fig. 4. Sample 128 × 128 grid obtained by aligning to the double-vortex field configuration.

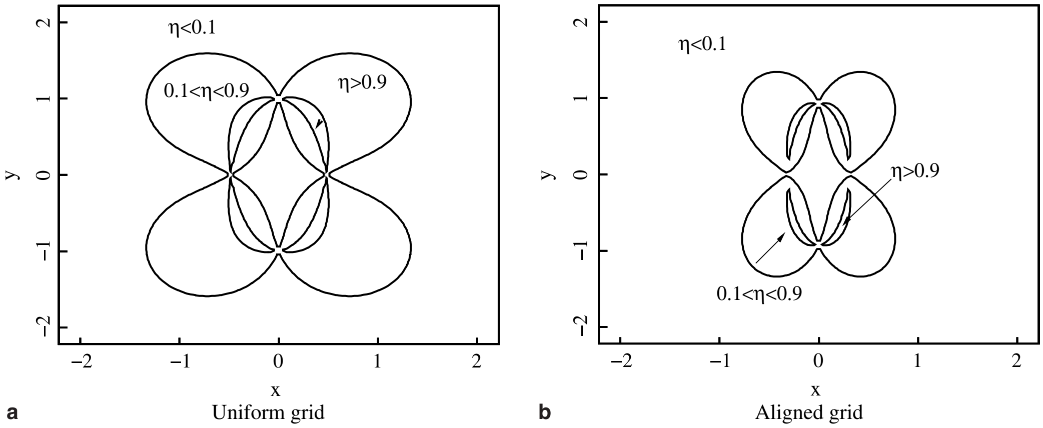

Fig. 5. Quantitative measure of the degree of one-dimensionality achieved by the grid alignment algorithm. The grid is essentially one-dimensional for regions $\eta < 0.1$ and $\eta > 0.9$. The aligned grid has succeeded in significantly reducing the area of the transition region $(0.1 < \eta < 0.9)$.

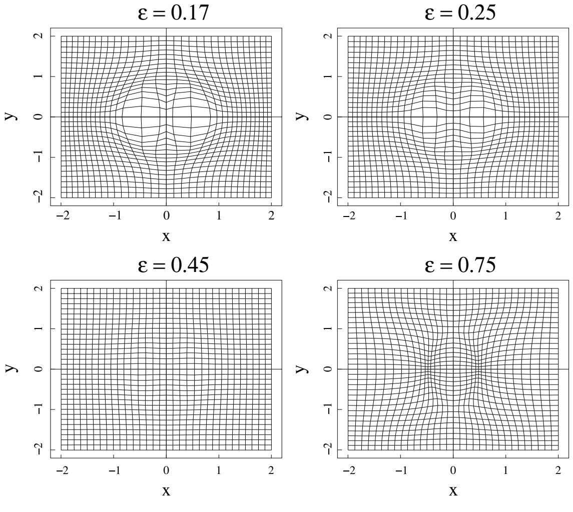

Fig. 6. Effect of various values of the parameter ϵ on the grid topology for 32 × 32 grids.

7. Conclusion

We have presented a fully nonlinear solver strategy for the extremely nonlinear grid generation equation obtained from Harmonic map theory. The advantage of Harmonic maps is that their existence and uniqueness is guaranteed under certain conditions, and that the final grid is parametrization-independent. The Beltrami grid generation equation, which satisfies the conditions of the HSY theorem, produces Harmonic maps, and therefore its solvability is guaranteed under certain conditions.

The fundamental contributions of this paper are as follows. Firstly, we have derived a well-behaved discrete inverse Beltrami equation (IBE) that is block-diagonally dominant and hence amenable to a multigrid treatment (i.e., it possesses the smoothing property). Secondly, we have based our nonlinear solver strategy on Newton–Krylov methods, using multigrid in the preconditioner stage. The multigrid preconditioner is so effective that very few Krylov iterations are needed. Furthermore, it produces bounded linear and nonlinear iteration count, with virtually no scaling under grid refinement, and produces CPU times that scale linearly with the number of unknowns. Thirdly, we have demonstrated the effectiveness of the solver with challenging adaptive and alignment examples. Such effectiveness opens the possibility of a fully nonlinear, coupled solution of moving grid strategies, where the physics and grid evolution equations are advanced simultaneously. And finally, we have described an effective, robust, grid alignment approach for arbitrary vector fields.

