A COMPARISON OF ACCELERATION TECHNIQUES APPLIED TO THE SOR METHOD

RUDNEI DIAS DA CUNHA
Computing Laboratory, University of Kent at Canterbury, U.K.
Centro de Processamento de Dados, Universidade Federal do Rio Grande do Sul, Brasil

and

TIM HOPKINS
Computing Laboratory, University of Kent at Canterbury, U.K.

Abstract. In this paper we investigate the performance of four different SOR acceleration techniques on a variety of linear systems. Two of these techniques have been proposed by Dancis [1] who uses a polynomial acceleration together with a sub-optimal ω. The two other techniques discussed are vector accelerations; the ε algorithm proposed by Wynn [9] and a generalisation of Aitken's Δ^2 algorithm, proposed by Graves-Morris [3].

The experimental results show that these accelerations can reduce the amount of work required to obtain a solution and that their rates of convergence are generally less sensitive to the value of ω than the straightforward SOR method. However a poor choice of ω can result in particularly inefficient solutions and more work is required to enable cheap estimates of a effective parameter to be obtained.

Necessary conditions for the reduction in the computational work required for convergence are given for each of the accelerations, based on the number of floating-point operations.

It is shown experimentally that the reduction in the number of iterations is related to the separation between the two largest eigenvalues of the SOR iteration matrix for a given ω. This separation influences the convergence of all the acceleration techniques above.

Another important characteristic exhibited by these accelerations is that even if the number of iterations is not reduced significantly compared to the SOR method, they are competitive in terms of number of floating-point operations used and thus they reduce the overall computational workload.

Key words: SOR, Acceleration

1. Introduction

Consider the solution of the non-singular, symmetric, positive-definite system of n linear equations

$$Ax = b \tag{1}$$

by the SOR iteration

$$(I + \omega D^{-1} A_L) x^{(k+1)} = \left((1-\omega)I - \omega D^{-1} A_U\right) x^{(k)} + \omega D^{-1} b, \quad k = 0, 1, \ldots \tag{2}$$

where $D = diag(A)$, A_L and A_U are the strictly lower and upper triangular parts of A and ω is the relaxation parameter. Convergence of the method is guaranteed for $0 < \omega < 2$.

The SOR iterative method is commonly used for the solution of large, sparse, linear systems that arise from the approximation of partial differential equations. Its rate of convergence is dependent on the value chosen for the iteration parameter ω. Following Young [5], the minimum number of iterations required for convergence is obtained when this parameter has an optimal value, ω_b, which minimises the spectral radius of the SOR iteration matrix,

$$\mathcal{L}_\omega = (I + \omega D^{-1} A_L)^{-1} \left((1-\omega)I - \omega D^{-1} A_U\right).$$

The value of ω_b may be obtained from the largest eigenvalue, μ_1, of the Jacobi iteration matrix $B = -D^{-1}(A_L + A_U)$ using

$$\omega_b = \frac{2}{1 + \sqrt{1 - \mu_1^2}} \tag{3}$$

However computing ω_b is relatively expensive in most cases. Adaptive procedures exist that can be used to update some initial approximation to ω_b, as in the ITPACK 2C package [4], but some of the initial iterations have large error vectors (when compared to SOR using some $\omega > 1$). In this case some of the initial estimates of the solution are "wasted" during the iterative process and this may be undesirable.

An alternative is to use an acceleration technique that may not be so sensitive to the choice of ω. The first two acceleration techniques, detailed in §2 were proposed by Dancis [1] and follow the usual approach of trying to select some ω for the SOR iteration. We show that for one of his techniques the selection of ω is not as sensitive as expected. In section 3 we look at an extension of the ε algorithm of Wynn applicable to vector and matrices iterations ([8][9]) and in §4 a generalisation of Aitken's Δ^2 algorithm as proposed by Graves-Morris [3] is described. These are *vector* accelerations and thus do not give a prescription for selecting ω.

Our investigation is aimed at establishing how sensitive these accelerations are with respect to the value chosen for ω. Section §6 describes the test problems used in the investigation and the results obtained from the experiments. In section §7 we summarise the results and draw some conclusions.

