

Abstract Submitted
for the MAR99 Meeting of
The American Physical Society

Sorting Category: PC7 (Computation/~~Simulation~~)

RECEIVED
SEP 11 1999
O.S.T.

Multilevel Iterative Methods in Computational Plasma Physics¹ D.A. KNOLL, D.C. BARNES, J.U. BRACKBILL, L. CHACÓN, G. LAPENTA, Los Alamos National Laboratory — Plasma physics phenomena occur on a wide range of spatial scales and on a wide range of time scales. When attempting to model plasma physics problems numerically we are inevitably faced with the need for both fine spatial resolution (fine grids) and implicit time integration methods. Fine grids can tax the efficiency of iterative methods and large time steps can challenge the robustness of iterative methods. To meet these challenges we are developing a hybrid approach where multigrid methods are used as preconditioners to Krylov subspace based iterative methods, such as conjugate gradients or GMRES. For nonlinear problems we apply multigrid preconditioning to a matrix-free Newton-GMRES method [1]. Results are presented for application of these multilevel iterative methods to the field solves in implicit moment method PIC [2], multidimensional nonlinear Fokker-Planck problems [3], and our initial efforts in particle MHD.

- [1] D.A. Knoll and W.J. Rider, LA-UR-97-4013, to appear in SIAM J. Sci. Comput.
- [2] D.A. Knoll, G. Lapenta and J.U. Brackbill, LA-UR-98-2159, to appear in J. Comput. Phys.
- [3] D.A. Knoll, L. Chacon, D.C. Barnes, APS DPP meeting, Nov. 1998

¹Work supported by U.S. DoE under DOE contract W-7405-ENG-36 at LANL

- Prefer Oral Session
- Prefer Poster Session

Dana Knoll
nol@lanl.gov
LANL

DISCLAIMER

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, make any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

DISCLAIMER

**Portions of this document may be illegible
in electronic image products. Images are
produced from the best available original
document.**

1 Jacobian-Free Newton-Krylov Method

Newton's method requires the solution of the linear system

$$\mathbf{J}^k \delta \mathbf{u}^k = -\mathbf{F}(\mathbf{u}^k), \quad \mathbf{u}^{k+1} = \mathbf{u}^k + \delta \mathbf{u}^k; \quad (1)$$

where \mathbf{J} is the Jacobian matrix, $\mathbf{F}(\mathbf{u})$ is the nonlinear system of equations, \mathbf{u} is the state vector, and k is the nonlinear iteration index.

In vector notation (where i is the finite volume index), the $(i, j)^{th}$ element of the Jacobian matrix is,

$$J_{i,j} = \frac{\partial F_i(\mathbf{u})}{\partial u_j}. \quad (2)$$

Forming each element of \mathbf{J} requires taking analytic or numerical derivatives of $F_i(\mathbf{u})$ with respect to \mathbf{u} at each grid point. This can be both difficult and time consuming.

1.1 Matrix-free Approximation

The Generalized Minimal RESidual (GMRES) algorithm (Saad and Schultz, SIAM J. Sci. Comput. vol 7, 1986) is used to solve Eq. (1). GMRES (or any other Krylov method such as conjugate gradients) defines an initial linear residual, \mathbf{r}_0 given an initial guess, $\delta \mathbf{u}_0$,

$$\mathbf{r}_0 = -\mathbf{F}(\mathbf{u}) - \mathbf{J}\delta\mathbf{u}_0. \quad (3)$$

Note that the nonlinear iteration index, k , has been dropped.

The l^{th} GMRES iteration minimizes $\| \mathbf{J}\delta\mathbf{u}_l + \mathbf{F}(\mathbf{u}) \|_2$ with a least squares approach. $\delta\mathbf{u}_l$ is constructed from a linear combination of the Krylov vectors (search directions) $\{\mathbf{r}_0, \mathbf{J}\mathbf{r}_0, (\mathbf{J})^2\mathbf{r}_0, \dots, (\mathbf{J})^{l-1}\mathbf{r}_0\}$, which were constructed during the previous $l - 1$ GMRES iterations. This linear combination of Krylov vectors can be written as,

$$\delta\mathbf{u}_l = \delta\mathbf{u}_0 + \sum_{j=0}^{l-1} \alpha_j (\mathbf{J})^j \mathbf{r}_0, \quad (4)$$