The demonstration of the algorithm proposed here has been done in a serial fashion. We do not envision any fundamental limitation for its parallelization, since Beltrami's equation is elliptic in nature (and therefore can be treated similarly to other elliptic problems as far as parallelization is concerned). Modern parallel toolkits such as PETSc [48] provide all the ingredients required, namely, parallel Newton–Krylov drivers and parallel multigrid capabilities.

Acknowledgments

The authors acknowledge useful conversations with J.U. Brackbill, A.H. Glasser, V. Liseikin, and V.S. Lukin. This work was funded by the Laboratory Directed Research and Development program (LDRD), under the auspices of the US Department of Energy under DOE contract W-7405-ENG-36 at Los Alamos National Laboratory, LA-UR-05-2302.

References

[1] J.U. Brackbill, J.S. Saltzman, Adaptive zoning for singular problems in 2 dimensions, J. Comput. Phys. 46 (3) (1982) 342–368.
[2] A.E. Giannakopoulos, A.J. Engel, Directional control in grid generation, J. Comput. Phys. 74 (1988) 422–439.
[3] A.S. Dvinsky, Adaptive grid generation from Harmonic maps, J. Comput. Phys. 95 (1991) 450–476.
[4] J.U. Brackbill, An adaptive grid with directional control, J. Comput. Phys. 108 (1993) 38–50.
[5] M.J. Baines, Moving Finite Elements, Oxford University Press, Oxford, 1994.
[6] W. Huang, Y. Ren, R.D. Russell, Moving mesh partial differential equations (MMPDES) based on the equidistribution principle, SIAM J. Numer. Anal. 31 (3) (1994) 709–730.
[7] P. Knupp, Mesh generation using vector fields, J. Comput. Phys. 119 (1995) 142–148.
[8] S. Li, L. Petzold, Moving mesh methods with upwinding schemes for time dependent PDEs, J. Comput. Phys. 131 (1997) 368–377.
[9] W. Cao, W. Huang, R.D. Russell, An *r*-adaptive finite element method based upon moving mesh PDEs, J. Comput. Phys. 149 (1999) 221–244.
[10] W. Cao, W. Huang, R.D. Russell, A study of monitor functions for two-dimensional adaptive grid generation, SIAM J. Sci. Comput. 20 (6) (1999) 1978–1994.
[11] R. Li, T. Tang, P. Zhang, Moving mesh methods in multiple dimensions based on Harmonic maps, J. Comput. Phys. 170 (2001) 562–588.
[12] W. Huang, Variational mesh adaptation: isotropy and equidistribution, J. Comput. Phys. 174 (2001) 903–924.
[13] H. Tang, Solution of the shallow water equations using an adaptive moving mesh method, Int. J. Numer. Meth. Fluids 44 (2004) 789–810.
[14] V.D. Liseikin, Grid Generation Methods, Springer, Berlin, New York, 1999.
[15] P. Knupp, S. Steinberg, Fundationals of Grid Generation, CRC Press, Boca Raton, FL, 1994.
[16] J.F. Thompson, Z.A. Warsi, C.W. Mastin, Numerical Grid Generation: Foundations and Applications, North-Holland, New York, 1985.
[17] J.E. Castillo (Ed.), Mathematics Aspects of Numerical Grid Generation, SIAM, Philadelphia, PA, 1991.

[18] D.A. Knoll, L. Chacón, L. Margolin, V.A. Mousseau, On balanced approximations for the time integration of multiple time scale systems, J. Comput. Phys. 185 (2003) 583–611.

[19] V.D. Liseikin, A Computational Differential Geometry Approach to Grid Generation, Springer, Berlin, New York, 2003.

[20] R.M. Spitaleri, V. Regolo, Multiblock multigrid grid generation algorithms: overcoming multigrid anisotropy, Appl. Math. Comput. 84 (1997) 246–267.

[21] R.M. Spitaleri, V. Micacchi, A multiblock multigrid generation method for complex simulations, Math. Comput. Simulat. 46 (1998) 1–12.

[22] R.M. Spitaleri, Full-Fas multigrid grid generation algorithms, Appl. Numer. Math. 32 (2000) 483–494.

[23] W.M. Tscharnuter, K.-H. Winkler, A method for computing selfgravitating gas flows with radiation, Comput. Phys. Commun. 18 (1979) 171–199.

