

PREPARED FOR THE U.S. DEPARTMENT OF ENERGY,
UNDER CONTRACT DE-AC02-76CH03073

PPPL-3976
UC-70

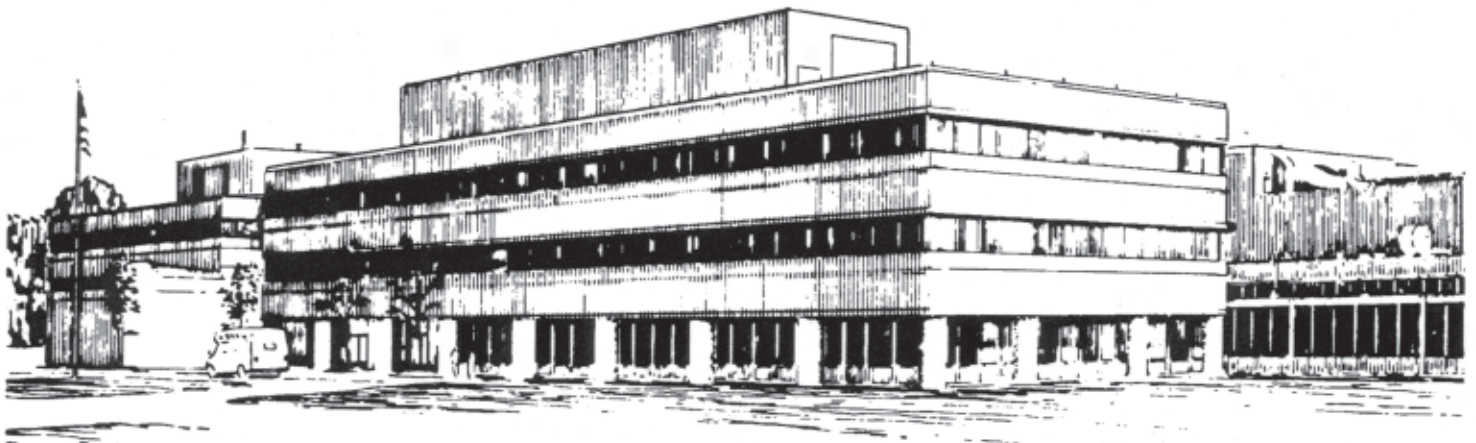
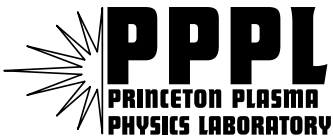
PPPL-3976

Self-correcting Multigrid Solver

by

Jerome L.V. Lewandowski

June 2004



**PRINCETON PLASMA PHYSICS LABORATORY
PRINCETON UNIVERSITY, PRINCETON, NEW JERSEY**

PPPL Reports 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, makes 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.

Availability

This report is posted on the U.S. Department of Energy's Princeton Plasma Physics Laboratory Publications and Reports web site in Fiscal Year 2004. The home page for PPPL Reports and Publications is: http://www.pppl.gov/pub_report/

DOE and DOE Contractors can obtain copies of this report from:

U.S. Department of Energy
Office of Scientific and Technical Information
DOE Technical Information Services (DTIS)
P.O. Box 62
Oak Ridge, TN 37831

Telephone: (865) 576-8401
Fax: (865) 576-5728
Email: reports@adonis.osti.gov

This report is available to the general public from:

National Technical Information Service
U.S. Department of Commerce
5285 Port Royal Road
Springfield, VA 22161

Telephone: 1-800-553-6847 or
(703) 605-6000
Fax: (703) 321-8547
Internet: <http://www.ntis.gov/ordering.htm>

Self-correcting Multigrid Solver

Jerome L. V. Lewandowski
Princeton University
Plasma Physics Laboratory,
Princeton, NJ 08543, USA

June 28, 2004

Abstract

A new multigrid algorithm based on the method of self-correction for the solution of elliptic problems is described. The method exploits information contained in the residual to dynamically modify the source term (right-hand side) of the elliptic problem. It is shown that the self-correcting solver is more efficient at damping the short wavelength modes of the algebraic error than its standard equivalent. When used in conjunction with a multigrid method, the resulting solver displays an improved convergence rate with no additional computational work.

1 Introduction

A very common approach for solving elliptic partial differential equations (PDEs) is often based on iterative solvers. It is well known that basic iterative methods (weighted Jacobi method, Gauss-Seidel method, etc.) for solving elliptic PDEs are very efficient at damping the short wavelength error components but converge poorly for the long wavelength error components [1, 2]. The poor convergence of basic iterative solvers can be mitigated by considering the original problem on a set of overlapping grids, as in the multigrid method [3]. The fundamental idea behind the multigrid method is to solve modified problems using different scales (grids) in order to suppress error components of different scales. The main requirement of the basic iterative solver for the multigrid method is that it must be efficient at damping the short wavelength components of the error. In this paper, we present a self-correcting iterative solver that uses the information in the residual to improve the damping rate of the short wavelength components of the error. The self-correcting iterative method can be integrated in a multigrid solver resulting in an improved convergence rate. Although the specific examples reported in this paper are for one-dimensional problems using the weighted Jacobi method, the method is quite general and can be applied to multi-dimensional elliptic problems and to various basic iterative solvers [4] such as the Gauss-Seidel method, the red-black Gauss-Seidel method, etc.

2 Self-correcting Iterative Solver

In this section, we present the basic algorithm of the self-correcting iterative solver. A numerical example for a one-dimensional Poisson equation based on the weighted Jacobi method is presented. The damping properties of the self-correcting iterative solver for this specific numerical example are obtained and discussed.

2.1 Preliminary Remarks

For illustrative purposes, consider the model elliptic problem

$$L(f) = S, \quad (1)$$

where $L(\bullet)$ is a *negative* definite elliptic operator (*e.g.* $L = \nabla^2$) and S is a known source term. Here f denotes the exact solution; in the remaining of this paper, F refers to the approximate solution. In practise, Eq.(1) can be solved using an iterative method; since the resulting solution is approximate, the local residual

$$r = S - L(F)$$

is, in general, nonzero. The residual actually contains information about the approximate solution and the iteration solver itself that can be exploited; one such method is the multigrid method which updates both the approximate solution and the algebraic error, $e \equiv f - F$, (which itself is related to the residual) through a set of nested (overlapping) grids. The method presented here is, however, based on the original grid; the method can be used in conjunction (for example as a preconditioner) with other more effective methods such as the multigrid method. This approach is discussed and tested in Section 3.

2.2 Basic Algorithm

The basic idea behind the self-correcting iterative solver is to exploit the information contained in the residual to correct the right-hand side of Eq.(1). Specifically, we introduce the intermediary quantity Q which is governed by the following equation

$$\frac{dQ}{dt} = r, \quad (2)$$

together with the auxiliary condition

$$Q = 0. \quad (3)$$

Here t is a time-like variable (continuous iteration parameter). Using an explicit time discretization, Eqs.(2,3) can be written as

$$Q^{(n+1)} - Q^{(n)} = \Delta t \left[S - L \left(F^{(n)} \right) \right] \equiv \Delta t r^{(n)} \quad (4)$$

and

$$Q^{(n+1)} = 0, \quad (5)$$

where n labels the iteration number and Δt denotes the time step. Using the auxiliary condition (5) in Eq.(4), one can write Eq.(1) as

$$L \left(F^{(n)} \right) = S + \nu Q^{(n)} \quad (6)$$

where $\nu \equiv 1/\Delta t$ is a free parameter. Note that the elliptic problem given by Eq.(6) is identical to the original problem, Eq.(1), except for the modified source term, $S \leftarrow S + \nu Q^{(n)}$ (a left arrow denotes replacement). As it is apparent from Eqs.(4,5), the modified source term in Eq.(6) depends on the history of the solution. The basic algorithm for the self-correcting iterative solver is as follows: starting with the initial guess $Q^{(0)} = 0$, one has

$$\begin{aligned} L \left(F^{(n)} \right) &= S + \nu Q^{(n)} \\ r^{(n)} &= S - L \left(F^{(n)} \right) \\ Q^{(n)} &\leftarrow Q^{(n)} + \nu^{-1} r^{(n)} \end{aligned} \quad (7)$$

for $n = 0, 1, \dots$. The above algorithm requires only one additional vector (or matrix) for the storage of Q since the last two steps of algorithm (7) can be cast in the form

$$Q^{(n)} \Leftarrow Q^{(n)} + \nu^{-1} \left[S - L \left(F^{(n)} \right) \right] .$$

