A Hybrid Multigrid Technique for Computing Steady-State Solutions to Supersonic Flows.

Final Report
NASA/ASEE Summer Faculty Fellowship Program – 1992
Johnson Space Center

Prepared by:	Richard Sanders
Academic Rank:	Associate Professor
University & Rank:	University of Houston Department of Mathematics Houston, Texas 77204-3476
Directorate:	Engineering
Branch:	Aerosciences (EG3)
JSC Colleague:	Chien Li
Date Submitted:	September 21, 1992
Contact Number:	NGT-44-001-800

§1. Introduction. Recently, Li and Sanders [2] have introduced a class of finite difference schemes to approximate generally discontinuous solutions to hyperbolic systems of conservation laws. These equations have the form

$$\begin{aligned}\frac{\partial}{\partial t}\mathbf{q} + \nabla \cdot \mathbf{g}(\mathbf{q}) + \mathbf{s}(\mathbf{q}) = 0\\ \mathbf{q}(0) = \mathbf{q}_0,\end{aligned}$$

together with relevant boundary conditions. When modelling hypersonic spacecraft reentry, the differential equations above are frequently given by the compressible Euler equations coupled with a nonequilibrium chemistry model. For these applications, steady state solutions are often sought. Many tens (to hundreds) of supercomputer hours can be devoted to a single three space dimensional simulation. The primary difficulty is the inability to rapidly and reliably capture the steady state. In these notes, we demonstrate that a particular variant from the schemes presented in [2] can be combined with a particular multigrid approach to capture steady state solutions to the compressible Euler equations in one space dimension. We show that the rate of convergence to steady state coming from this multigrid implementation is vastly superior to the traditional approach of artificial time relaxation. Moreover, we demonstrate virtual grid independence. That is, the rate of convergence does not depend on the degree of spatial grid refinement.

Before continuing, we review the particular variant of the numerical discretization of $\partial_x \mathbf{g}(\mathbf{q})$ we wish to combine with multigrid. We assume that the problem to be solved is hyperbolic. That is, the Jacobian matrix of $\mathbf{g}(\mathbf{q})$, denoted here by $\partial_{\mathbf{q}}\mathbf{g}(\mathbf{q})$, has real eigenvalues $\lambda_i(\mathbf{q})$, $i = 1, \ldots, m$ and a complete set of eigenvectors $\mathbf{r}_i(\mathbf{q})$. The approximation of $\partial_x \mathbf{g}(\mathbf{q})$ in grid cell j is given by

$$\frac{1}{\Delta x}\left(\mathbf{h}_{\mathbf{g}}(\mathbf{q}_{j+1/2}; \mathbf{q}^l_{j+1/2}, \mathbf{q}^r_{j+1/2}) - \mathbf{h}_{\mathbf{g}}(\mathbf{q}_{j-1/2}; \mathbf{q}^l_{j-1/2}, \mathbf{q}^r_{j-1/2})\right),$$

where $\mathbf{h}_{\mathbf{g}}$ is a two point numerical flux function consistent to $\mathbf{g}$. That is, $\mathbf{h}_{\mathbf{g}}(\mathbf{p}; \mathbf{q}, \mathbf{q}) = \mathbf{g}(\mathbf{q})$. The particular numerical flux function we take below is of Roe type [1]

$$\mathbf{h}_{\mathbf{g}}(\mathbf{p}; \mathbf{a}, \mathbf{b}) = \frac{1}{2}\left(\mathbf{g}(\mathbf{a}) + \mathbf{g}(\mathbf{b}) - R(\mathbf{p})|\Lambda(\mathbf{p})|L(\mathbf{p})(\mathbf{b} - \mathbf{a})\right),$$

where $R(\mathbf{p})$ is the matrix of right eigenvectors to $\partial_{\mathbf{q}}\mathbf{g}(\mathbf{p})$, $L(\mathbf{p})$ the inverse to $R(\mathbf{p})$ and $|\Lambda(\mathbf{p})| = diag(|\lambda_i(\mathbf{p})| \vee \delta)$. We let $\mathbf{q}_{j+1/2} = \frac{1}{2}(\mathbf{q}_{j+1} + \mathbf{q}_j)$ and compute $\mathbf{q}^l_{j+1/2}$ and $\mathbf{q}^r_{j+1/2}$ according to the following recipe.

(i) At each cell interface $j + 1/2$, compute

$$\begin{aligned}D^{1,0} &= L(\mathbf{q}_{j+1/2})(\mathbf{q}_{j+1} - \mathbf{q}_j)\\ D^{2,+} &= L(\mathbf{q}_{j+1/2})(\mathbf{q}_{j+2} - 2\mathbf{q}_{j+1} + \mathbf{q}_j)\\ D^{2,-} &= L(\mathbf{q}_{j+1/2})(\mathbf{q}_{j+1} - 2\mathbf{q}_j + \mathbf{q}_{j-1})\end{aligned}$$

and componentwise ($i = 1, \dots, m$)

$$(c_{j+1/2})_i = \begin{cases} \min(|D_i^{2,-}|, 3|D_i^{1,0}|)\operatorname{sign}(D_i^{2,-}) & \text{if } \lambda_i(q_{j+1/2}) > 0 \\ \min(|D_i^{2,+}|, 3|D_i^{1,0}|)\operatorname{sign}(D_i^{2,+}) & \text{otherwise} \end{cases},$$

to obtain interface values

$$q^m_{j+1/2} = q_{j+1/2} - \frac{1}{6}R(q_{j+1/2})c_{j+1/2}.$$

(ii) Limit the interface values by

$$\begin{aligned} q^l_{j+1/2} &= q_j \\ &\quad + R(q_j)\left(\operatorname{minmod}(L(q_j)(q^m_{j+1/2} - q_j), \rho L(q_j)(q^m_{j-1/2} - q_j))\right) \\ q^r_{j+1/2} &= q_{j+1} \\ &\quad + R(q_{j+1})\left(\operatorname{minmod}(L(q_{j+1})(q^m_{j+1/2} - q_{j+1}), \rho L(q_{j+1})(q^m_{j+3/2} - q_{j+1}))\right) \end{aligned}$$

where $\rho > 1$ is called a *compression factor.*

Note that for ρ taken larger than 1 we have generically that $q^l_{j+1/2} = q^r_{j+1/2} = q^m_{j+1/2}$. Therefore, generically

$$h_g(q_{j+1/2}; q^l_{j+1/2}, q^r_{j+1/2}) = g(q^m_{j+1/2}).$$

In the case when $g(q) = q$ and $q \in \mathbf{R}^1$, the finite difference formula above reduces generically to

$$(\frac{\partial}{\partial x}q)_j \approx \frac{1}{\Delta x}\left(\frac{1}{2}(q_{j+1} - q_{j-1}) - \frac{1}{6}(q_{j+1} - 3q_j + 3q_{j-1} - q_{j-2})\right).$$

(See Section 3.2.)