where evaluating the scalars α_j is part of the GMRES iteration. Upon examining Eq.(4) we see that GMRES requires the action of the Jacobian only in the form of matrix-vector products, which can be approximated by (Brown and Saad, SIAM J. Sci. Comput. vol 11, 1990) ;

$$\mathbf{J}\mathbf{v} \approx [\mathbf{F}(\mathbf{u} + \epsilon\mathbf{v}) - \mathbf{F}(\mathbf{u})] / \epsilon, \quad (5)$$

where \mathbf{v} is a Krylov vector (i.e. one of $\{\mathbf{r}_0, \mathbf{J}\mathbf{r}_0, (\mathbf{J})^2\mathbf{r}_0, \dots, (\mathbf{J})^{l-1}\mathbf{r}_0\}$), and ϵ is a small perturbation.

Equation 5 is a first order Taylor series expansion approximation to the Jacobian, \mathbf{J} , times a vector, \mathbf{v} .

For illustration consider the two coupled nonlinear equations $F_1(u_1, u_2) = 0$, $F_2(u_1, u_2) = 0$. The Jacobian for this problem is;

$$\mathbf{J} = \begin{bmatrix} \frac{\partial F_1}{\partial u_1} & \frac{\partial F_1}{\partial u_2} \\ \frac{\partial F_2}{\partial u_1} & \frac{\partial F_2}{\partial u_2} \end{bmatrix}.$$

Working backwards from Eq.(5), we have;

$$\frac{\mathbf{F}(\mathbf{u} + \epsilon\mathbf{v}) - \mathbf{F}(\mathbf{u})}{\epsilon} = \begin{pmatrix} \frac{F_1(u_1 + \epsilon v_1, u_2 + \epsilon v_2) - F_1(u_1, u_2)}{\epsilon} \\ \frac{F_2(u_1 + \epsilon v_1, u_2 + \epsilon v_2) - F_2(u_1, u_2)}{\epsilon} \end{pmatrix}.$$

Approximating $\mathbf{F}(\mathbf{u} + \epsilon\mathbf{v})$ with a first order Taylor series expansion about \mathbf{u} , we have;

$$\frac{\mathbf{F}(\mathbf{u} + \epsilon\mathbf{v}) - \mathbf{F}(\mathbf{u})}{\epsilon} \approx \begin{pmatrix} \frac{F_1(u_1, u_2) + \epsilon v_1 \frac{\partial F_1}{\partial u_1} + \epsilon v_2 \frac{\partial F_1}{\partial u_2} - F_1(u_1, u_2)}{\epsilon} \\ \frac{F_2(u_1, u_2) + \epsilon v_1 \frac{\partial F_2}{\partial u_1} + \epsilon v_2 \frac{\partial F_2}{\partial u_2} - F_2(u_1, u_2)}{\epsilon} \end{pmatrix}.$$

This expression can be simplified to;

$$\begin{pmatrix} v_1 \frac{\partial F_1}{\partial u_1} + v_2 \frac{\partial F_1}{\partial u_2} \\ v_1 \frac{\partial F_2}{\partial u_1} + v_2 \frac{\partial F_2}{\partial u_2} \end{pmatrix} = \mathbf{J}\mathbf{v}$$

This matrix-free approach, besides its obvious memory advantage, has many unique capabilities. Namely, Newton-like nonlinear convergence without *forming* or *inverting* the true Jacobian.

To complete the description of this technique we provide a prescription for evaluating the scalar perturbation. In this study ϵ is given by,

$$\epsilon = \frac{1}{N\|\mathbf{v}\|_2} \sum_{m=1}^N b|u_m|, \quad (6)$$

where N is the linear system dimension and b is a constant whose magnitude is approximately the square root of machine roundoff ($b = 10^{-5}$ for most of this study).

Preconditioned GMRES

- Compare to CG, work and storage, as a function of iteration

CG: work scales linearly, storage is constant

GMRES: work scales quadratically, storage scales linearly

- Restart, GMRES(k) : (results in constant storage)

An often employed "fix" is to store only k , Krylov vectors. If linear convergence is not achieved after k iterations a new temporary linear solution is constructed from the existing k vectors and GMRES is restarted. This restarting can significantly effect linear convergence rate.