2. Dancis's Acceleration

Dancis proposes the use of a polynomial accelerator with SOR, ω being chosen such that the coordinate of the error vector, corresponding to the largest eigenvalue of $\mathcal{L}_\omega$, is annihilated. Dancis recommends that $\omega = \lambda_2 + 1$, where λ_2 is the second largest eigenvalue of $\mathcal{L}_\omega$. By computing the second largest eigenvalue, μ_2, of the Jacobi iteration matrix we can obtain the value of ω using Equation (3), with μ_1 replaced by μ_2.

Two different accelerations are proposed. The first, which we refer to as PSOR1, is as follows. Perform $r-1$ SOR iterations and then apply

$$x_{SOR}^{(r)} = \tilde{x} = \frac{1}{1-\lambda_1^r} x_{SOR}^{(r-1)} - \frac{\lambda_1^r}{1-\lambda_1^r} x^{(0)} \tag{4}$$

continuing with the SOR iterations thereafter.

The second acceleration (referred to as PSOR2) is obtained by

$$\hat{x}^{(i+1)} = \frac{1-a^i}{1-a^{i+1}} x_{SOR}^{(i+1)} + \frac{a^i(1-a)}{1-a^{i+1}} x^{(0)}, \quad i = 1, 2, \ldots, r \tag{5}$$

where $a = \omega - 1$. After r steps of the above iteration have been performed, we set $x_{SOR}^{(r+1)} = \hat{x}^{(r+1)}$ and the SOR iteration is used from then on.

Dancis propose that the value of r be chosen according to a spectral radius analysis (see [1, page 827]); its value is given by

$$\min_r = (\omega - 1)^{m-r} \left((\omega - 1)^r + \lambda_1^r\right) / (1 - \lambda_1^r)(\omega_b - 1)^m, \quad r = 1, 2, \ldots, \tag{6}$$

where m is some arbitrarily chosen number of iterations.

3. Wynn's ε Algorithm

In 1962 Wynn proposed an extension of the ε algorithm ([8]) for vector and matrix iterations ([9]). Consider a sequence $S = \{s_k\}_{k=0}^{\infty}$ which is slowly convergent. If we define the sequences $\varepsilon_{-1}^{(k)} = \{0\}_{k=1}^{\infty}$ and $\varepsilon_0^{(k)} = S$ then a new sequence $\varepsilon_{i+1}^{(k)}$ is generated by

$$\varepsilon_{i+1}^{(k)} = (\varepsilon_i^{(k+1)} - \varepsilon_i^{(k)})^{-1} + \varepsilon_{i-1}^{(k+1)}, \quad i = 0, 1, \ldots, \quad k = 0, 1, \ldots \tag{7}$$

In certain circumstances, the sequences $\varepsilon_{2i}^{(k)}$, $i = 1, 2, \ldots$ converge faster to the limit of S.

We investigate the behaviour of the $\varepsilon_2^{(k)}$ sequence obtained from vectors generated by SOR. We can express the new vector iterates, generated by two successive applications of (7) to three SOR vectors, $x_{SOR}^{(k-1)}$, $x_{SOR}^{(k)}$ and $x_{SOR}^{(k+1)}$, as

$$\varepsilon_2^{(k+1)} = \left((x_{SOR}^{(k+1)} - x_{SOR}^{(k)})^{-1} - (x_{SOR}^{(k)} - x_{SOR}^{(k-1)})^{-1}\right)^{-1} + x_{SOR}^{(k)} \tag{8}$$

where $u^{-1} = \bar{u}/(|\, u \,|^2)$ is the Moore-Penrose generalised inverse and $\bar{u}$ denotes the complex conjugate.

In [3] it is shown that the value of ω should be taken as $1 + \lambda_{2i}$ when using a sequence $\{\varepsilon_{2i}^{(k)}\}_i^{\infty}$. For the acceleration of SOR shown above, with $\varepsilon_2^{(k)}$ as the sequence to be used, then $\omega = 1 + \lambda_2$, which is the value proposed by Dancis.

4. Graves-Morris's Acceleration

Graves-Morris suggests, [3, page 25], that the sequence of vectors $x^{(k)}$ generated by (2) may be accelerated using

$$t^{(k)} = x^{(k-1)} - \frac{(\Delta x^{(k-2)})^2}{\Delta x^{(k-2)} \Delta^2 x^{(k-2)}} \Delta x^{(k-1)}, \quad k = 2, 3, \ldots \tag{9}$$

where $\Delta x^{(k)} = x^{(k+1)} - x^{(k)}$, which is a generalisation of Aitken's Δ^2 process [6]. We will refer to (9) as the G-M iteration.