As it turns out, the convergence properties of the self-correcting algorithm are weakly depend on the parameter ν . This parameter can be computed dynamically as follows. Writing Q_{n+1} as a Taylor expansion around Q_n and using Eq.(2) one has

$$Q_{n+1} = Q_n + \dot{Q}_n \Delta + \ddot{Q}_n \frac{(\Delta t)^2}{2} + \dots = Q_n + r_n \Delta t + \mathcal{O}((\Delta t)^2) ,$$

where a dot denotes a derivative with respect to time. Introducing a convenient (discrete) scalar product, $\langle \bullet, \bullet \rangle$, we demand that the values of Q between consecutives iterations to be orthogonal which yields (neglecting second-order corrections)

$$\nu_n = \left| \frac{\langle Q_n, r_n \rangle}{\langle Q_n, Q_n \rangle} \right| . \quad (8)$$

The absolute value sign is required to ensure stability.

2.3 Numerical Example

The algorithm described in the previous section is quite general. In order to appreciate the usefulness of the self-correcting iterative solver, we consider the one-dimensional Poisson equation

$$\frac{d^2 f}{dx^2} = S(x) , \quad (9)$$

on the unit interval $x \in [0, 1]$ with boundary conditions $f(0) = f(1) = 0$. We consider 2 different source terms. The first source term (referred to as case **1**) is given by

$$S(x) = 2(1-x) [(1-x)(1-5x) - x(2-5x)] \quad (10)$$

which corresponds to a smooth exact solution of the form

$$f(x) = x^2 (1-x)^3 . \quad (11)$$

The second source term includes a short wavelength contribution (case **2**)

$$S(x) = 2\pi m \epsilon (1-2x) \cos \theta - 2(1 + \epsilon \sin \theta) - m^2 \pi^2 \epsilon x (1-x) \sin \theta \quad (12)$$

where $\theta = m\pi x$, $m \gg 1$ (integer) is called the mode number. The corresponding exact solution is

$$f(x) = x(1-x) [1 + \epsilon \sin \theta] . \quad (13)$$

Here the parameter ϵ denotes the amplitude of the short wavelength contribution. The basic iterative solver used to solve Eq.(9) is the damped Jacobi solver. The boundary conditions are $f(0) = f(1) = 0$. A uniform grid $x_j = j\Delta x$ is setup with spacing $\Delta x = 1/N_g$ (the label j runs from 0 to N_g). The explicit form of the self-correcting iterative solver is

$$\begin{aligned} \tilde{S}_j &= S_j + \nu Q_j \quad (\forall j \in J) \quad \text{step 1} \\ \hat{F}_j &= \frac{F_{j+1} + F_{j-1} - (\Delta x)^2 \tilde{S}_j}{2} \quad (\forall j \in J) \quad \text{step 2} \\ F_j &\Leftarrow (1 - \omega) F_j + \omega \hat{F}_j \quad (\forall j \in J) \quad \text{step 3} \\ r_j &= S_j - \frac{F_{j+1} - 2F_j + F_{j-1}}{(\Delta x)^2} \quad (\forall j \in J) \quad \text{step 4} \\ Q_j &\Leftarrow Q_j + \nu^{-1} r_j \quad (\forall j \in J) \quad \text{step 5} \end{aligned}$$

where $J = \{1, 2, \dots, N_g - 1\}$ (interior nodes). The correction step in the above algorithm are steps **4** and **5**; these steps are carried out every P time steps. All the simulations were performed with a damping parameter of $\omega = 2/3$, $\nu = 1$ (the results are weakly affected by a variation of this parameter) and $N = 1024$. For case **2**, the mode number used is $m = 14$ and the amplitude of the fast mode is $\epsilon = 0.25$. In all simulations the initial profile is $F = 0$.

Figure 1 shows the L^2 norm of the residual as a function of the iteration number for the standard Jacobi method (dashed line) and for the self-corrected Jacobi method (plain line); the corrective steps have been carried out for each cycle ($P = 1$). The oscillatory nature of the residual for the self-corrected case is due to the fact that the corrected source term actually records the history of the system (see analysis in the next section). If the corrective steps are only implemented every $P = 10$ cycles, the period of the oscillations decrease, as can be seen in Figure 2. For $P = 100$, the oscillation period decreases even further (Figure 3). The most interesting aspect of Figure 1 is that the L^2 norm of the residual for the self-corrected case drops sharply after a few iterations. The first minima of $\|r\|_2$ occurs at $n = n_c = 353$ in Figure 1; at $n = n_c$, we have $\|r\|_2 = 0.3452$. In comparison, $N_{it} = 12512$ standard Jacobi iterations would be required to reach the same value of $\|r\|_2$. Although the self-correcting Jacobi solver requires the evaluation of the residual, the overall computational efficiency of the scheme is clearly much better than that of the standard Jacobi solver. Comparing the central process unit (CPU) time, it is found that the self-corrected is approximately 23 times more efficient. As it is evident from Figures (1-3), the location of the first minima is dependent on the value of P used. The table below shows that the values of n_c , $\|r\|_2$ and N_{it} are weakly depend on P . Therefore, the value of $P = 1$ is sufficient for most applications.

P	n_c	CPU_{sc}	$\ r\ _2(n_c)$	N_{it}	CPU
1	353	0.181	0.3452	12512	4.128
2	488	0.213	0.3399	13319	4.407
3	591	0.241	0.3365	13857	4.513
4	680	0.264	0.3338	14298	4.638
5	755	0.285	0.3317	14648	4.756

Table 1: Comparison of the CPU time for the standard Jacobi solver (CPU) and the self-correcting Jacobi solver (CPU_{sc}).

Figure 4 shows the exact solution (plain line) and the approximate solutions (dotted line: standard Jacobi solver; dashed line: self-correcting Jacobi solver) at $n = n_c$ (first minima of $\|r\|_2$ in Figure 1). Clearly the self-correcting Jacobi method is able to resolve the long-wavelength modes associated with the solution whereas the standard Jacobi does not. Figure 5 and Figure 6 show the L^2 norm of the residual as a function of the iteration number for case **1** and case **2**, respectively. We note the first minima of $\|r\|_2$ for the case **2** clearly stands out. Figure 7 shows the exact solution (plain line) and the approximates solutions (dotted line: standard Jacobi solver; dashed line: self-correcting Jacobi solver) at $n = n_c$ (first minima of $\|r\|_2$ in Figure 6) for case **2**. Here the self-correcting iterative solver captures the high-frequency modes in some detail but fails to resolve the envelope. However the standard Jacobi solver performs very poorly for all the modes. Figure 6 shows that the self-correcting Jacobi solver is actually bad when the number of iterations is large. However, our goal is to find a basic solver (smoother) for multigrid algorithm that is efficient at damping the short-wavelength modes in the error in a few iterations. As shown in the next section, the self-correcting Jacobi solver is indeed very efficient at damping the high-frequency modes of the error.