In the next section we motivate the basics of multigrid and study its convergence for an upwind scheme. In Section 3 a hybrid of multigrid with an approximate Newton's iteration is introduced and analysed. In the last section, we numerically treat an example supersonic flow problem using the difference formula given above together with a multigrid algorithm developed below.

§2.1 The basics of multigrid. Textbooks are filled with iterative techniques to solve large, sparse linear systems. Such systems are generally encountered when seeking approximations to steady-state solutions of partial differential equations in one, two or three space dimensions. The rate of convergence offered by almost all iterative techniques unfortunately depend on the size of the system. Specifically, by increasing the number of spatial grid points, the iterative technique will require a greater number of iterations to achieve a given error reduction. This phenomena can be exhibited by considering the finite difference scheme

$$\frac{1}{\Delta x}(u_j - u_{j-1}) + f(x_j) = 0 \quad j = 1, 2, \ldots, J,$$
$$\frac{1}{\Delta x} = J,$$

with periodic boundary conditions,

$$u_0 = u_J,$$

used to approximate the solution to the differential equation

$$\frac{\partial}{\partial x}u + f(x) = 0,$$
$$u(0) = u(1).$$

(The finite difference scheme above has a nonunique solution provided that $\sum_{j=1}^{J} f(x_j) = 0$.) Letting $\mathbf{u}$ represent the J dimensional vector of unknowns u_j, and $\mathbf{f}$ the J dimensional vector of $f(x_j)$, we can write the finite difference scheme above symbolically as

$$D_J\mathbf{u} + \mathbf{f} = 0, \tag{2.1}$$

where D_J is a $J \times J$ matrix, and we will consider the rate of convergence of the artificial time relaxation technique

$$\begin{aligned} \mathbf{u}^k &= \mathbf{u}^{k-1} - \rho(D_J\mathbf{u}^{k-1} + \mathbf{f}), \\ \mathbf{u}^0 &= \mathbf{v}, \end{aligned} \tag{2.2}$$

where $\rho > 0$ is a relaxation parameter and $\mathbf{v}$ is an arbitrary starting vector. Clearly, if $\lim_{k\to\infty} \mathbf{u}^k$ exists, the limit solves (2.1).

Recall the discrete Fourier transform of a J-periodic grid function v_j:

$$\hat{v}_l = \frac{1}{\sqrt{J}} \sum_{j=1}^{J} v_j e^{\frac{2\pi i}{J} lj},$$

and its associated inversion formula,

$$v_j = \frac{1}{\sqrt{J}} \sum_{l=1}^{J} \hat{v}_l e^{-\frac{2\pi i}{J} lj}.$$

Let $\mathbf{u}^k$ denote the kth iterate coming from (2.2), let $\mathbf{u}$ denote the unique stationary solution to (2.1) normalized so that $\sum_{j=1}^{J} u_j = \sum_{j=1}^{J} u_j^0$, and let $\epsilon^k = \mathbf{u} - \mathbf{u}^k$ denote the artificial time iteration error after k iterations. By taking

$$\epsilon_j^k = \frac{1}{\sqrt{J}} \sum_{l=1}^{J} \hat{\epsilon}_l^k e^{-\frac{2\pi i}{J} lj},$$

one easily calculates that the the lth Fourier coefficient of error at iteration step k is given explicitly by

$$\hat{\epsilon}_l^k = \left((1 - \frac{\rho}{\Delta x}) + \frac{\rho}{\Delta x} e^{\frac{2\pi i}{J} l}\right)^k \hat{\epsilon}_l^0.$$

(Note that the normalization of $\mathbf{u}$ implies $\hat{\epsilon}_J^0 = 0$.) Taking $\sigma \equiv \rho/\Delta x < 1$, we have that

$$|(1 - \frac{\rho}{\Delta x}) + \frac{\rho}{\Delta x} e^{\frac{2\pi i}{J} l}| < 1,$$

for each $l = 1, 2, \ldots, J-1$. Moreover, the high frequency components of the error are contained in the Fourier coefficients $\hat{\epsilon}_l^k$ with $l \approx J/2$. From above, we see that these high frequency components decay at a rate on the order of

$$|1 - 2\sigma|.$$

Therefore, these coefficients decay at a rate that is independent of the number of grid points J. On the other hand, the low frequency components of the error are contained in $\hat{\epsilon}_l^k$ with $l \approx 1$ or with $l \approx J-1$. These components decay at a rate on the order of

$$1 - \frac{2\pi^2}{J^2} \sigma(1 - \sigma).$$

Therefore, for this example, doubling the number of grid points J can necessitate many more than twice the number of artificial time iterations to achieve a given error reduction.

The basic idea of multigrid is the following: One attempts to capture the low frequency components of $\mathbf{u}$ on a coarse mesh. This can be accomplished by a variety of methods. The coarse grid information is then included into the current guess on the fine grid, and the high frequency components of the error can be rapidly damped there by using an iterative technique such as the one above. Normally, the divide and conquer strategy is employed in multigrid. That is, a nested sequence of coarse meshes is utilized to rapidly capture all frequencies of the solution.