- Preconditioning: $(\mathbf{A}\mathbf{P}^{-1})(\mathbf{P}\phi) = \rho$

Each GMRES iterations requires a preconditioned "matvec"

$$\mathbf{w} = (\mathbf{A}\mathbf{P}^{-1})\mathbf{v} \quad (1)$$

Step 1, Preconditioning (iterative solve):

$$\mathbf{P}\mathbf{y} = \mathbf{v} \quad (2)$$

Step 2, "matvec":

$$\mathbf{w} = \mathbf{A}\mathbf{y} \quad (3)$$

2-Grid V-cycle

Iterative solution to $\mathbf{P}\mathbf{y} = \mathbf{v}$ with $f \equiv$ fine, and $c \equiv$ coarse.

1. Relax $\mathbf{P}^f \mathbf{y}_f^0 = \mathbf{v}$ for \mathbf{y}_f^1 (Jacobi, Gauss-Seidel ...)
2. Evaluate linear residual $\mathbf{res}^f = \mathbf{v} - \mathbf{P}^f \mathbf{y}_f^1$, and restrict to coarse grid, $\mathbf{res}^c = \mathcal{R} * \mathbf{res}^f$
3. Solve coarse grid problem, $\mathbf{P}^c \delta \mathbf{y}_c = \mathbf{res}^c$, for coarse grid correction $\delta \mathbf{y}_c$.
4. Prolongate coarse grid correction and update fine grid solution vector, $\mathbf{y}_f^2 = \mathbf{y}_f^1 + \mathcal{P} * \delta \mathbf{y}_c$
5. Relax $\mathbf{P}^f \mathbf{y}_f^2 = \mathbf{v}$ for \mathbf{y}_f^3 (Jacobi, Gauss-Seidel ...)

What are \mathcal{P} and \mathcal{R} ?

How does one form \mathbf{P}_c ?

Note: A multigrid (greater than 2) V-Cycle is realized by recursively inserting steps 1-5 into step 3 until a "very coarse" grid has been reached.

Additive Correction Multigrid

Assume a simple 2-D 5 point operator, penta-diagonal matrix.

$$\mathbf{P}^f = (P1^f, P2^f, P3^f, P4^f, P5^f) \quad (4)$$

Want solution to $\mathbf{P}\mathbf{y} = \mathbf{v}$.

$$\begin{aligned} P1_{i,j}^f * y_{i-1,j} + P2_{i,j}^f * y_{i,j-1} + P3_{i,j}^f * y_{i,j} \\ + P4_{i,j}^f * y_{i,j+1} + P5_{i,j}^f * y_{i+1,j} = v_{i,j} \end{aligned} \quad (5)$$

Have initial fine grid solution y^* ;

$$\begin{aligned} res_{i,j}^f = v_{i,j} - P1_{i,j}^f * y_{i-1,j}^* - P2_{i,j}^f * y_{i,j-1}^* \\ - P3_{i,j}^f * y_{i,j}^* - P4_{i,j}^f * y_{i,j+1}^* - P5_{i,j}^f * y_{i+1,j}^* \end{aligned} \quad (6)$$

Define (piece-wise constant prolongation, \mathcal{P}), for coarse grid correction.

$$\begin{aligned} y_{i,j} &= y_{i,j}^* + \delta y_{I,J} \\ y_{i+1,j} &= y_{i+1,j}^* + \delta y_{I,J} \\ y_{i,j+1} &= y_{i,j+1}^* + \delta y_{I,J} \\ y_{i+1,j+1} &= y_{i+1,j+1}^* + \delta y_{I,J} \end{aligned} \quad (7)$$

Additive Correction Multigrid (cont.)

Add the four equations (perhaps volume weighted) for cells $(i, j), (i+1, j), (i, j+1), (i+1, j+1)$ that result from inserting Eq. (7) into Eq. (5).

(resulting equation for cell (i, j))

$$\begin{aligned}
 & P1_{i,j}^f * (y_{i-1,j}^* + \delta y_{I-1,J}) + P2_{i,j}^f * (y_{i,j-1}^* + \delta y_{I,J-1}) \\
 & + P3_{i,j}^f * (y_{i,j}^* + \delta y_{I,J}) + P4_{i,j}^f * (y_{i,j+1}^* + \delta y_{I,J}) \\
 & + P5_{i,j}^f * (y_{i+1,j}^* + \delta y_{I,J}) = v_{i,j} \quad (8)
 \end{aligned}$$

After adding equations, collect like terms, using the definition (also perhaps volume weighted) for the coarse grid residual. (piece-wise constant Restriction, \mathcal{R})

$$res_{I,J}^c = res_{i,j}^f + res_{i+1,j}^f + res_{i,j+1}^f + res_{i+1,j+1}^f, \quad (9)$$

The following coarse grid correction equation results for cell (I, J) .

$$\begin{aligned}
 & P1_{I,J}^c * \delta y_{I-1,J} + P2_{I,J}^c * \delta y_{I,J-1} + P3_{I,J}^c * \delta y_{I,J} \\
 & + P4_{I,J}^c * \delta y_{I,J+1} + P5_{I,J}^c * \delta y_{I+1,J} = res_{I,J}^c \quad (10)
 \end{aligned}$$

Additive Correction Multigrid (cont.)