The standard Jacobi solver requires $4N'(N_g - 1)$ multiplications and $3N'(N_g - 1)$ additions/subtractions, where the total number of grid points is $N_g + 1$ (including 2 boundary nodes) and N' is the number of relaxation sweeps. The self-correcting Jacobi solver requires $M(4N + 1)(N_g - 1)$ multiplications and $M(3N + 4)(N_g - 1)$ additions/subtractions, where N is the number of Jacobi relaxation sweeps and M is the number of self-correcting steps. The self-correcting solver requires a smaller number of relaxation sweeps compared to the standard solver since it is more efficient at damping the short-wavelength modes of the error ($N \ll N'$). When the parameters N' (for the standard solver) and N and M (for the self-correcting solver) are chosen such that the computational work is comparable for the 2 solvers, the self-correcting multigrid solver displays an improved convergence rate (section 3).

2.4 Damping Properties of the Self-correcting Jacobi Solver

In this section, we study the damping properties of a one-dimensional Poisson equation using the self-correcting Jacobi solver. In view of the linearity of the problem, the mode amplitudes can be determined exactly. We consider the one-dimensional Poisson equation

$$\frac{d^2 f}{dx^2} = S(x), \quad (14)$$

on the unit interval with boundary conditions $f(0) = f(1) = 0$. The source term is chosen such as to satisfy $S(0) = S(1)$ in which case it can be written as a sine Fourier series

$$S(x) = \sum_{\ell} S_{\ell} \sin(\ell\pi x), \quad (15)$$

where

$$S_{\ell} = 2 \int_0^1 S(x) \sin(\ell\pi x) dx. \quad (16)$$

The exact solution of Eq.(14) can then be written as a sine Fourier series $f(x) = \sum_{\ell} f_{\ell} \sin(\ell\pi x)$ with $f_{\ell} = -S_{\ell}/(\ell^2\pi^2)$. If the initial profile is chosen to be of the form

$$F^{(0)} = \sum_{\ell} F_{\ell}^{(0)} \sin(\ell\pi x),$$

then the amplitudes of each Fourier modes after n self-correcting steps (with N relaxation sweeps for each Jacobi solve) are given by (see Appendix)

$$F_{\ell}^{(n)} = \mathcal{A}_{\ell}^{(n)} F_{\ell}^{(0)} + \mathcal{B}_{\ell}^{(n)} S_{\ell} \quad (17)$$

where $\mathcal{A}_{\ell}^{(1)} = \xi_{\ell}^N$, $\mathcal{B}_{\ell}^{(1)} = -\alpha\mu_{\ell}$, $\mathcal{A}_{\ell}^{(2)} = \xi_{\ell}^N \varphi_{\ell}$, $\mathcal{B}_{\ell}^{(2)} = -\alpha\mu_{\ell}(1 + \eta + \varphi_{\ell})$ and

$$\left. \begin{aligned} \mathcal{A}_{\ell}^{(n)} &= \varphi_{\ell} \mathcal{A}_{\ell}^{(n-1)} - \eta\alpha\mu_{\ell}\lambda_{\ell} \sum_{k=1}^{n-2} \mathcal{A}_{\ell}^{(k)} \\ \mathcal{B}_{\ell}^{(n)} &= \varphi_{\ell} \mathcal{B}_{\ell}^{(n-1)} - \eta\alpha\mu_{\ell}\lambda_{\ell} \sum_{k=1}^{n-2} \mathcal{B}_{\ell}^{(k)} - \alpha\mu_{\ell}(1 + \eta(n-1)) \end{aligned} \right\} \text{for } n = 3, 4, \dots, M \quad (18)$$

where $\alpha = \omega(\Delta x)^2/2$, $\xi_{\ell} = 1 - \omega + \omega \cos(\ell\pi\Delta x)$, $\lambda_{\ell} = 2(\Delta x)^{-2} [1 - \cos(\ell\pi\Delta x)]$, $\mu_{\ell} = \sum_{k=0}^{N-1} \xi_{\ell}^k = (1 - \xi_{\ell}^N)/(1 - \xi_{\ell})$ and $\varphi_{\ell} = \xi_{\ell}^N - \eta\alpha\mu_{\ell}\lambda_{\ell}$. Here the parameter η is unity for the self-correcting solver and $\eta = 0$ for the standard Jacobi solver. Eqs.(17,18) show that the

mode amplitude at iteration n is a weighted average of the history of that mode. For the standard Jacobi solver ($\eta = 0$), Eq.(18) is simply

$$\mathcal{A}_\ell^{(n)} = \mathcal{B}_\ell^{(n)} = \xi_\ell^{N'} , \quad (19)$$

where $N' = nN$. Clearly the amplitudes of the short wavelength modes are strongly dependent on the value of $\xi_\ell \simeq 1 - \omega$. The self-correcting solver damps the short wavelength modes faster than the standard Jacobi solver due to a combination of two factors: first the quantity φ_ℓ is smaller than ξ_ℓ ; second, the term $\alpha\mu_\ell\lambda_\ell$ in Eq.(18) is a positive definite quantity and as a result both $\mathcal{A}_\ell^{(n)}$ and $\mathcal{B}_\ell^{(n)}$ are reduced.

Figure 8 shows the mode amplitude as a function of the mode number ℓ for the standard Jacobi solver (plain line; $N' = 4$) and the self-correcting Jacobi solver (dotted line; $M = N = 2$). The initial profile Eq. (17) contains 16 modes. The number of grid points is $N_g = 1024$ and the damping parameter is $\omega = 0.5$. As it is evident from Figure 8, the short wavelength modes are more damped for the self-correcting solver compared to the standard solver. This suggests the use of the self-correcting solver as the basic solver (smoother) in multigrid algorithms. This approach is discussed in the next section.

3 Self-correcting Multigrid Algorithm

As described in the previous sections, the self-correcting iterative solver is efficient at damping the short wavelength components of the error. This suggests that the self-correcting algorithm is used as the basic solver (smoother) in a multigrid algorithm. For illustrative purposes, the case of the V cycle algorithm [3,5] is discussed here although the basic method can be extended to other multigrid algorithms.

3.1 Self-correcting V Cycle

As is well known the fundamental idea behind the multigrid method is to solve modified problems using different scales (grids) in order to suppress error components of different scales. Problem (1) can be defined on a set of overlapping grids (or levels) as

$$L^{(p)}F^{(p)} = S^{(p)} ,$$

where $p = 0, 1, \dots, Q$ labels the level and Q is the total number of levels compatible with the original number of grid points N_g . The algebraic error, $e \equiv f - F$, and the residual, $r = S - LF$, are related through the residual equation $Le = r$. If $A^{(p)}$ denotes an approximation to the inverse of $L^{(p)}$, then the approximate solution on grid level p is

$$F^{(p)} = A^{(p)} \left(S^{(p)}; F_0^{(p)}; \nu_p \right) ,$$

where $F_0^{(p)}$ denotes the initial guess and ν_p (integer) is the number of relaxation sweeps. Taking into account that the original problem [Eq.(1)] and the residual equation are of the same form, the multigrid V cycle can be cast in algorithmic form as

MultiGrid V cycle

$$\left. \begin{aligned} F^{(p)} &= A^{(p)} \left(S^{(p)}; F_0^{(p)}; \nu_p \right) \\ r^{(p)} &= S^{(p)} - L^{(p)}F^{(p)} \\ S^{(p+1)} &= I(p \mapsto p+1) r^{(p)} \end{aligned} \right\} p = 0, 1, \dots, Q-1$$

$$\left. \begin{aligned} F^{(p)} &= A^{(p)} \left(S^{(p)}; F_0^{(p)}; \nu_p \right) \\ F_0^{(p)} &= F^{(p)} + I(p+1 \mapsto p) F^{(p+1)} \end{aligned} \right\} p = Q, Q-1, \dots, 1$$

Here we have defined the intergrid transfer operators from coarse grid to fine grid, $I(p+1 \mapsto p)$ (prolongation operator), and from fine grid to coarse grid, $I(p \mapsto p+1)$ (restriction operator). The basic solver (smoother) on grid $\Omega^{(p)}$ can be cast in algorithmic form as

Basic Solver ($\Omega^{(p)}$)

$$\left. \begin{aligned} r^{(p)} &= S^{(p)} - L(F^{(p)}) \\ Q_\ell^{(p)} &= Q_{\ell-1}^{(p)} + \nu^{-1} r^{(p)} \\ \widehat{S}_\ell^{(p)} &= S^{(p)} + \nu Q_\ell^{(p)} \\ L^{(p)}(F^{(p)}) &= \widehat{S}_\ell^{(p)} \quad (N \text{ relaxations}) \end{aligned} \right\} \ell = 1, \dots, M \quad (20)$$

with the initial condition $Q_0^{(p)} = 0$. As in the previous section, a weighted Jacobi solver is used to perform the relaxation of $L^{(p)}(F^{(p)}) = \widehat{S}_\ell^{(p)}$. Note that the algorithm (20) performs a total of MN Jacobi relaxation sweeps on each grid $\Omega^{(p)}$.