To illustrate the multigrid approach, we will apply an *ideal* two grid multigrid strategy to example (2.1) above. Given the current approximation $\mathbf{u}^k$ to $\mathbf{u}$, we have that

$$D_J\mathbf{u} + \mathbf{f} = D_J(\mathbf{u}^k + \epsilon^k) + \mathbf{f} = 0$$

or,

$$D_J\epsilon^k + \mathbf{r}^k = 0, \tag{2.3}$$

where the residual is defined by $\mathbf{r}^k = D_J\mathbf{u}^k + \mathbf{f}$, and D_J is the $J \times J$ difference matrix defined above. The solution to the residual equation (2.3) is approximated by solving

$$D_{J/2}\epsilon^k_{J/2} + \mathbf{r}^k_{J/2} = 0$$

where,

$$\mathbf{r}^k_{J/2} = I_{(J\rightarrow J/2)}\mathbf{r}^k.$$

$I_{(J\rightarrow J/2)}$ denotes an injection operator that takes vectors from $\mathbf{R}^J$ into the lower dimensional space $\mathbf{R}^{J/2}$. (Solving the coarse grid equation above exactly is what defines an ideal two grid strategy. In practice, multigrid is nested. That is, a nested sequence of lower dimensional multigrid iterations is applied to a nested sequence of coarser grid residual equations.) The fine grid error is then approximated by

$$\epsilon^k \approx I_{(J/2\rightarrow J)}\epsilon^k_{J/2} \equiv \epsilon^k_{cg},$$

where $I_{(J/2\rightarrow J)}$ denotes an interpolation operator that takes vectors from $\mathbf{R}^{J/2}$ back into $\mathbf{R}^J$. ϵ^k_{cg} is called the *coarse grid correction*. One expects that ϵ^k_{cg} will agree well with the true error in the low frequencies. However, it in general will not agree well with the true error in the high frequency regime. To capture the high frequencies, we may for example use the artificial time scheme above. e.g., for $n = 1, 2, \ldots, \nu$

$$\mathbf{e}^n = \mathbf{e}^{n-1} - \rho(D_J\mathbf{e}^{n-1} + \mathbf{r}^k), \tag{2.4}$$

with

$$\mathbf{e}^0 = \epsilon^k_{cg}.$$

(This step of multigrid is often referred to as the *smoother*. The term smoother is in fact a misnomer. In actuality, this step is used to capture the high frequency components of the error.) Finally, since

$$\mathbf{u} = \mathbf{u}^k + \epsilon^k \approx \mathbf{u}^k + \mathbf{e}^\nu,$$

we update $\mathbf{u}^k$ via

$$\mathbf{u}^{k+1} = \mathbf{u}^k + \mathbf{e}^\nu.$$

This defines one cycle of a simple two level multigrid technique. We should remark that the update step above is equivalent to setting

$$\mathbf{u}^{k+1} = S^\nu_J(\mathbf{u}^k + \epsilon^k_{cg}, \mathbf{f}),$$

where $S^\nu_J(\mathbf{v}, \mathbf{f})$ denotes ν iterations of the basic iteration scheme (2.2) on grid J.

§2.2 Ideal two grid analysis for the example first order one sided scheme. A multigrid strategy requires three basic ingredients. They are:

(i) The so-called smoother; see (2.4) above.

(ii) The fine to coarse grid residual injection operator; $I_{(J\to J/2)}$ above.

(iii) The coarse to fine grid error interpolation operator; $I_{(J/2\to J)}$ above.

The performance of a multigrid algorithm relies heavily on the choice of these ingredients. In this subsection, we analyse the convergence rate of the strategy outlined above taking the artificial time relaxation technique (2.2) as the smoother, and

$$(I_{(J\to J/2)}\mathbf{r})_j = \frac{1}{2}(r_{2j} + r_{2j-1}), \tag{2.5}$$

as the fine to coarse grid injection operator, and

$$\begin{aligned} (I_{(J/2\to J)}\epsilon)_{2j} &= \epsilon_j \\ (I_{(J/2\to J)}\epsilon)_{2j-1} &= \epsilon_j, \end{aligned} \tag{2.6}$$

as the coarse to fine grid interpolation operator. In the context of finite volume, these intergrid operators seem most natural. A multigrid iteration step above is written symbolically as

$$\begin{aligned} \epsilon^k_{cg} &= I_{(J/2\to J)} S^{\infty}_{J/2}(0, I_{(J\to J/2)}\mathbf{r}^k) \\ \mathbf{u}^{k+1} &= S^{\nu}_{J}(\mathbf{u}^k + \epsilon^k_{cg}, \mathbf{f}). \end{aligned} \tag{2.7}$$

Using the fact that $\mathbf{u} = S^{\nu}_{J}(\mathbf{u}, \mathbf{f})$, we may write (2.7) in terms of the error ϵ^k as

$$\begin{aligned} \epsilon^k_{cg} &= I_{(J/2\to J)} S^{\infty}_{J/2}(0, -I_{(J\to J/2)} D_J \epsilon^k) \\ \epsilon^{k+1} &= S^{\nu}_{J}(\epsilon^k - \epsilon^k_{cg}, 0), \end{aligned} \tag{2.8}$$

where again we normalize $\mathbf{u}$ such that $\sum_{j=1}^{J}(u_j - u^0_j) \equiv \sum_{j=1}^{J} \epsilon^0_j = 0$.

Unfortunately, unlike the single grid example above, the two grid multigrid algorithm (2.8) does not completely decouple into Fourier modes. Following [3], a J-periodic grid function $\mathbf{v}$ can be decomposed into a Fourier series on the odd grid points, and a Fourier series on the even grid points. The explicit formulae are

$$v_j = \begin{cases} \dfrac{1}{\sqrt{J/2}} \displaystyle\sum_{l=1}^{J/2} \tilde{v}_l e^{-\frac{2\pi i}{J/2} l(j+1)/2} & \text{if } j = 1, 3, \ldots, J-1 \\ \dfrac{1}{\sqrt{J/2}} \displaystyle\sum_{l=1}^{J/2} \tilde{\tilde{v}}_l e^{-\frac{2\pi i}{J/2} lj/2} & \text{if } j = 2, 4, \ldots, J \end{cases} . \tag{2.9}$$

In fact, if $v_j = \frac{1}{\sqrt{J}} \sum_{l=1}^{J} \hat{v}_l e^{-\frac{2\pi i}{J} lj}$, a simple calculation will reveal that

$$\begin{aligned} \bar{v}_l &= \frac{\sqrt{2}}{2} \left(\hat{v}_l - \hat{v}_{l+J/2}\right) e^{\frac{2\pi i}{J} l}, \\ \bar{\bar{v}}_l &= \frac{\sqrt{2}}{2} \left(\hat{v}_l + \hat{v}_{l+J/2}\right), \end{aligned} \tag{2.10}$$

for each $l = 1, 2, \ldots, J/2$.