The coarse grid matrix is defined as follows,

$$\begin{aligned} P1_{I,J}^c &= P1_{i,j}^f + P1_{i,j+1}^f \\ P2_{I,J}^c &= P2_{i,j}^f + P2_{i+1,j}^f \\ P4_{I,J}^c &= P4_{i,j+1}^f + P4_{i+1,j+1}^f \\ P5_{I,J}^c &= P5_{i+1,j}^f + P5_{i+1,j+1}^f \\ P3_{I,J}^c &= \text{Sum of All } P^f \text{ s} \\ -P1_{I,J}^c - P2_{I,J}^c - P4_{I,J}^c - P5_{I,J}^c \end{aligned} \tag{11}$$

- Thus a simple, straight forward to form, coarse grid matrix has resulted from simple choices for \mathcal{R} and \mathcal{P} .
- This multigrid preconditioner is easy to add to an existing preconditioned GMRES solver.
- This simple multigrid concept originates from Settari and Aziz, SIAM J. Num. Anal., vol 10, 1973 and Hutchinson and Raithby, Num. Heat Tran., vol 9, 1986.

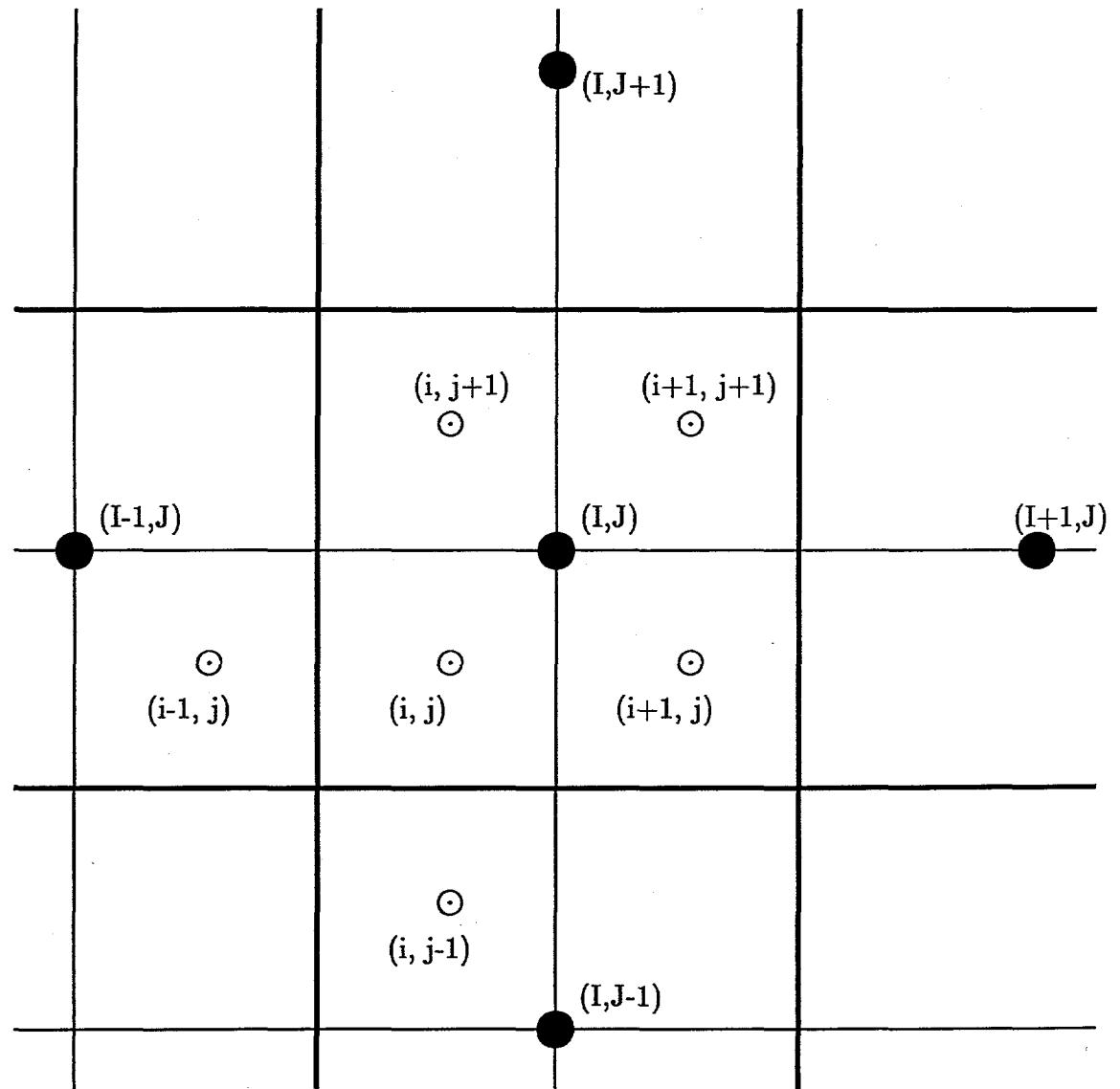


Figure 2: Two-level finite volume grid

Implicit Moment Method

Potential Equation

- Work with G. Lapenta and J.U. Brackbill on elliptic field solver for CELEST2D and CELEST3D (Knoll, Lapenta, and Brackbill, J. Comput. Phys., vol 149 1999).
- Maximum allowable time step is often governed by the ability to solve the electrostatic potential equation [H.X. Vu and J.U. Brackbill, Comput. Phys. Comm., Vol. 69 (1992)]:

$$\nabla \cdot [(\mathbf{I} + \mu(\vec{r})) \cdot \nabla \phi] = \rho(\vec{r}). \quad (12)$$

Here, \vec{r} is the position vector, $\mu(\vec{r})$ is a spatially varying electric susceptibility **TENSOR**, and $\rho(\vec{r})$ is the space charge.