Experimental results on a few model problems given in [3] show that, for the G-M iteration, using the value of ω which maximises the separation between the largest eigenvalue of the SOR iteration matrix and the other eigenvalues, a reduction in the number of iterations needed for convergence occurs. We investigate whether this reduction is also observed in larger, practical problems and, if so, how critical the eigenvalue separation is to the rate of convergence.

The value of ω is chosen in a similar way as for the ε algorithm, following [3].

5. Conditions for the Effectiveness of the Acceleration Techniques

Our aim is to discover whether we can reduce the amount of computation needed by SOR to solve a system of the form (1) by using one of the accelerations techniques.

For instance, using the G-M acceleration one might expect (intuitively) that if at least one iteration is saved, the computing workload is reduced by a factor of roughly n^2 multiplications compared to the SOR iteration. We present below necessary conditions for the techniques discussed to require a smaller number of floating point multiplications (*flops*) than SOR. These conditions are obtained in terms of the number of iterations (k) and the number of flops per iteration. For SOR and each of the accelerations we consider that the total number of flops is

$$\mathit{flops}_{SOR} = k_{SOR}(n^2+n)O_*$$

$$\mathit{flops}_{PSOR1} = \left(k_{PSOR1}(n^2+n)+2n\right)O_*$$

$$\mathit{flops}_{PSOR2} = \left(k_{PSOR2}(n^2+n)+2rn\right)O_*$$

$$\mathit{flops}_{\varepsilon_2} = \left(k_{\varepsilon_2}(n^2+7n)-6n\right)O_*$$

$$\mathit{flops}_{G\text{-}M} = k_{G\text{-}M}(n^2+3n)O_*$$

where O_* represents a floating-point multiplication.

It is easy to show that for $k_{SOR} > k_{accel}$ where *accel* denotes any of the acceleration techniques, we have

$$\mathit{flops}_{SOR} > \mathit{flops}_{accel}$$

if and only if the following conditions are satisfied

PSOR1: $n > (2 + k_{PSOR1} - k_{SOR})/(k_{SOR} - k_{PSOR1})$

PSOR2: $n > (2r + k_{PSOR2} - k_{SOR})/(k_{SOR} - k_{PSOR2})$

ε_2: $n > (7k_{\varepsilon_2} - k_{SOR} - 6)/(k_{SOR} - k_{\varepsilon_2})$

G-M: $n > (3k_{G\text{-}M} - k_{SOR})/(k_{SOR} - k_{G\text{-}M})$

6. Description of the Test Problems and Experiments

In this section we present the results obtained from solving a set of problems using MATLAB implementations of the above methods. Three of the problems are taken from the Boeing-Harwell library [2] and for these we used Fortran77 and BLAS routines to implement the methods and LAPACK subroutines to compute the eigenvalues.

The problems solved present different characteristics with respect to the distribution of eigenvalues and are of interest to the analysis of the acceleration techniques discussed.

In each experiment, we describe the system of linear equations used, the value of ω_b computed using (3) and the convergence criteria. The results are tabulated for each method in terms of number of iterations to achieve convergence and the flops counting of SOR and the ratios between the flops counting of each acceleration with respect to SOR.

For the analysis of the G-M iteration, we provide a graph showing the two largest eigenvalues of the SOR iteration matrix, λ_1 and λ_2, such that $|\lambda_1| > |\lambda_2|$. For these test problems, the value of r for the Dancis's accelerations was found to be 1.