3.2 Numerical Example

In this section, we compare the performance of the self-correcting multigrid solver with the standard multigrid solver. Although the numerical example presented here is specific to a one-dimensional problem, the advantage of the self-correcting multigrid solver extends to multi-dimensional problems as well as other smoothers such as the Gauss-Seidel solver, the red-black Gauss-Seidel solver, etc. For illustrative purposes, we consider the one-dimensional problem on the unit interval

$$\left(\frac{d^2}{dx^2} + A(x) \frac{d}{dx} + B(x) \right) f(x) = S(x), \quad (21)$$

where $A(x) = x(1-x)$ and $B(x) = \sin(\pi x)$. The boundary conditions are $f(0) = f(1) = 0$. The initial profile is of the form

$$F_0(x) = \sum_{k=1}^{16} \sin(k\pi x). \quad (22)$$

The multigrid algorithm used here is the V cycle with the full weighting scheme as restriction operator and the linear interpolation method as prolongation operator.

Figure 9 shows the L^2 norm of the residual as a function of the number of V cycles for 2 different smoothers; the plain line is for the standard Jacobi solver ($N' = 4$ relaxation sweeps on each grid level) whereas the dotted line is for the self-correcting Jacobi solver ($M = N = 2$). The number of grid points is $N_g = 2048$ and the damping parameter is $\omega = 0.5$. The L^2 norm of the residual after 15 consecutive V cycles is approximately 2 orders of magnitude smaller for the self-correcting solver compared with that of the standard solver. Figure 10 shows the convergence rate computed from $\rho = \left(\|r\|_2^{(k)} / \|r\|_2^{(0)} \right)^{1/k}$, where k is the total number of V cycles (here $k = 15$), as a function of the number of grid points for the standard Jacobi solver (plain line) and for the self-correcting Jacobi solver (dotted line). The initial profile and other parameters are the same as those of Figure 9. For a sufficiently large number of grid points the convergence rate is independent of N_g (multigrid convergence). The convergence rate for the self-correcting multigrid solver is about one third smaller than that for the standard multigrid solver although the computational work of both solvers are comparable.

4 Conclusion

We have introduced the self-correcting iterative solver for elliptic problems $L(f) = S$, where S is the source term. The basic idea behind this solver is to exploit the information contained in the (instantaneous) residual by dynamically modifying the source term S . A simple mode analysis has shown that the self-correcting solver damps the short-wavelength modes of the algebraic error more efficiently as compared to the standard solver. This property suggests to use the self-correcting solver as a basic smoother for multigrid algorithms. The self-correcting multigrid solver has a better convergence rate than its standard equivalent for a comparable computational work.

Acknowledgments

This research was supported by Contract No DE-AC02-76CH03073 and the Scientific Discovery through Advanced Computing (SciDAC) initiative (U.S. Department of Energy).

References

- [1] F. De la Vallee Poussin, *SIAM J. of Num. Analysis*, **5**(2), 340 (1968).
- [2] A. Settari and K. Aziz, *SIAM J. of Num. Analysis*, **10**(3), 506 (1973).
- [3] P. Wesseling, *An Introduction to Multigrid Methods* (Wiley, Chichester, 1992).
- [4] W.S Press, S.A. Teukolsky, W.T. Vetterling and B.P. Flannery, *Numerical Recipes in Fortran* (Cambridge University Press, New York, 1992).
- [5] J.L.V. Lewandowski, *Phys. of Plasmas*, **10**, 3204 (2003).

Appendix: Damping properties of the self-correcting Jacobi Solver

In this appendix, we consider a one-dimensional elliptic equation on the unit interval

$$\frac{d^2 f}{dx^2} = S(x), \quad (23)$$

with a source term given by

$$S(x) = \sum_{\ell} S_{\ell} \sin(\ell\pi x). \quad (24)$$

As before, f denotes the exact solution, whereas F denotes the approximate solution. The exact solution of Eq.(23) can then be written as

$$f(x) = \sum_{\ell} f_{\ell} \sin(\ell\pi x) \quad (25)$$

where $f_{\ell} = -S_{\ell}/(\ell^2\pi^2)$, $\forall \ell$. The initial profile is taken to be of the form

$$F_{\text{initial}}(x) = g(x) = \sum_{\ell} A_{\ell} \sin(\ell\pi x). \quad (26)$$

We first consider the standard weighted Jacobi solver for the discrete version of Eq.(23)

$$F_{j+1} - 2F_j + F_{j-1} = (\Delta x)^2 S_j \quad (j = 1, \dots, N_g - 1)$$

where $\Delta x = 1/N_g$ is the (uniform) grid spacing. The algorithm for the standard weighted Jacobi solver is

$$\left. \begin{aligned} \widehat{F}_j^{(q)} &= \frac{1}{2} \left(F_{j+1}^{(q)} + F_{j-1}^{(q)} \right) - \frac{(\Delta x)^2}{2} S_j \\ F_j^{(q+1)} &= (1 - \omega) F_j^{(q)} + \omega \widehat{F}_j^{(q)} \end{aligned} \right\} q = 0, \dots, N - 1 \quad (27)$$

where N is the total number of relaxation sweeps. For the given initial profile, we obtain (in Fourier space) the relations of

$$\left. \begin{aligned} \widehat{F}_{\ell}^{(q)} &= \cos(\ell\pi\Delta x) F_{\ell}^{(q)} - \frac{(\Delta x)^2}{2} S_{\ell} \\ F_{\ell}^{(q)} &= (1 - \omega) F_{\ell}^{(q)} + \omega \widehat{F}_{\ell}^{(q)} \end{aligned} \right\} \quad (28)$$

The approximate solution, $F(x) = \sum_{\ell} F_{\ell} \sin(\ell\pi x)$, after N Jacobi relaxation sweeps is determined by using Eqs.(28) in the algorithm (27); the amplitudes, F_{ℓ} , are given by

$$F_{\ell} = \xi_{\ell}^N F_{\ell}^{(0)} - \alpha \mu_{\ell} S_{\ell}, \quad (29)$$

where $F_{\ell}^{(0)} = A_{\ell}$ (see Eq.(26)), $\alpha = \omega(\Delta x)^2/2$, $\xi_{\ell} = 1 - \omega + \omega \cos(\ell\pi\Delta x)$ and $\mu_{\ell} = \sum_{k=0}^{N-1} \xi_{\ell}^k = (1 - \xi_{\ell}^N)/(1 - \xi_{\ell})$. Clearly the short wavelength modes of F (large ℓ) are damped faster than the long wavelength modes since ξ_{ℓ} is smaller for these modes. We now consider the damping properties of the self-correcting Jacobi solver. The algorithm for the self-correcting Jacobi solver is