- The potential equation is discretized using finite volumes, in a general coordinate system, which produces a 9-point numerical stencil in 2-D and a 27-point stencil in 3-D (Sulsky and Brackbill, J. Comput. Phys. , vol 96, 1991).
- The major difficulty arises from the fact that the off diagonal components of $\mu(\vec{r})$, which are proportional to δt , cause the resulting matrix equation for ϕ ($A\phi = \rho$) to be **NON-SYMMETRIC**. In some cases diagonal dominance may be lost.

RESULT 1

- A single grid Jacobi based preconditioner is compared to a multi-grid Jacobi based preconditioner.
- The number of sweeps (Jacobi iterations) is 3
- 5 GMRES vectors are stored and restart is employed with a linear convergence tolerance of 1.0×10^{-6} .
- Same time step on all grids, data averaged over 3 time steps
- UNIFORM grid, CRAY Y-MP

Grid	Single Grid (SG) Precond. GMRES Its.	Multigrid (MG) Precond. GMRES Its.	CPU ratio SG / MG
32×32	48	8.7	1.8
64×64	116	9	2.8
128×128	398	11	5.65
256×256	691	9	10.5

RESULT 2

- NON-UNIFORM grid (matrix may not be diagonally dominant)
- A single grid Jacobi based preconditioner is compared to a multi-grid Jacobi based preconditioner.
- The number of sweeps (Jacobi iterations) is 1
- 5 GMRES vectors are stored and restart is employed with a linear convergence tolerance of 1.0×10^{-6} .
- Same time step on all grids, data averaged over 3 time steps

Grid	Single Grid (SG)	Multigrid (MG)	CPU ratio
	Precond.	Precond	SG / MG
32×32	451	24	8.2
64×64	1672	39.5	11.5
128×128	6217	64	14.5
256×256	7322	39	19.8

- On 128×128 , with 10 GMRES vectors 39 iterations required, with 20 GMRES vectors stored 33 iterations required.