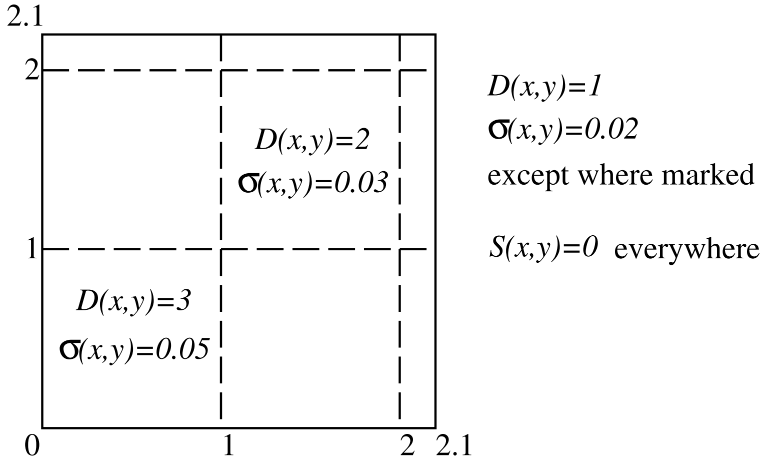
6.1. Varga's Problem

This is a system of order $n = 16$ described in [7, Appendix B] derived from the five-point finite-difference discretisation of

$$-\frac{\partial}{\partial x}\frac{\partial D(x,y)u}{\partial x} - \frac{\partial}{\partial y}\frac{\partial D(x,y)u}{\partial y} + \sigma(x,y)u = S(x,y) \tag{10}$$

in the region $R = (0, 2.1) \times (0, 2.1)$ subject to the boundary condition $\partial n = 0$, where ∂n is the outward normal. The functions D, σ and S are as shown in Figure 1. The condition number is $\kappa(A) = 3.5888 \times 10^3$. The value of ω_b is 1.9177.

Fig. 1. Region for Varga's problem.



6.1.1. *Experiment*
In this problem, we are solving

$$Ax = 0 \tag{11}$$

which has the zero vector as the unique solution. The initial vector $x^{(0)}$ was set to $(1, 1, \ldots, 1)^T$ and we iterated until the ∞-norm of the solution vector was less than 10^{-4} (this stopping criterion was used in order to reproduce the behaviour of SOR presented in [7, Appendix B, page 304]). A maximum of 2000 iterations was allowed.

An impressive reduction in the number of iterations is achieved by the G-M, ε_2 and PSOR1 accelerations. Note that while the minimum number of iterations for SOR and PSOR2 is obtained when $\omega = \omega_b$, for G-M, ε_2 and PSOR1 this minimum occurs at some $\omega < \omega_b$.

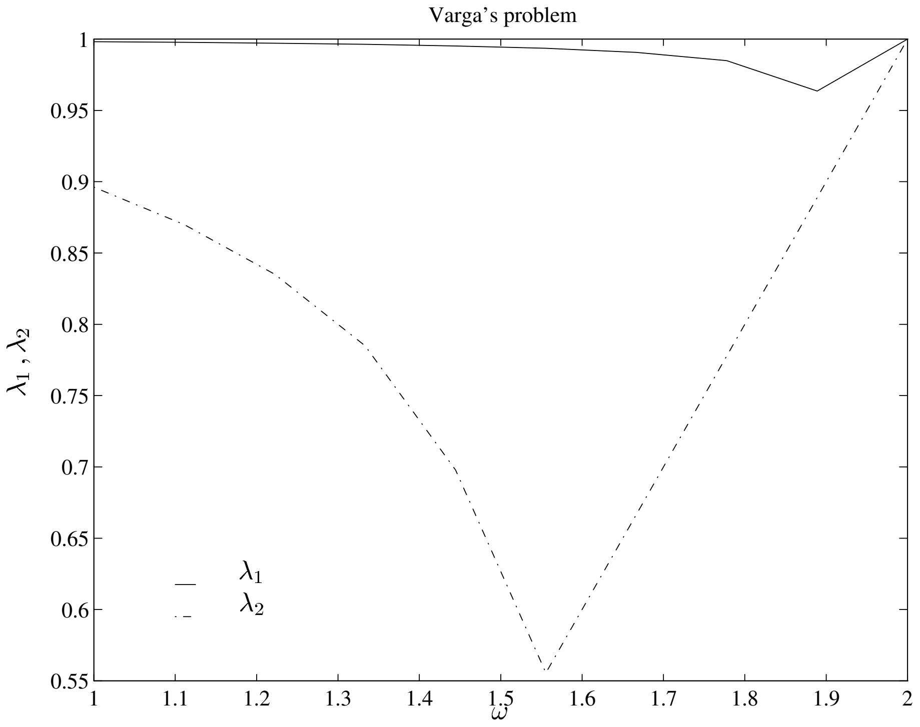
TABLE I
Varga's problem: number of iterations.