$$\left. \begin{aligned} Q^{(n)} &= Q^{(n-1)} + \nu^{-1} r^{(n-1)} \\ L(F^{(n)}) &= S + \eta\nu Q^{(n)} = S^{(n)} \quad (N \text{ Jacobi relaxations}) \\ r^{(n)} &= S - L(F^{(n)}) \end{aligned} \right\} n = 1, \dots, M$$

with $Q^{(0)} = r^{(0)} = 0$. Here η is a switch such that: $\eta = 0$, for the standard Jacobi solver; and $\eta = 1$, for the self-correcting Jacobi solver. Using Eq.(29), we have the following relations (in Fourier space)

$$\left. \begin{aligned} Q_\ell^{(n)} &= Q_\ell^{(n-1)} + \nu^{-1} r_\ell^{(n-1)} \\ S_\ell^{(n)} &= S_\ell + \eta \nu Q_\ell^{(n)} \\ F_\ell^{(n)} &= \xi_\ell^N F_\ell^{(n-1)} - \alpha \mu_\ell S_\ell^{(n)} \\ r_\ell^{(n)} &= S_\ell + \lambda_\ell F_\ell^{(n)} \end{aligned} \right\} \quad (30)$$

for $n = 1, \dots, M$ and $\lambda_\ell = 2(\Delta x)^{-2} (1 - \cos(\ell\pi\Delta x))$. Using the first 2 relations in Eq.(30) it is easy to show that the amplitudes of the modified source term, $S_\ell^{(n)}$, depend on the ‘history’ of the amplitudes of the residual since

$$S_\ell^{(n)} = S_\ell + \eta \sum_{k=1}^{n-1} r_\ell^{(k)} \quad (n > 1).$$

Solving the linear system of equations (30) for the amplitudes of F , one obtains

$$F_\ell^{(n)} = \mathcal{A}_\ell^{(n)} F_\ell^{(0)} + \mathcal{B}_\ell^{(n)} S_\ell \quad (31)$$

where $\mathcal{A}_\ell^{(1)} = \xi_\ell^N$, $\mathcal{B}_\ell^{(1)} = -\alpha \mu_\ell$, $\mathcal{A}_\ell^{(2)} = \xi_\ell^N \varphi_\ell$, $\mathcal{B}_\ell^{(2)} = -\alpha \mu_\ell (1 + \eta + \varphi_\ell)$ and

$$\left. \begin{aligned} \mathcal{A}_\ell^{(n)} &= \varphi_\ell \mathcal{A}_\ell^{(n-1)} - \eta \alpha \mu_\ell \lambda_\ell \sum_{k=1}^{n-2} \mathcal{A}_\ell^{(k)} \\ \mathcal{B}_\ell^{(n)} &= \varphi_\ell \mathcal{B}_\ell^{(n-1)} - \eta \alpha \mu_\ell \lambda_\ell \sum_{k=1}^{n-2} \mathcal{B}_\ell^{(k)} - \alpha \mu_\ell (1 + \eta(n-1)) \end{aligned} \right\} \text{for } n > 2 \quad (32)$$

We have defined $\varphi_\ell = \xi_\ell^N - \eta \alpha \mu_\ell \lambda_\ell$.

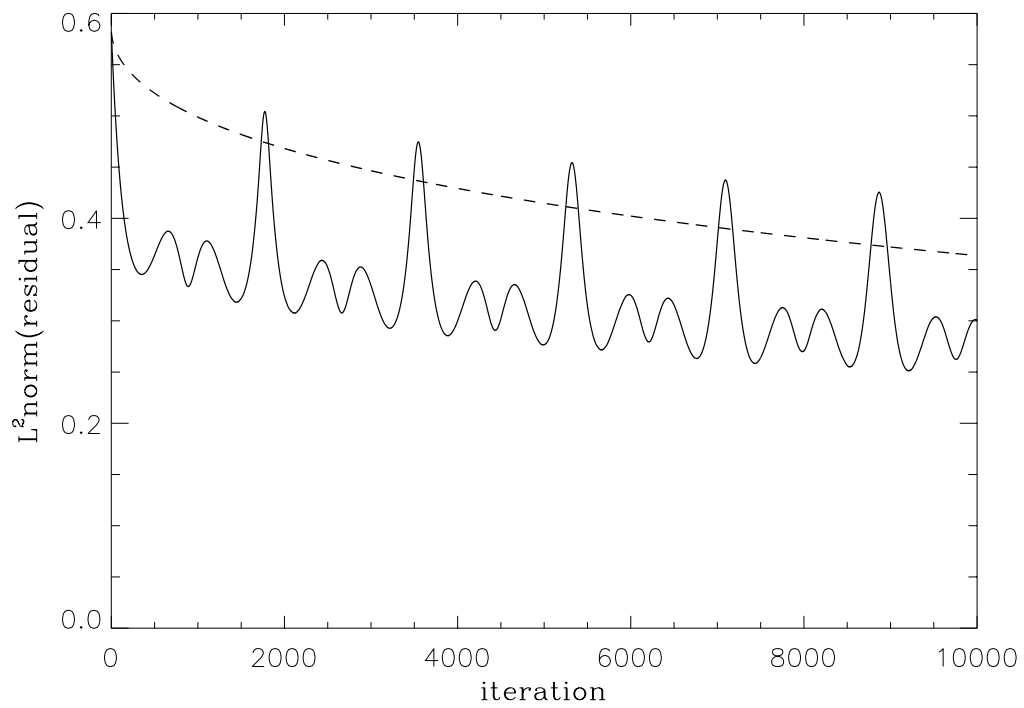


Figure 1: L^2 of the residual as function of the iteration number for case 1 (dashed line: standard Jacobi solver; plain line: self-correcting Jacobi solver). The corrective steps are implemented at each cycle ($P = 1$).

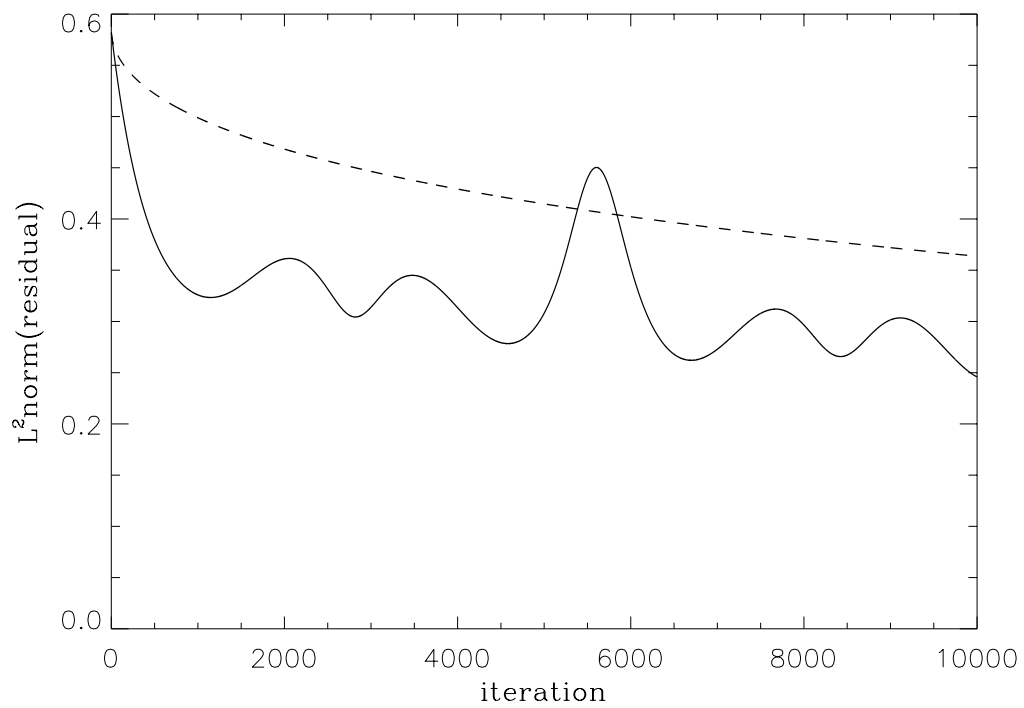


Figure 2: L^2 of the residual as function of the iteration number for case 1 (dashed line: standard Jacobi solver; plain line: self-correcting Jacobi solver). The corrective steps are implemented every $P = 10$ cycle.