(2.9) and (2.10) allow us to examine the degree to which the coarse grid correction ϵ^k_{cg} approximates the true error ϵ^k. Writing

$$\epsilon^k_j = \frac{1}{\sqrt{J}} \sum_{l=1}^{J} \hat{\epsilon}^k_l e^{-\frac{2\pi i}{J} lj},$$

or equivalently

$$\epsilon^k_j = \begin{cases} \dfrac{1}{\sqrt{J/2}} \displaystyle\sum_{l=1}^{J/2} \bar{\epsilon}^k_l e^{-\frac{2\pi i}{J/2} l(j+1)/2} & \text{if } j = 1, 3, \ldots, J-1 \\ \dfrac{1}{\sqrt{J/2}} \displaystyle\sum_{l=1}^{J/2} \bar{\bar{\epsilon}}^k_l e^{-\frac{2\pi i}{J/2} lj/2} & \text{if } j = 2, 4, \ldots, J \end{cases},$$

we compute that

$$\begin{aligned} \widetilde{D_J \epsilon^k}_l &= \frac{1}{\Delta x} \left(\bar{\epsilon}^k_l - \bar{\bar{\epsilon}}^k_l e^{\frac{2\pi i}{J/2} l}\right), \\ \widetilde{\widetilde{D_J \epsilon^k}}_l &= \frac{1}{\Delta x} \left(\bar{\bar{\epsilon}}^k_l - \bar{\epsilon}^k_l\right). \end{aligned}$$

Therefore, on the coarse grid, with j running from 1 to $J/2$, we have

$$\begin{aligned} &(I_{(J \to J/2)} D_J \epsilon^k)_j = \frac{1}{2} \left((D_J \epsilon^k)_{2j-1} + (D_J \epsilon^k)_{2j}\right) \\ &= \frac{1}{\sqrt{J/2}} \sum_{l=1}^{J/2} \frac{1}{2} \left(\widetilde{D_J \epsilon^k}_l + \widetilde{\widetilde{D_J \epsilon^k}}_l\right) e^{-\frac{2\pi i}{J/2} lj} = \frac{1}{\sqrt{J/2}} \sum_{l=1}^{J/2} \frac{1}{2\Delta x} (1 - e^{\frac{2\pi i}{J/2} l}) \bar{\bar{\epsilon}}^k_l e^{-\frac{2\pi i}{J/2} lj}. \end{aligned}$$

Solving the coarse grid problem exactly leads to

$$S^{\infty}_{J/2}(0, -I_{(J \to J/2)} D_J \epsilon^k) = \frac{1}{\sqrt{J/2}} \sum_{l=1}^{J/2} \bar{\bar{\epsilon}}^k_l e^{-\frac{2\pi i}{J/2} lj},$$

and we find, after applying the coarse to fine grid interpolation (2.6), that the coarse grid correction is given by

$$(\epsilon^k_{cg})_{2j-1} = (\epsilon^k_{cg})_{2j} = \frac{1}{\sqrt{J/2}} \sum_{l=1}^{J/2} \bar{\bar{\epsilon}}^k_l e^{-\frac{2\pi i}{J/2} lj}.$$

Making use of the formulae in (2.10), we can write the coarse grid correction in terms of $\{\hat{\epsilon}^k_l\}_{l=1}^J$

$$(\hat{\epsilon}^k_{cg})_l = \frac{1}{2}(1 + e^{-\frac{2\pi i}{J}l})(\hat{\epsilon}^k_l + \hat{\epsilon}^k_{l+J/2}) \tag{2.11a}$$

$$(\hat{\epsilon}^k_{cg})_{l+J/2} = \frac{1}{2}(1 - e^{-\frac{2\pi i}{J}l})(\hat{\epsilon}^k_{l+J/2} + \hat{\epsilon}^k_l) \tag{2.11b}$$

$$(l = 1, 2, \ldots, J/2).$$

Notice that for $l \approx 1$, the coarse grid correction coefficients given in (2.11a) agree well with the true error coefficients modulo a high frequency component. The same conclusion follows from (2.11b) with $l + J/2 \approx J$. That is, modulo high frequency pollution, the coarse grid correction accurately predicts the low frequency components of the error.

Finally, after applying ν iterations of the smoother $(\epsilon^{k+1} = S^\nu_J(\epsilon^k - \epsilon^k_{cg}, 0))$, one cycle of this ideal multigrid iteration yields an error which satisfies the coupled system

$$\begin{pmatrix} \hat{\epsilon}^{k+1}_l \\ \hat{\epsilon}^{k+1}_{l+J/2} \end{pmatrix} = \frac{1}{2} \begin{pmatrix} a^\nu_l(1 - e^{-\frac{2\pi i}{J}l}) & -a^\nu_l(1 + e^{-\frac{2\pi i}{J}l}) \\ -a^\nu_{l+J/2}(1 - e^{-\frac{2\pi i}{J}l}) & a^\nu_{l+J/2}(1 + e^{-\frac{2\pi i}{J}l}) \end{pmatrix} \begin{pmatrix} \hat{\epsilon}^k_l \\ \hat{\epsilon}^k_{l+J/2} \end{pmatrix}, \tag{2.12}$$

where

$$a^\nu_l = ((1 - \sigma) + \sigma e^{\frac{2\pi i}{J}l})^\nu.$$

We calculate the eigenvalues of the amplification matrix above:

$$\lambda_1 = 0$$
$$\lambda_2 = \frac{1}{2}(a^\nu_l(1 - e^{-\frac{2\pi i}{J}l}) + a^\nu_{l+J/2}(1 + e^{-\frac{2\pi i}{J}l})).$$

In the special case when $\nu = 1$ and $\sigma = 1/2$, we have that $\lambda_2 = 0$. Therefore, this multigrid scheme yields the exact solution **u** after at most two cycles regardless of its initial guess. More generally for any $\nu \geq 1$ and any fixed $0 < \sigma < 1$, one can show, independently of the number of grid points J, that $|\lambda_2|$ is strictly less than one. So in the general case, the rate of convergence of this ideal multigrid scheme does not degrade with grid size.