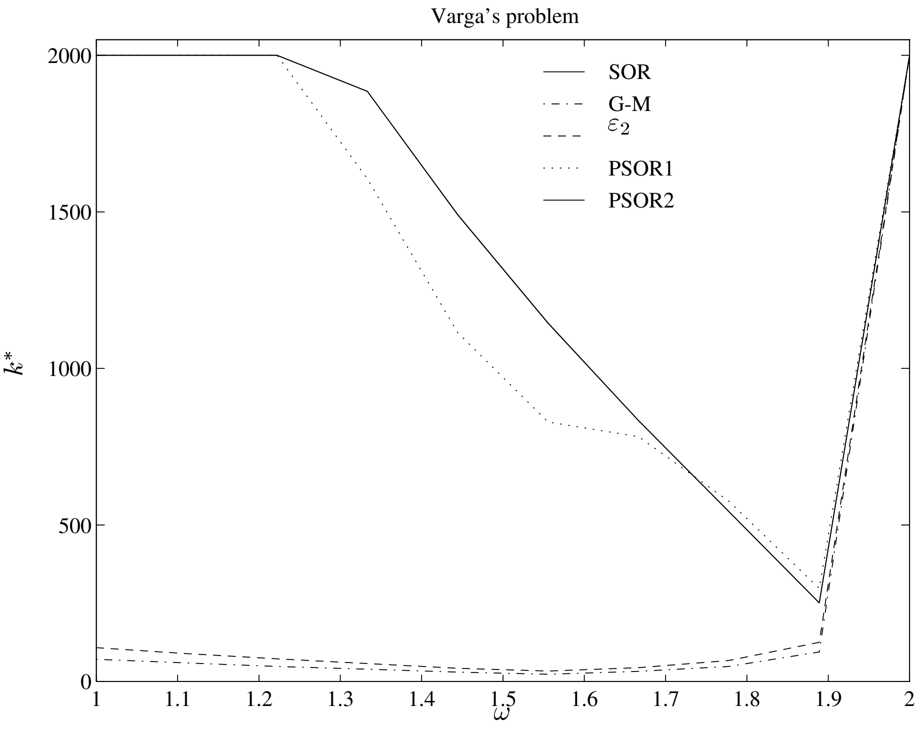
ω	SOR	G-M	ε_2	PSOR1	PSOR2
1.1019	2000	60	90	60	2000
1.3059	1992	41	60	41	1992
1.5098	1282	23	31	23	1282
1.7137	708	37	52	37	708
1.9177	146	119	152	119	146

TABLE II
Varga's problem: flops counting.

	Mflops		ratio to SOR(%)		
ω	SOR	G-M	ε_2	PSOR1	PSOR2
1.1019	0.54	3.35	6.07	3.01	100.01
1.3059	0.54	2.30	4.06	2.06	100.01
1.5098	0.35	2.01	3.24	1.80	100.01
1.7137	0.19	5.84	9.89	5.24	100.02
1.9177	0.04	91.10	140.61	81.59	100.08
ω	ratio to minimum SOR flops(%)				
1.1019	1369.86	45.93	83.16	41.18	1369.94
1.3059	1364.38	31.39	55.36	28.16	1364.46
1.5098	878.08	17.61	28.49	15.83	878.16
1.7137	484.93	28.32	47.95	25.42	485.01
1.9177	100.00	91.10	140.61	81.59	100.08

Fig. 2. Varga's problem: Eigenvalue separation and number of iterations of the methods.





6.2. Problem TRIDIAG

This problem of order $n = 100$ is the system defined by

$$A = \begin{bmatrix} 10 & 3 & & & \\ 3 & 10 & 3 & & \\ & \ddots & \ddots & \ddots & \\ & & 3 & 10 & 3 \\ & & & 3 & 10 \end{bmatrix}, b = \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \\ 1 \end{bmatrix} \tag{12}$$

The condition number is $\kappa(A) = 3.9964$ and the value of ω_b is 1.1110.

6.2.1. *Experiment*

The starting vector used is $x^{(0)} = (0, 0, \ldots, 0)^T$ and the iterations proceed until the 2-norm of the residual of the solution vector is less than 10^{-10} or the number of iterations exceeded 200.

This example shows a situation where ω_b is close to 1. The experiment shows that in this case little, if any, gain is achieved by the accelerations. Nonetheless even if a single iteration is spared a reduction in the computational effort is verified.