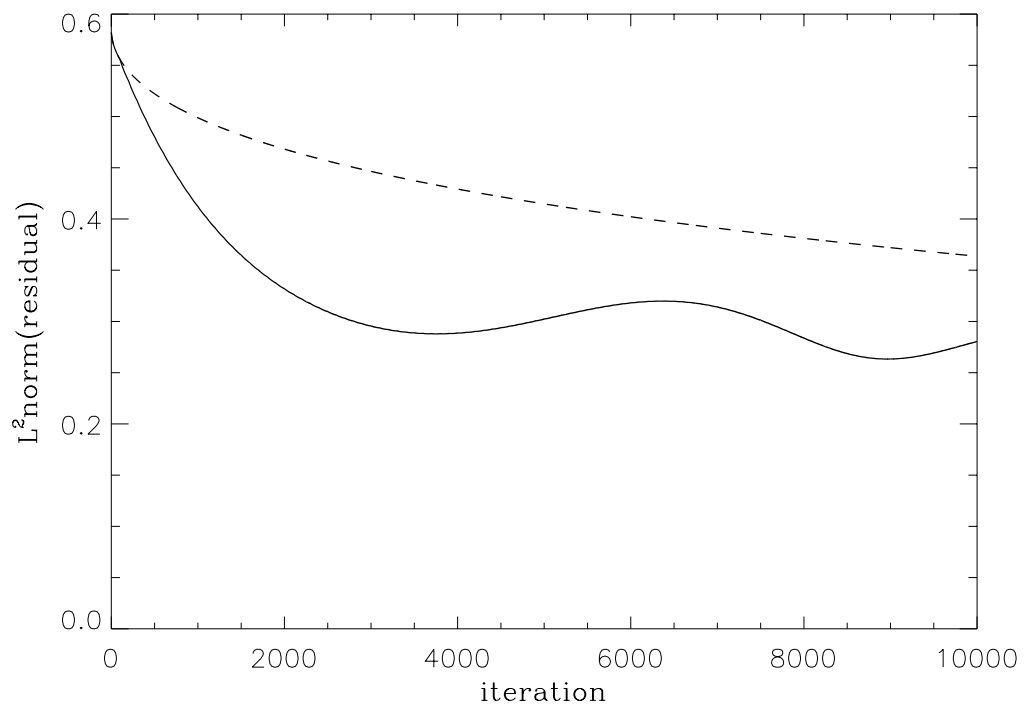


Figure 3: L^2 of the residual as function of the iteration number for case 1 (dashed line: standard Jacobi solver; plain line: self-correcting Jacobi solver). The corrective steps are implemented every $P = 100$ cycle.

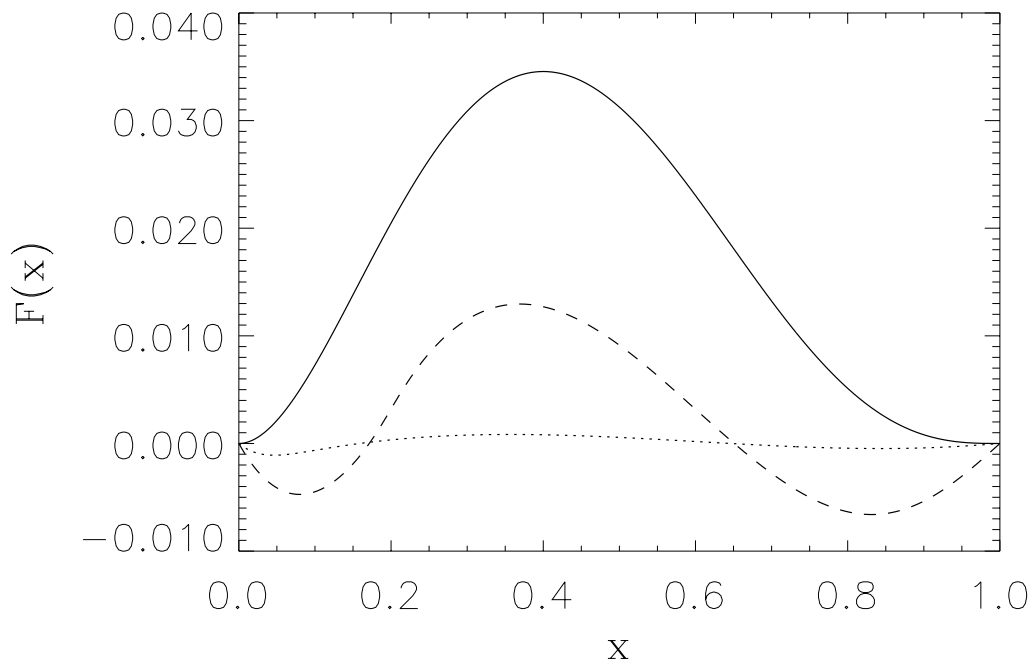


Figure 4: Exact solution for case **1** is shown as a plain line. The approximate solutions at $n = n_c$ (which corresponds to the first minima of $\|r\|_2$ in Figure 1) for the standard Jacobi solver (dotted line) and the self-correcting Jacobi solver (dashed line) are also shown.

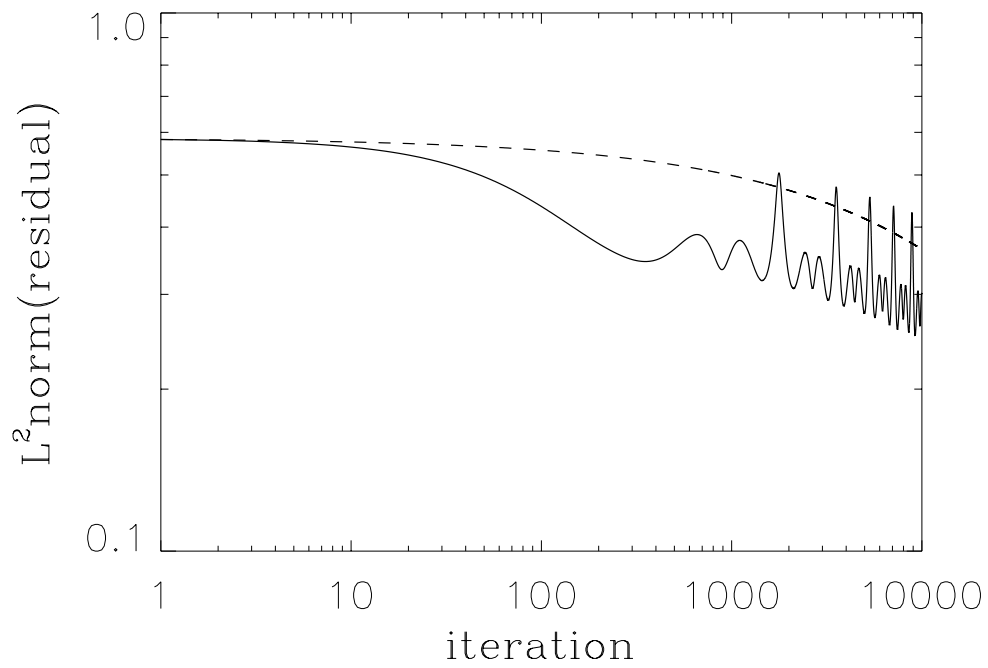


Figure 5: L^2 of the residual as function of the iteration number for case 1 (dashed line: standard Jacobi solver; plain line: self-correcting Jacobi solver). The corrective steps are implemented at each cycle ($P = 1$).