Multilevel Preconditioning and Newton-Krylov methods for 2-D Fokker-Planck Problems

- Work with L. Chacón and D.C Barnes on an efficient and energy conserving nonlinear solver for 2-D Fokker-Planck (single species, ions only) Problems. This is a nonlinear integro-differential problem ! (Chacón, Barnes, Knoll, Miley, LA-UR-99-????, submitted to J. Comput. Phys.)
- Problem statement:

$$\frac{\partial f}{\partial t} = -\Gamma \frac{\partial}{\partial \vec{v}} \cdot \left[f \frac{\partial H(f)}{\partial \vec{v}} - \frac{1}{2} \frac{\partial}{\partial \vec{v}} \cdot \left(\frac{\partial^2 G(f)}{\partial \vec{v} \partial \vec{v}} f \right) \right] \quad (13)$$

Here, f the distribution function, $\vec{v} = (v_\perp, v_r)$, $\Gamma = \frac{4\pi e^4 \lambda}{m}$, and e, m are the charge and mass, G and H are the Rosenbluth potentials which satisfy:

$$\nabla_{\vec{v}}^2 H = -8\pi f \quad (14)$$

$$\nabla_{\vec{v}}^2 G = H \quad (15)$$

- A typical, Non-energy conserving, implicit time integration is:

$$\frac{f^{n+1,k} - f^n}{\Delta t} = -\Gamma \frac{\partial}{\partial \vec{v}} \cdot \left[f^{n+1,k} \frac{\partial H(f^{n+1,k-1})}{\partial \vec{v}} - \frac{1}{2} \frac{\partial}{\partial \vec{v}} \cdot \left(\frac{\partial^2 G(f^{n+1,k-1})}{\partial \vec{v} \partial \vec{v}} f^{n+1,k} \right) \right] \quad (16)$$

2-D Fokker-Planck cont.

- Here, k is an iteration index, and the effects of the Rosenbluth potential are lagged on iteration level to produce a 9 banded matrix instead of a dense matrix. For acceptable energy conservation small time steps must be used.
- NOTE: MG preconditioned GMRES applied to solve Eq (16), is by itself a new and useful contribution.
- We employ matrix-free Newton-GMRES methods to solve an energy conservative form,

$$F(f^{n+1,k}) = \frac{f^{n+1,k} - f^n}{\Delta t} + \Gamma \frac{\partial}{\partial \vec{v}} \cdot \left[f^{n+1,k} \frac{\partial H(f^{n+1,k})}{\partial \vec{v}} - \frac{1}{2} \frac{\partial}{\partial \vec{v}} \cdot \left(\frac{\partial^2 G(f^{n+1,k})}{\partial \vec{v} \partial \vec{v}} f^{n+1,k} \right) \right] \quad (17)$$

- Eq (16), and the simple multigrid method of the previous section, are used as a preconditioner, thus the dense matrix is never formed.
- The Rosenbluth potentials are solved as part of evaluating the residual of Eq (17), as required from the Newton-GMRES iteration.

2-D Fokker-Planck cont.

- MG preconditioner uses 2 V-cycles and the smoother is 5 passes of symmetric Gauss-Seidel (SGS). Compare performance with single grid 10 SGS preconditioner for 2 different time steps.
- For time steps on the order of the collision frequency MG preconditioning is providing more than a factor of 5 reduction in GMRES iterations and CPU time.
- Note: Energy conservative discretization in velocity space is very important. See (Chacón, Barnes, Knoll, Miley, LA-UR-99-????, submitted to J. Comput. Phys.)