As mentioned above, the ideal scheme is not practical. One does not wish to solve the coarse grid problem exactly. Nevertheless, the analysis above does indicate that the fully nested multigrid approach should converge rapidly in a grid size independent manner.

§3.1 A hybrid of multigrid. Consider for the moment the possibly nonlinear differential equation

$$\frac{\partial}{\partial x}g(u)+f(x)=0,$$

together with given boundary conditions. Assuming that this equation has a solution, Newton's method is developed by setting $u=u^k+\epsilon^k$, where u^k is the current guess to the solution, and expanding $g(u^k+\epsilon^k)=g(u^k)+g'(u^k)\epsilon^k+\cdots$. Therefore,

$$0=\frac{\partial}{\partial x}g(u)+f(x)\approx\frac{\partial}{\partial x}(g(u^k)+g'(u^k)\epsilon^k)+f(x),$$

and we let ϵ^k_{nm} solve the linearized equation

$$\frac{\partial}{\partial x}(g'(u^k)\epsilon^k_{nm})+\frac{\partial}{\partial x}g(u^k)+f(x)=0,$$

to update

$$u^{k+1}=u^k+\epsilon^k_{nm}.$$

This idea extends equally well to finite difference schemes associated to this differential equation. That is, if we seek the solution to the finite difference scheme

$$D_J(g;\mathbf{u})+\mathbf{f}=0,$$

where $D_J(g;\mathbf{u})$ denotes a finite difference operator consistent to $\partial_x g(u)$, we solve

$$\partial_{\mathbf{u}}(D_J(g;\mathbf{u}^k))\epsilon^k_{nm}+D_J(g;\mathbf{u}^k)+\mathbf{f}=0,$$

where $\partial_{\mathbf{u}}(D_J(g;\mathbf{u}^k))$ denotes the Jacobian matrix of $D_J(g;\mathbf{u})$ at $\mathbf{u}=\mathbf{u}^k$, and update

$$\mathbf{u}^{k+1}=\mathbf{u}^k+\epsilon^k_{nm}.$$

Newton's method has two main drawbacks: (i) Each iteration requires the inversion of a large linear system. (ii) The Jacobian matrix of $D_J(g;\mathbf{u}))$ may be quite complicated to compute analytically. In fact, for many modern finite difference schemes, the function $D_J(g;\mathbf{u}))$ may lack the necessary smoothness to gain the quadratic convergence offered by Newton's method. We propose the following simplification: Suppose we wish to solve

$$D_J^{(hi)}(g;\mathbf{u})+\mathbf{f}=0,$$

where $D_J^{(hi)}(g;\mathbf{u})$ denotes a (possibly high order) finite difference operator consistent with $\partial_x(g(u))$. Let $D_J^{(lo)}(g'(u))\epsilon$ denote a (possibly low order) finite difference operator that is consistent to $\partial_x(g'(u)\epsilon)$. Rather than solving the correct linearization of the finite difference scheme above, we solve (or approximate the solution of)

$$D_J^{(lo)}(g'(u^k))\epsilon^k_{\underset{\sim}{nm}}+D_J^{(hi)}(g;\mathbf{u}^k)+\mathbf{f}=0, \tag{3.1}$$

to update $\mathbf{u}^{k+1} = \mathbf{u}^k + \epsilon^k_{\widetilde{nm}}$. Certainly there is no reason to expect quadratic convergence from this approach. Therefore, rather than solving (3.1) exactly, we approximate $\epsilon^k_{\widetilde{nm}}$ by applying multigrid to (3.1). Specifically, suppose that $\mathbf{u}^k$ is known. Then

(3.2)

Step 1: Compute $R(\mathbf{u}^k) = D_J^{(hi)}(g; \mathbf{u}^k) + \mathbf{f}$.

Step 2: Apply $\eta \geq 1$ cycles of multigrid with initial guess 0 to approximate the solution to

$$D_J^{(lo)}(g'(\mathbf{u}^k))\epsilon^k_{\widetilde{nm}} + R(\mathbf{u}^k) = 0,$$

calling the result ϵ^k_{mg}.

Step 3: Update $\mathbf{u}^{k+1} = \mathbf{u}^k + \epsilon^k_{mg}$.

Step 4: (Optional) Steps 1-3 may not capture the high frequency components of $\mathbf{u}$ well. Therefore, applying a few iteration of a smoother consistent to $D_J^{(hi)}(g; \mathbf{u}) + \mathbf{f} = 0$ may be advised; (see below).

§3.2 Convergence of the hybrid multigrid approach for a model problem. Whether iterates coming from (3.2) converge or not depends in a crucial way on the choice of both $D_J^{(lo)}$ and $D_J^{(hi)}$. We again only perform ideal two grid multigrid analysis to the linear problem $g(u) = u$. The problem is made somewhat more interesting by taking an upwind third order scheme for $D_J^{(hi)}$

$$(3.3a) \qquad (D_J^{(hi)}\mathbf{u})_j = \frac{1}{2\Delta x}(u_{j+1} - u_{j-1}) - \frac{1}{6\Delta x}(u_{j+1} - 3u_j + 3u_{j-1} - u_{j-2})$$

and a first order upwind scheme for $D_J^{(lo)}$

$$(3.3b) \qquad (D_J^{(lo)}\epsilon)_j = \frac{1}{\Delta x}(\epsilon_j - \epsilon_{j-1}),$$

together with periodic boundary conditions. (Note that (3.3a) is third order in the sense of cell averages.) We still require that $\sum_{j=1}^J f(x_j) = 0$ for solvability.