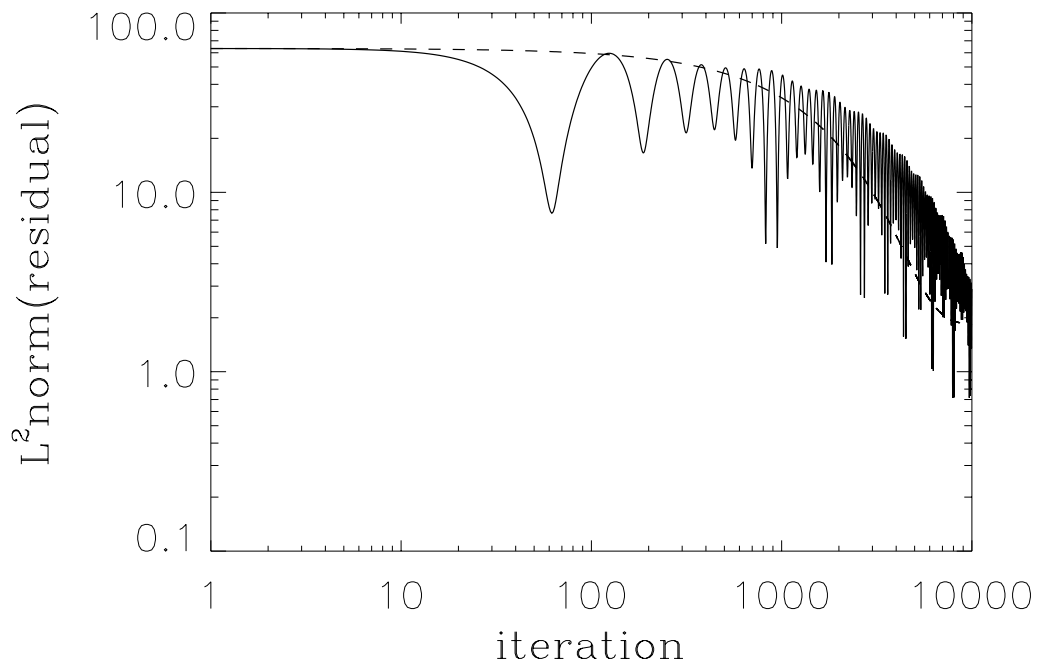


Figure 6: L^2 of the residual as function of the iteration number for case 2 (dashed line: standard Jacobi solver; plain line: self-correcting Jacobi solver). The corrective steps are implemented at each cycle ($P = 1$).

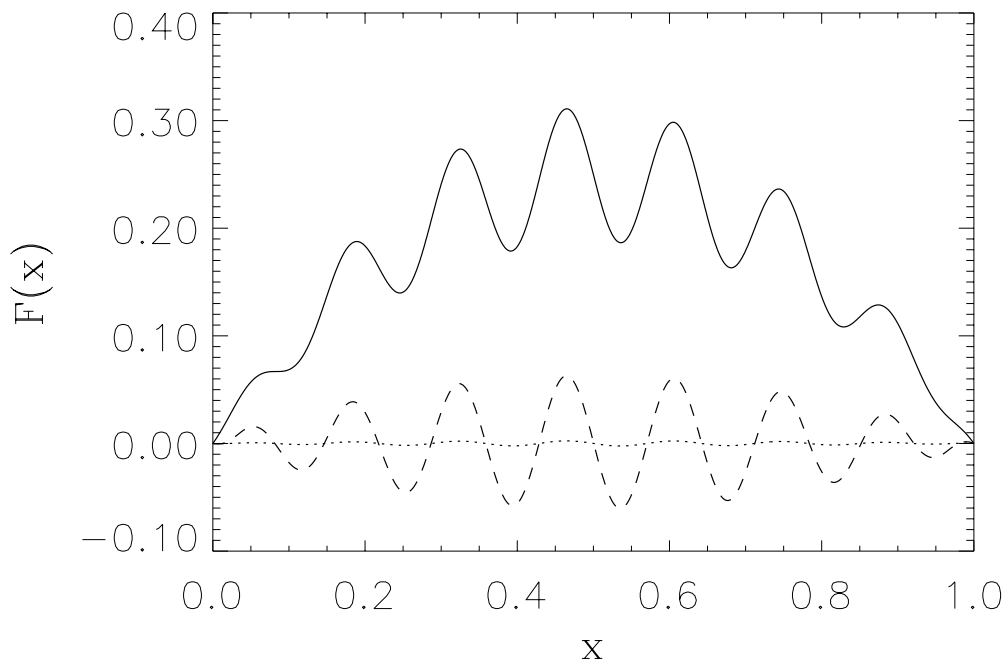


Figure 7: Exact solution for case **2** is shown as a plain line. The approximate solutions at $n = n_c$ (which corresponds to the first minima of $\|r\|_2$ in Figure 6) for the standard Jacobi solver (dotted line) and the self-correcting Jacobi solver (dashed line) are also shown.

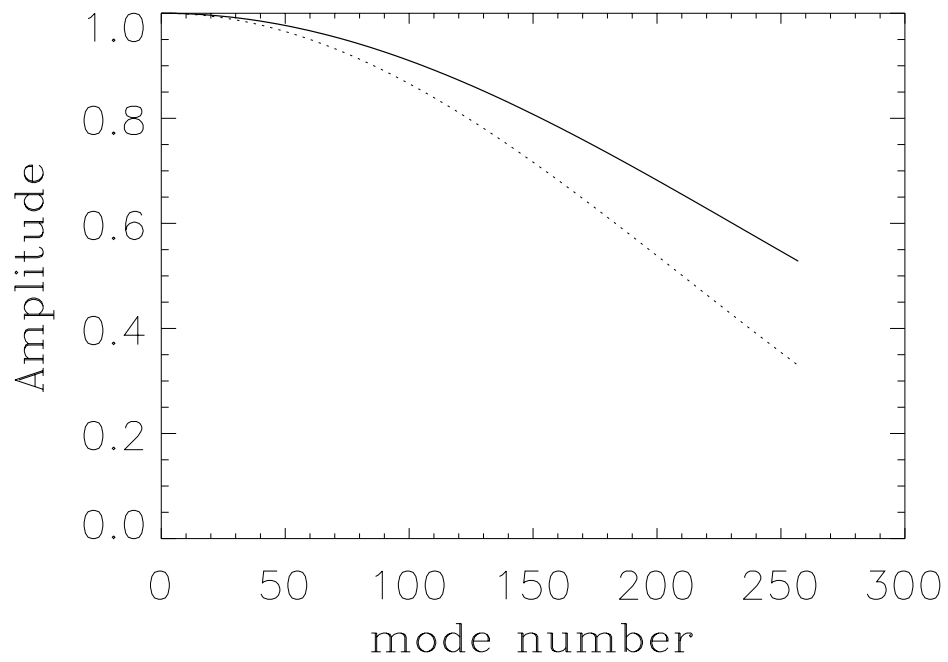


Figure 8: Amplitude of the Fourier modes for the standard Jacobi solver ($N' = 4$ relaxation sweeps) and for the self-correcting solver ($M = 2$ solves with $N = 2$ Jacobi relaxation sweeps per solve). The computational work of the two solvers is comparable.