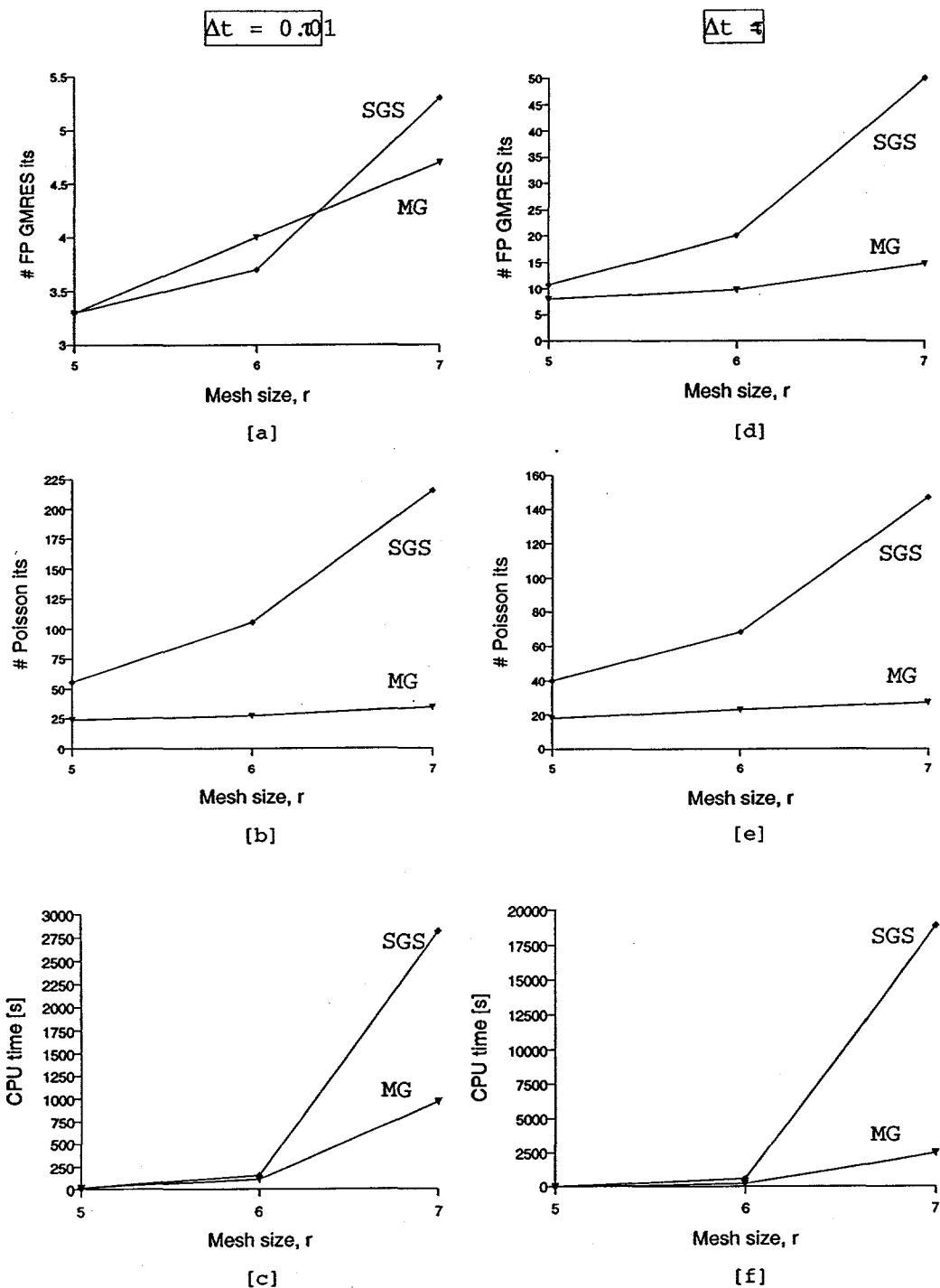


Figure 4:

Multilevel Newton-Krylov Methods

for 3-D MHD

- Work with J.U. Brackbill starting with 3-D FLIP-MHD code (J.U. Brackbill, J. Comput. Phys., vol. 96, 163-192, 1991).
- FLIP-MHD solves the field equations in a Lagrangian frame and advection is done with a particle method.

$$\frac{\partial \mathbf{U}}{\partial t} = \frac{d\mathbf{U}}{dt} - \mathbf{V} \cdot \nabla \mathbf{U} \quad (1)$$

$$\mathbf{U} = [\rho, \mathbf{V}, \mathbf{B}, \epsilon] \quad (2)$$

- Goal: Multilevel Newton-Krylov applied to non-Ideal MHD in 3-D for trans-sonic, trans-Alfvénic Applications
- Start: Replace existing conjugate residual (CR) solver (P.J. O'Rourke and A.A. Amsden LANL Report LA-10849-MS) with matrix-free Newton-GMRES solver and benchmark performance on 3-D Ideal MHD problem

Ideal MHD Equations in Lagrangian Frame

$$\frac{d\rho}{dt} + \rho(\nabla \cdot \mathbf{V}) = 0 \quad (3)$$

$$\frac{d\mathbf{V}}{dt} + \frac{1}{\rho}[\nabla P + \mathbf{J} \times \mathbf{B}] = 0 \quad (4)$$

$$\frac{d\mathbf{B}}{dt} + (\mathbf{B} \cdot \nabla) \mathbf{V} + \mathbf{B}(\nabla \cdot \mathbf{V}) = 0 \quad (5)$$

$$\frac{d\epsilon}{dt} + \frac{P}{\rho}(\nabla \cdot \mathbf{V}) = 0 \quad (6)$$

$$\mathbf{J} = \frac{1}{\mu} \nabla \times \mathbf{B} \quad (7)$$

$$P = \rho R T \quad (8)$$

$$\epsilon = C_v T \quad (9)$$

FLIP-MHD is a finite volume code. In this code ρ, ϵ and \mathbf{B} are at cell centers and velocities, \mathbf{V} , are at cell vertices.