[24] K.-H.A. Winkler, M.L. Normand, M.J. Newmann, Adaptive mesh techniques for fronts in star formation, Physics D 12D (1–3) (1984) 408–425.

[25] E.A. Dorfi, L.O. Drury, Simple adaptive grids for 1-D initial value problems, J. Comput. Phys. 69 (1987) 175–195.

[26] M. Gehmeyr, D. Mihalas, Adaptive grid radiation hydrodynamics with TITAN, Physics D 77 (1994) 320–341.

[27] C.T. Kelley, Iterative Methods for Linear and Nonlinear Equations, SIAM, Philadelphia, 1995.

[28] R. Dembo, S. Eisenstat, R. Steihaug, Inexact Newton methods, J. Numer. Anal. 19 (1982) 400.

[29] L. Chacón, D.A. Knoll, A 2D high-β Hall MHD implicit nonlinear solver, J. Comput. Phys. 188 (2) (2003) 573–592.

[30] G. Liao, A study of regularity problem of Harmonic maps, Pacific J. Math. 131 (1988) 291–302.

[31] G. Liao, N. Smale, Harmonic maps with nontrivial higher-dimensional singularities, Lecture Notes in Pure and Applied Mathematics, vol. 144, Dekker, New York, 1993, pp. 79–89.

[32] J. Eells, J.H. Sampson, Harmonic mappings of Riemannian manifolds, Am. J. Math. 86 (1) (1964) 109–160.

[33] R. Hamilton, Harmonic maps of manifolds with boundary, Lecture Notes in Mathematics, vol. 471, Springer, New York, 1975.

[34] R. Schoen, S.T. Yau, On univalent harmonic maps between surfaces, Invent. Math. 44 (1978) 265–278.

[35] P. Wesseling, An Introduction to Multigrid Methods, Wiley, Chichester, 1992.

[36] D.A. Knoll, W.J. Rider, A multigrid preconditioned Newton–Krylov method, SIAM J. Sci. Comput. 21 (2) (1999) 691–710.

[37] W.J. Rider, D.A. Knoll, G.L. Olson, A multigrid Newton–Krylov method for multimaterial equilibrium radiation diffusion, J. Comput. Phys. 152 (1) (1999) 164–191.

[38] D.A. Knoll, G. Lapenta, J.U. Brackbill, A multilevel iterative field solver for implicit, kinetic, plasma simulation, J. Comput. Phys. 149 (1999) 377–388.

[39] L. Chacón, D.A. Knoll, J.M. Finn, Implicit, nonlinear reduced resistive MHD nonlinear solver, J. Comput. Phys. 178 (1) (2002) 15–36.

[40] A.R. Curtis, M.J.D. Powell, J.K. Reid, On the estimation of sparse Jacobian matrices, J. Inst. Math. Appl. 13 (1974) 117–119.

[41] T.F. Coleman, J.J. Moré, Estimation of sparse Jacobian matrices and graph coloring problems, SIAM J. Numer. Anal. 20 (1983) 187–209.

[42] M. Ainsworth, J.T. Oden, A Posteriori Error Estimation in Finite Element Analysis, Wiley, New York, 2000.

[43] G. Lapenta, Variational grid adaptation based on the minimization of local truncation error: time independent problems, J. Comput. Phys. 193 (2004) 159.

[44] G. Lapenta, A recipe to detect the origin of error in discretization schemes, Int. J. Numer. Meth. Engng. 59 (2004) 2065.

[45] A. Winslow, Numerical solution of the quasi-linear poisson equation in a nonuniform triangle mesh, J. Comput. Phys. 1 (1967) 149.

[46] S.I. Braginskii, Transport processes in a plasma, in: M.A. Leontovich (Ed.), Reviews of Plasma Physics, vol. 1, Consultants Bureau, New York, 1965, pp. 205–311.

[47] H.R. Strauss, D.W. Longcope, An adaptive finite element method for magnetohydrodynamics, J. Comput. Phys. 147 (2) (1998) 318–336.

[48] S. Balay, K. Buschelman, V. Eijkhout, W.D. Gropp, D. Kaushik, M.G. Knepley, L.C. McInnes, B.F. Smith, H. Zhang, PETSc users manual, Technical Report ANL-95/11 – Revision 2.1.5, Argonne National Laboratory, 2004.