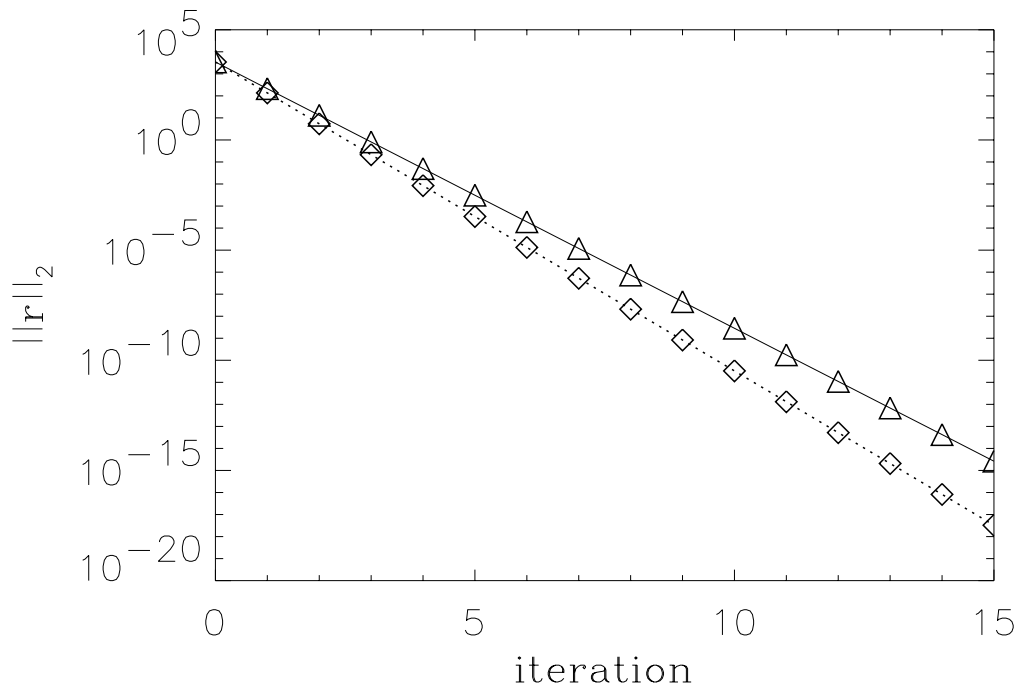


Figure 9: L^2 norm of the residual as a function of the iteration number for the standard Jacobi solver (triangles; plain line) and the self-correcting Jacobi solver (diamonds; dotted line). The number of grid points is $N_g = 2048$, the damping parameter is $\omega = 0.5$ and the initial profile is

$$F_0(x) = \sum_{k=1}^{16} \sin(k\pi x).$$

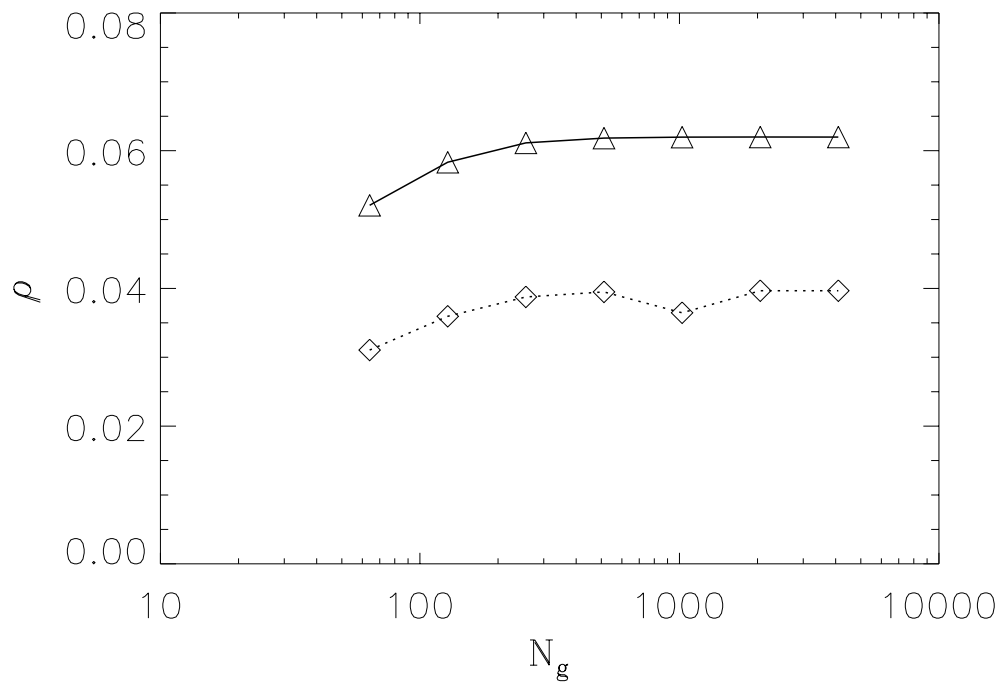


Figure 10: Multigrid convergence rate as a function of the number of grid points for the standard Jacobi solver (triangles; plain line) and the self-correcting Jacobi solver (diamonds; dotted line).

The damping parameter is $\omega = 0.5$ and the initial profile is $F_0(x) = \sum_{k=1}^{16} \sin(k\pi x)$.

External Distribution

Plasma Research Laboratory, Australian National University, Australia
Professor I.R. Jones, Flinders University, Australia
Professor João Canalle, Instituto de Fisica DEQ/IF - UERJ, Brazil
Mr. Gerson O. Ludwig, Instituto Nacional de Pesquisas, Brazil
Dr. P.H. Sakanaka, Instituto Fisica, Brazil
The Librarian, Culham Laboratory, England
Mrs. S.A. Hutchinson, JET Library, England
Professor M.N. Bussac, Ecole Polytechnique, France
Librarian, Max-Planck-Institut für Plasmaphysik, Germany
Jolan Moldvai, Reports Library, Hungarian Academy of Sciences, Central Research Institute
for Physics, Hungary
Dr. P. Kaw, Institute for Plasma Research, India
Ms. P.J. Pathak, Librarian, Institute for Plasma Research, India
Ms. Clelia De Palo, Associazione EURATOM-ENEA, Italy
Dr. G. Grosso, Instituto di Fisica del Plasma, Italy
Librarian, Naka Fusion Research Establishment, JAERI, Japan
Library, Laboratory for Complex Energy Processes, Institute for Advanced Study,
Kyoto University, Japan
Research Information Center, National Institute for Fusion Science, Japan
Dr. O. Mitarai, Kyushu Tokai University, Japan
Dr. Jiengang Li, Institute of Plasma Physics, Chinese Academy of Sciences,
People's Republic of China
Professor Yuping Huo, School of Physical Science and Technology, People's Republic of China
Library, Academia Sinica, Institute of Plasma Physics, People's Republic of China
Librarian, Institute of Physics, Chinese Academy of Sciences, People's Republic of China
Dr. S. Mirnov, TRINITI, Troitsk, Russian Federation, Russia
Dr. V.S. Strelkov, Kurchatov Institute, Russian Federation, Russia
Professor Peter Lukac, Katedra Fyziky Plazmy MFF UK, Mlynska dolina F-2,
Komenskeho Univerzita, SK-842 15 Bratislava, Slovakia
Dr. G.S. Lee, Korea Basic Science Institute, South Korea
Institute for Plasma Research, University of Maryland, USA
Librarian, Fusion Energy Division, Oak Ridge National Laboratory, USA
Librarian, Institute of Fusion Studies, University of Texas, USA
Librarian, Magnetic Fusion Program, Lawrence Livermore National Laboratory, USA
Library, General Atomics, USA
Plasma Physics Group, Fusion Energy Research Program, University of California
at San Diego, USA
Plasma Physics Library, Columbia University, USA
Alkesh Punjabi, Center for Fusion Research and Training, Hampton University, USA
Dr. W.M. Stacey, Fusion Research Center, Georgia Institute of Technology, USA
Dr. John Willis, U.S. Department of Energy, Office of Fusion Energy Sciences, USA
Mr. Paul H. Wright, Indianapolis, Indiana, USA

The Princeton Plasma Physics Laboratory is operated
by Princeton University under contract
with the U.S. Department of Energy.

Information Services
Princeton Plasma Physics Laboratory
P.O. Box 451
Princeton, NJ 08543

Phone: 609-243-2750
Fax: 609-243-2751
e-mail: pppl_info@pppl.gov
Internet Address: <http://www.pppl.gov>