3-D MHD cont.

- For trans-sonic and trans-Alfvénic problems there is often a desire to step over stiff time scales. In order to achieve this Eq (4) is discretized implicitly,

$$\mathbf{V}^{n+1} = \mathbf{V}^n - \frac{\Delta t}{\rho^{n+1}} [\nabla P^{n+1} + (\mathbf{J} \times \mathbf{B})^{n+1}] \quad (10)$$

- Currently the ICE method is used to help remove time step restrictions due to sound waves (this approximation will be removed by solving the energy equation coupled inside the Newton-Krylov method)

$$\Delta P = \Delta \rho R T + \rho R \Delta T \approx \Delta \rho R T \quad (11)$$

$$P^{n+1} = P^n + (\rho^{n+1} - \rho^n) R T^n \quad (12)$$

- Inserting Eq. (12) into Eq. (10) results in \mathbf{V}^{n+1} being a function of ρ^{n+1} and \mathbf{B}^{n+1} ,

$$\mathbf{V}^{n+1} = \mathbf{V}^n - \frac{\Delta t}{\rho^{n+1}} [\nabla (P^n + (\rho^{n+1} - \rho^n) R T^n) + (\mathbf{J} \times \mathbf{B})^{n+1}] \quad (13)$$

- Then inserting this into implicitly discretized forms of Eqs. (3) and (5) results in

$$F_\rho(\rho, \mathbf{B}) = \rho^{n+1} - \rho^n + \Delta t \rho^{n+1} (\nabla \cdot \mathbf{V}^{n+1}) \quad (14)$$

$$F_{\mathbf{B}}(\rho, \mathbf{B}) = \mathbf{B}^{n+1} - \mathbf{B}^n + \Delta t [(\mathbf{B}^{n+1} \cdot \nabla) \mathbf{V}^{n+1} + \mathbf{B}^{n+1} (\nabla \cdot \mathbf{V}^{n+1})] \quad (15)$$

3-D MHD cont.

- In 3-D this results in 4 coupled equations, $F_\rho, F_{Bx}, F_{By}, F_{Bz}$ and four unknowns ρ, Bx, By, Bz .
- Now we apply Newton's method $\mathbf{J}\delta\mathbf{x} = -\mathbf{F}(\mathbf{x})$ with

$$\mathbf{x} = [\rho_1, Bx_1, By_1, Bz_1, \dots, \rho_N, Bx_N, By_N, Bz_N] \quad (16)$$

$$\mathbf{F} = [F_{\rho_1}, F_{Bx_1}, F_{By_1}, F_{Bz_1}, \dots, F_{\rho_N}, F_{Bx_N}, F_{By_N}, F_{Bz_N}] \quad (17)$$

- matrix-free Newton-GMRES is used to drive the residual, $\mathbf{F}(\mathbf{x})$ below some tolerance.
- Right preconditioning is used (same as with the previous CR method). The block diagonal of the Jacobian (a 4x4 at each cell) is formed numerically and the preconditioning step is block diagonal scaling (same as before). For cell i the block diagonal is,

$$\mathbf{D}_i = \begin{bmatrix} \frac{\partial F_{\rho_i}}{\partial \rho_i} & \frac{\partial F_{\rho_i}}{\partial Bx_i} & \frac{\partial F_{\rho_i}}{\partial By_i} & \frac{\partial F_{\rho_i}}{\partial Bz_i} \\ \frac{\partial F_{Bx_i}}{\partial \rho_i} & \frac{\partial F_{Bx_i}}{\partial Bx_i} & \frac{\partial F_{Bx_i}}{\partial By_i} & \frac{\partial F_{Bx_i}}{\partial Bz_i} \\ \frac{\partial F_{By_i}}{\partial \rho_i} & \frac{\partial F_{By_i}}{\partial Bx_i} & \frac{\partial F_{By_i}}{\partial By_i} & \frac{\partial F_{By_i}}{\partial Bz_i} \\ \frac{\partial F_{Bz_i}}{\partial \rho_i} & \frac{\partial F_{Bz_i}}{\partial Bx_i} & \frac{\partial F_{Bz_i}}{\partial By_i} & \frac{\partial F_{Bz_i}}{\partial Bz_i} \end{bmatrix}.$$

- Problem: 3-D Kelvin-Helmholtz instability in the earth's magnetosphere.