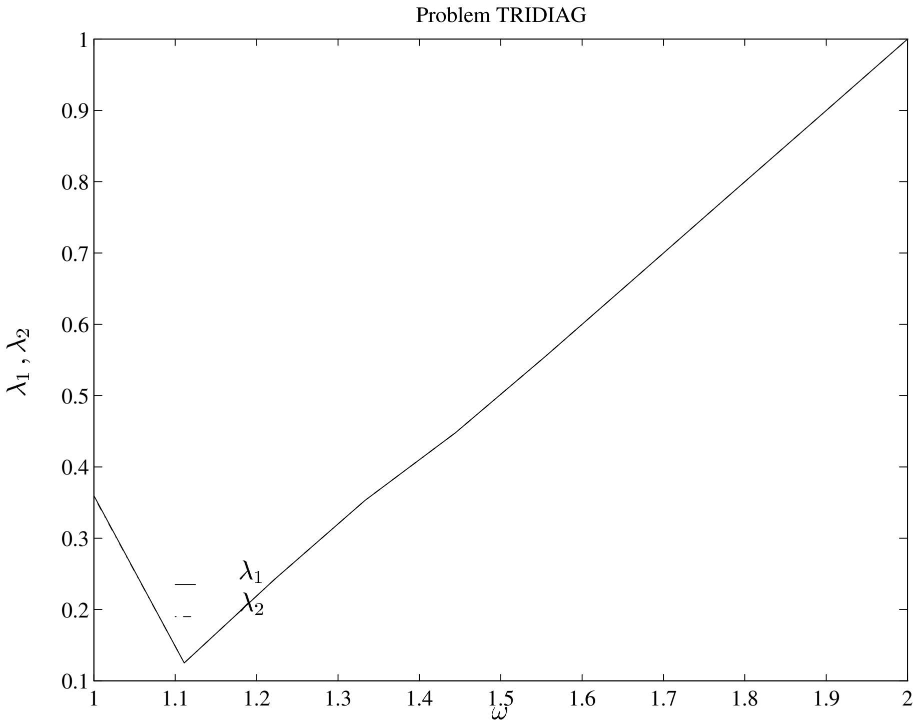
TABLE III
Problem TRIDIAG: number of iterations.

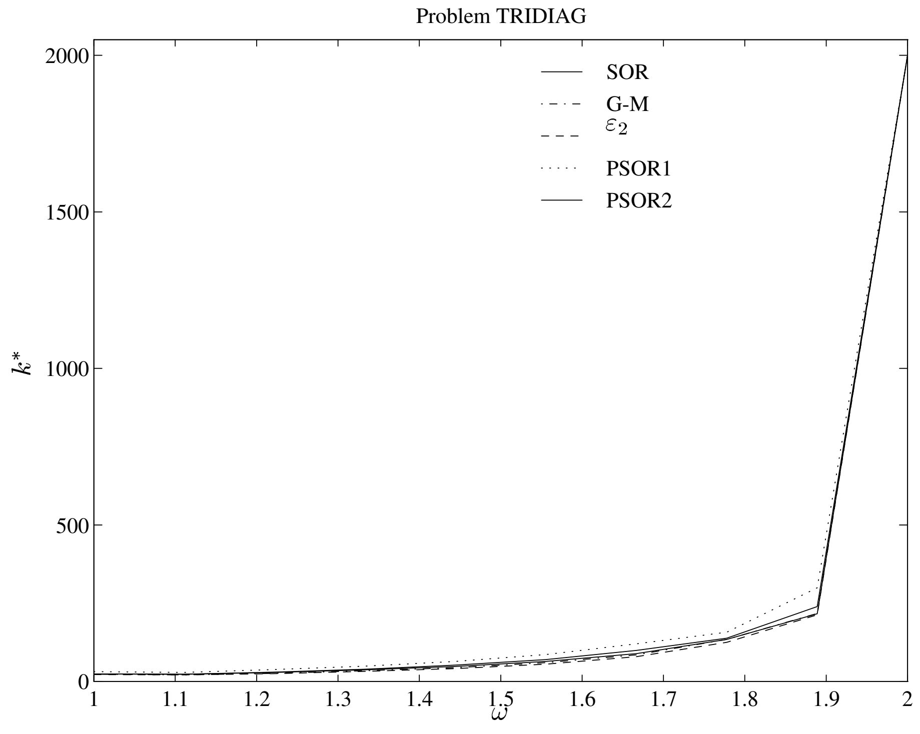
ω	SOR	G-M	ε_2	PSOR1	PSOR2
1.0123	24	22	21	22	24
1.0369	23	21	22	22	23
1.0616	22	22	23	22	22
1.0863	22	21	21	23	22
1.1109	23	21	21	24	23

TABLE IV
Problem TRIDIAG: flops counting.