Before studying (3.1) & (3.3) hybridized with multigrid, observe that if $\epsilon^k = \mathbf{u} - \mathbf{u}^k$ (where again $\mathbf{u}$ is normalized such that $\sum_{j=1}^J (u_j - u_j^0) = 0$), then $\epsilon^k_{\widetilde{nm}}$ (see 3.1) satisfies

$$(3.4) \qquad (\hat{\epsilon}^k_{\widetilde{nm}})_l = \psi_l \hat{\epsilon}^k_l,$$

where the amplification factor ψ_l is given by

$$\psi_l = \frac{(-i\sin(\theta_l) - 1/3(\cos(\theta_l) - 1)(1 - e^{i\theta_l}))}{(1 - e^{i\theta_l})}, \qquad \theta_l = \frac{2\pi}{J}l.$$

The error reduction rate expected from algorithm (3.1) (or equivalently algorithm (3.2) with Step 2 solving exactly) is on the order of $\max_{1\leq l\leq J-1} |1-\psi_l| \approx 0.53$ and the maximum is attained in the mid frequencies. For the high frequencies $l \approx J/2$ we have that $|1-\psi_l| \approx 1/3$, and for the low frequencies $l \approx 1$ or $l \approx J-1$ we have that $|1-\psi_l| \approx 0$. So we see that the low frequency components of the error are captured quite rapidly whereas the mid to high frequencies decay less rapidly. Step 4 can therefore be utilized to knock down the mid to high frequency components of the error without affecting the low frequency convergence.

The results of Section 2 make the analysis of the hybrid multigrid algorithm (3.2) applied to the difference operators in (3.3) an easy matter. Since $\epsilon^k_{\underset{nm}{\sim}} - \epsilon^k_{mg}$ is the error of Step 2, using (2.12) we find that

$$\begin{pmatrix} (\widehat{\epsilon}^k_{\underset{nm}{\sim}} - \widehat{\epsilon}^k_{mg})_l \\ (\widehat{\epsilon}^k_{\underset{nm}{\sim}} - \widehat{\epsilon}^k_{mg})_{l+J/2} \end{pmatrix}$$
$$= \left[\frac{1}{2}\begin{pmatrix} a^\nu_l(1-e^{-\frac{2\pi i}{J}l}) & -a^\nu_l(1+e^{-\frac{2\pi i}{J}l}) \\ -a^\nu_{l+J/2}(1-e^{-\frac{2\pi i}{J}l}) & a^\nu_{l+J/2}(1+e^{-\frac{2\pi i}{J}l}) \end{pmatrix}\right]^\eta \begin{pmatrix} (\widehat{\epsilon}^k_{\underset{nm}{\sim}})_l \\ (\widehat{\epsilon}^k_{\underset{nm}{\sim}})_{l+J/2} \end{pmatrix}.$$

So if we denote the 2×2 multigrid amplification matrix above by $\widehat{M}_l^{(\sigma,\nu,\eta)}$, and define

$$\widehat{\Psi}_l = \begin{pmatrix} \psi_l & 0 \\ 0 & \psi_{l+J/2} \end{pmatrix},$$

then referring back to (3.4) we find that the 2×2 matrices

$$\widehat{A}_l = \left(I - (I - \widehat{M}_l^{(\sigma,\nu,\eta)})\widehat{\Psi}_l\right), \qquad (l = 1, 2, \ldots, J/2), \tag{3.5}$$

define the amplification matrix to the full (Steps 1-3) hybrid algorithm (3.2). The spectral radius of the hybrid amplification factor is plotted as a function of σ in Figure 1. (Recall that σ is the ratio of the artificial time step size ρ to the grid size Δx in the approximate linearization multigrid smoother.) Note that for $\sigma = 0.40$ we actually achieve better error reduction using multigrid in Step 2 than if we solved (3.1) exactly.

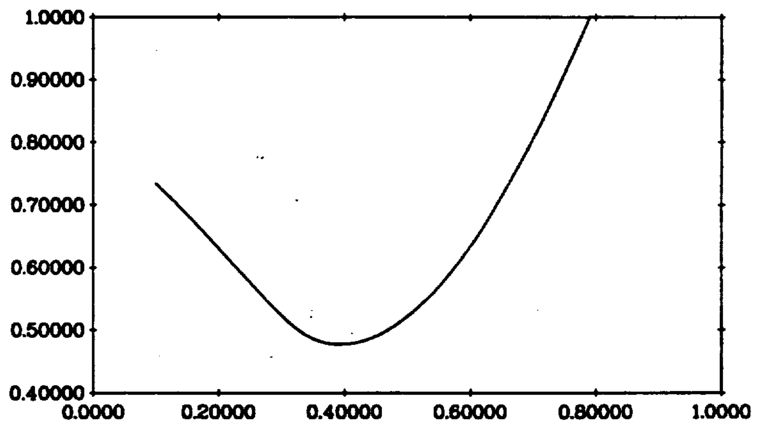

Figure 1. The spectral radius of $diag(\hat{A}_l)$ plotted against σ. ($\nu = 2, \eta = 2$.)

Notice the sensitivity of the error reduction rate to σ in Figure 1. This clearly indicates the relationship of multigrid error reduction to linearization wave speeds. Therefore, for nonlinear problems where the wave speeds are not constant, (especially for nonlinear systems with multiple wave speeds), we expect a poor convergence rate from the primitive algorithm outlined above. This problem will be addressed in the next section.

§4. Application to the one dimensional Euler equations. The partial differential equations that govern the flow of a compressible and inviscid gas in a quasi one dimensional expanding duct are

$$\begin{aligned}
&\frac{\partial}{\partial t}(\rho A) + \frac{\partial}{\partial x}(\rho u A) = 0\\
&\frac{\partial}{\partial t}(\rho u A) + \frac{\partial}{\partial x}\left((\rho u^2 + p)A\right) - p\frac{\partial}{\partial x}A = 0\\
&\frac{\partial}{\partial t}(\rho e A) + \frac{\partial}{\partial x}\left((\rho e + p)uA\right) = 0,
\end{aligned}$$

where ρ is the fluid's density, u its velocity and e its total energy per unit mass. The given function $A = A(x)$ defines the cross sectional area of the duct as a function of position along its length. p represents the fluid's pressure and is given by the equation of state $p = (\gamma - 1)\rho(e - u^2/2)$. We take $\gamma = 1.4$ in the results presented below. We seek a steady state solution to problem on the interval $0 \leq x \leq 10$, taking a supersonic inflow boundary condition at $x = 0$: $(\rho, u, p) = (0.502, 1.299, 0.3809)$; and a subsonic outflow condition at $x = 10$: $(\rho, u, p) = (0.776, 0.5063, 0.7475)$. The cross sectional area is given by $A(x) = 1.398 + 0.347\tanh(0.8x - 4)$.