	Mflops	ratio to SOR(%)			
ω	SOR	G-M	ε_2	PSOR1	PSOR2
1.0123	0.24	93.48	92.45	91.75	100.08
1.0369	0.23	93.11	101.08	95.74	100.09
1.0616	0.22	101.98	110.49	100.09	100.09
1.0863	0.22	97.34	100.86	104.64	100.09
1.1109	0.23	93.11	96.47	104.43	100.09
ω	ratio to minimum SOR flops(%)				
1.0123	109.09	101.98	100.86	100.09	109.18
1.0369	104.55	97.34	105.67	100.09	104.64
1.0616	100.00	101.98	110.49	100.09	100.09
1.0863	100.00	97.34	100.86	104.64	100.09
1.1109	104.55	97.34	100.86	109.18	104.64

Fig. 3. Problem TRIDIAG: Eigenvalue separation and number of iterations of the methods.





6.3. Problem NOS4

This problem is taken from the Harwell-Boeing sparse matrix collection [2, pp. 54-55]. It is derived from a finite-element approximation to a structural engineering problem. The system has order $n = 100$ and its condition number is $\kappa(A) = 1.5785 \times 10^3$. The RHS vector was chosen as $(1, 0, 0, \ldots, 0)^T$. The value of ω_b is 1.8810.

6.3.1. *Experiment*

In this problem we iterated until the 2-norm of the residual of the solution vector was less than 10^{-4} or the number of iterations exceeded 2000. The initial estimate of x was $(0, 0, \ldots, 0)^T$.

This example shows a similar behaviour to that of Varga's problem. However in this case the ε_2 acceleration is worse than SOR and PSOR1 and PSOR2 fail to produce any acceleration.

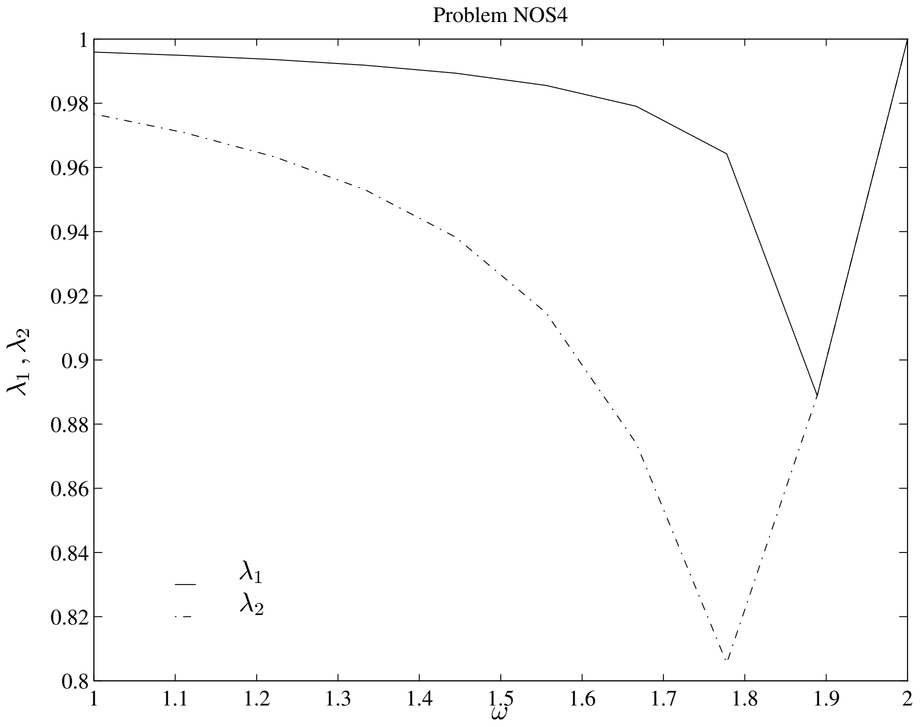
TABLE V
Problem NOS4: number of iterations.