To demonstrate a need for a technique that speeds convergence to steady state, we seek a steady state to this duct example by applying the third order accurate finite difference scheme from the introduction together with artificial time relaxation. Specifically, we iterate $k = 1, 2, \ldots$

$$\mathbf{q}^k = \mathbf{q}^{k-1} - \rho R^{(hi)}(\mathbf{q}^{k-1}),$$

where the residual $R^{(hi)}(\mathbf{q})$ is given by

$$(R^{(hi)}(\mathbf{q}))_j = \frac{1}{\Delta x}\left(\mathbf{h}_\mathbf{g}(\mathbf{q}_{j+1/2}; \mathbf{q}^l_{j+1/2}, \mathbf{q}^r_{j+1/2}) - \mathbf{h}_\mathbf{g}(\mathbf{q}_{j-1/2}; \mathbf{q}^l_{j-1/2}, \mathbf{q}^r_{j-1/2})\right) + \mathbf{s}(\mathbf{q}_j).$$

The compression factor in the slope limiting above as well as in all other examples presented is taken to be 3. We initialize the iteration by a linear interpolation of the boundary data in the conserved variables. The relaxation factor ρ is taken so that the CFL number (the ratio of time step size times the fastest wave speed to the space step size) is 0.30 and we

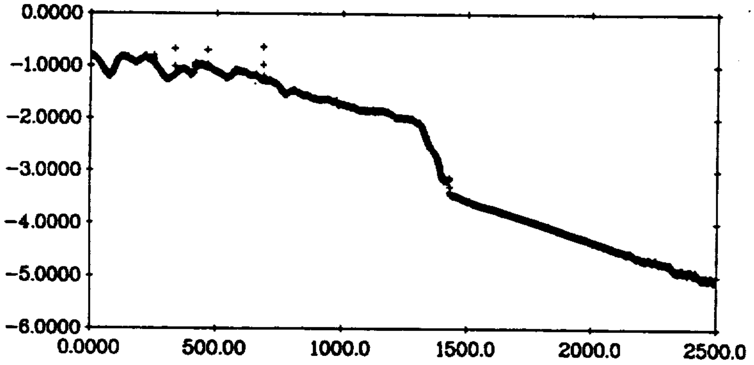

Figure 2. $\frac{1}{2}\log(\frac{1}{64}\sum_{j=1}^{64}(R_1^{(hi)}(q^k))_j^2)$ plotted against iterations for the artificial time scheme.

use 64 grid points on the interval [0,10]. In Figure 2, we plot the base 10 log of the L_2 norm of the (first component of the) residual versus the number of iterations.

To apply the multigrid algorithm (3.2), we need to decide on the approximate linearization of the third order scheme used above. We take for $D^{(lo)}(g'(\mathbf{q}))\epsilon$ the following first order scheme

$$\frac{1}{\Delta x}(\mathbf{h}^{(lo)}_{\partial_\mathbf{q}\mathbf{g}}(\mathbf{q}_{j+1/2};\epsilon_j,\epsilon_{j+1}) - \mathbf{h}^{(lo)}_{\partial_\mathbf{q}\mathbf{g}}(\mathbf{q}_{j-1/2};\epsilon_{j-1},\epsilon_j)),$$

where

$$\begin{aligned}&\mathbf{h}^{(lo)}_{\partial_\mathbf{q}\mathbf{g}}(\mathbf{q}_{j+1/2};\epsilon_j,\epsilon_{j+1})\\ &\quad=\frac{1}{2}(\partial_\mathbf{q}\mathbf{g}(\mathbf{q}_{j+1/2})(\epsilon_{j+1}+\epsilon_j) - |\partial_\mathbf{q}\mathbf{g}(\mathbf{q}_{j+1/2})|(\epsilon_{j+1}-\epsilon_j)).\end{aligned}$$

As usual, $|\partial_\mathbf{q}\mathbf{g}(\mathbf{q})|$ denotes the matrix $R(\mathbf{q})|\Lambda(\mathbf{q})|L(\mathbf{q})$. The obvious choice for a smoothing iteration scheme is

$$\begin{aligned}\epsilon_j^{k+1} = \epsilon_j^k - \rho\Big(&\frac{1}{\Delta x}(\mathbf{h}^{(lo)}_{\partial_\mathbf{q}\mathbf{g}}(\mathbf{q}_{j+1/2};\epsilon_j^k,\epsilon_{j+1}^k)\\ &- \mathbf{h}^{(lo)}_{\partial_\mathbf{q}\mathbf{g}}(\mathbf{q}_{j-1/2};\epsilon_{j-1}^k,\epsilon_j^k)) + \partial_\mathbf{q}\mathbf{s}(\mathbf{q}_j)\epsilon_j^k + (R^{(hi)}(\mathbf{q}))_j\Big),\end{aligned}$$

where ρ is fixed according to the largest wave speed of the variable coefficient problem. However, from our analysis in Section 3, we can expect this approach to yield very poor convergence rates. We base this conclusion on the following observation: The choice of ρ for the smoother above must be based on the largest eigenvalue of $\partial_\mathbf{q}\mathbf{g}$. This is needed for stability. Since the flow contains a supersonic region, the eigenvalue $u+c$ (c is the local speed of sound and $u > c$ is assumed) is quite large. Therefore, ρ must be taken quite small. On the other hand, near a sonic region, $u-c$ is quite small. Therefore with this simple smoother, the hybrid multigrid performance in sonic regions can be predicted by the results in Figure 1 with σ near zero. There, the error reduction rate is near unity.

For this reason, we propose characteristic time stepping. This is accomplished by a simple modification of the smoothing iteration above:

$$\epsilon_j^{k+1} = \epsilon_j^k - \rho R(\mathbf{q}_j)|1/\Lambda(\mathbf{q}_j)|L(\mathbf{q}_j)\Big(\frac{1}{\Delta x}(\mathbf{h}^{(lo)}_{\partial_{\mathbf{q}}\mathbf{g}}(\mathbf{q}_{j+1/2};\epsilon_j^k,\epsilon_{j+1}^k) - \mathbf{h}^{(lo)}_{\partial_{\mathbf{q}}\mathbf{g}}(\mathbf{q}_{j-1/2};\epsilon_{j-1}^k,\epsilon_j^k)) + \partial_{\mathbf{q}}\mathbf{s}(\mathbf{q}_j)\epsilon_j^k + (R^{(hi)}(\mathbf{q}))_j\Big),$$

where now, ρ is taken to be $\rho = 0.4\Delta x$.

We implement algorithm (3.2) to this duct test problem, again using 64 grid points on the interval $[0,10]$. Multigrid Step 2 is done using every grid level 2^1, 2^2, 2^3, 2^4, 2^5, 2^6, with $\nu = 4$ and $\eta = 1$. Step 3 is relaxed somewhat, specifically $\mathbf{q}^{k+1} = \mathbf{q}^k + \frac{1}{2}\epsilon_{mg}^k$, because of the presence of the supersonic-subsonic shock. Around the shock, the slope limiting algorithm introduces a small amount of numerical viscosity and this viscosity can introduce an instability into the approximate Newton iteration; (see equation (3.4)). Step 4 of the hybrid algorithm is also utilized with 2 characteristic time stepping iterations applied to the third order residual.

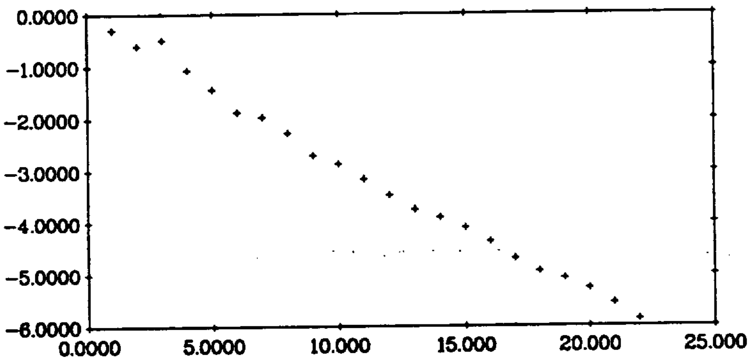

Figure 3. Log residual versus hybrid multigrid cycles, again applied to the duct problem with 64 grid points.

Comparing Figure 2 with Figure 3, we see that the multigrid algorithm reduces the residual of this test problem in 100 times fewer iterations than the artificial time scheme, whereas each multigrid cycle costs on the order of only 4 times the work. Figure 5 depicts the residual reduction from the hybrid multigrid scheme on 5 grids; from level=4 (16 grid points) to level=8 (256 grid points). This last figure demonstrates the virtual grid independence of the hybrid algorithm.

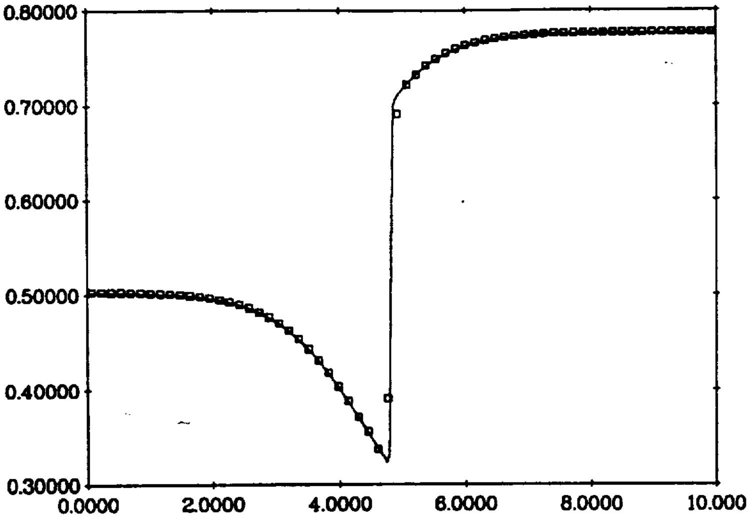

Figure 4. The computed density for the duct test problem using 64 grid points.

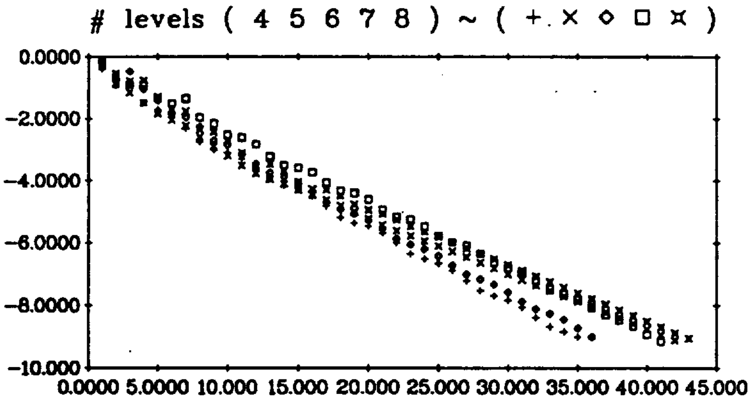

Figure 5. The log residual versus hybrid multigrid cycles for the duct test problem using 16 to 256 grid points.

REFERENCES

[1] P.L. Roe, "Approximate Riemann solvers, parameter vectors, and difference schemes", *J. Comp. Phys.*, 43 (1981), pp. 357-372.

[2] R. Sanders and C.P. Li, "A variation nonexpansive central differencing scheme for nonlinear hyperbolic conservation laws", in Proc. 10th Inter. Conf. Computing Methods in Applied Sciences and Engineering, R. Glowinski ed., Nova Sciences, New York, 1992, pp. 511- 524.

[3] B. Stoufflet, preprint.