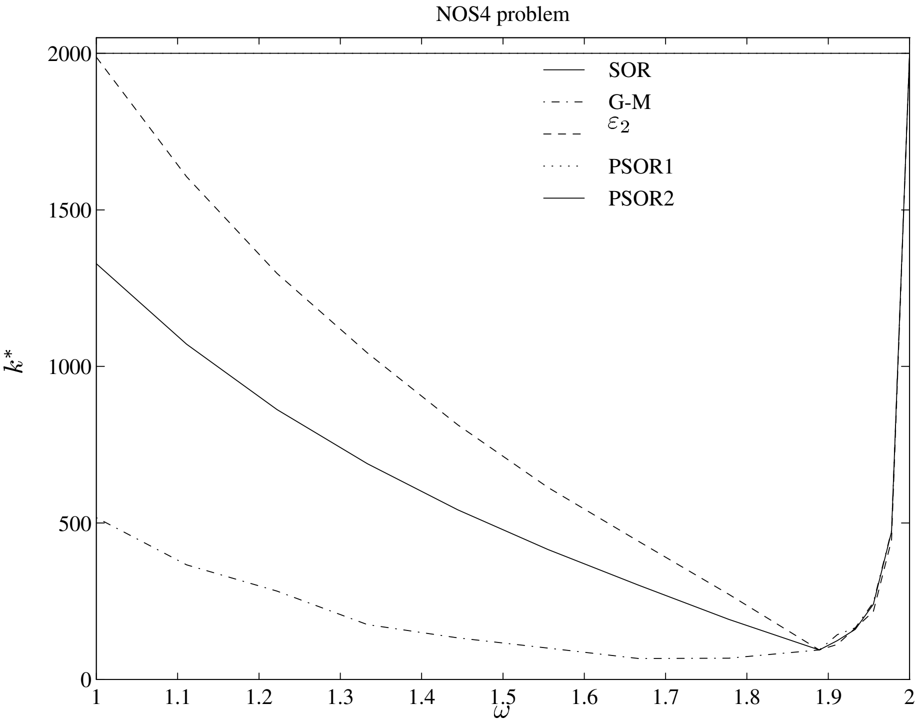
ω	SOR	G-M	ε_2	PSOR1	PSOR2
1.0978	1099	699	1667	2000	2000
1.2933	748	212	1135	2000	2000
1.4889	489	126	729	2000	2000
1.6845	283	62	409	2000	2000
1.8810	95	103	105	2000	2000

TABLE VI
Problem NOS4: flops counting.

	Mflops	ratio to SOR(%)			
ω	SOR	G-M	ε_2	PSOR1	PSOR2
1.0978	11.10	64.86	160.69	181.99	181.99
1.2933	7.55	28.90	160.74	267.38	267.38
1.4889	4.94	26.28	157.92	409.00	409.00
1.6845	2.86	22.34	153.09	706.72	706.72
1.8810	0.96	110.57	117.03	2105.28	2105.28
ω	ratio to minimum SOR flops(%)				
1.0978	1156.84	750.36	1858.92	2105.28	2105.28
1.2933	787.37	227.58	1265.65	2105.28	2105.28
1.4889	514.74	135.26	812.89	2105.28	2105.28
1.6845	297.89	66.56	456.04	2105.28	2105.28
1.8810	100.00	110.57	117.03	2105.28	2105.28

Fig. 4. Problem NOS4: Eigenvalue separation and number of iterations of the methods.





6.4. Problem 685 BUS

This problem is taken from the Harwell-Boeing sparse matrix collection [2, pp. 71]. The coefficient matrix is derived from the modelling of a power system network. The system has order $n = 685$ and its condition number is $\kappa(A) = 4.2305 \times 10^5$. The RHS vector was chosen as $(1, 0, 0, \ldots, 0)^T$. The value of ω_b is 1.9590.

6.4.1. *Experiment*

In this experiment we used the same stopping criteria as in problem NOS4. It shows a behaviour similar to that exhibited in Varga's and NOS4 problems except for the ε_2 acceleration which was always worse than SOR except at $\omega = \omega_b$.

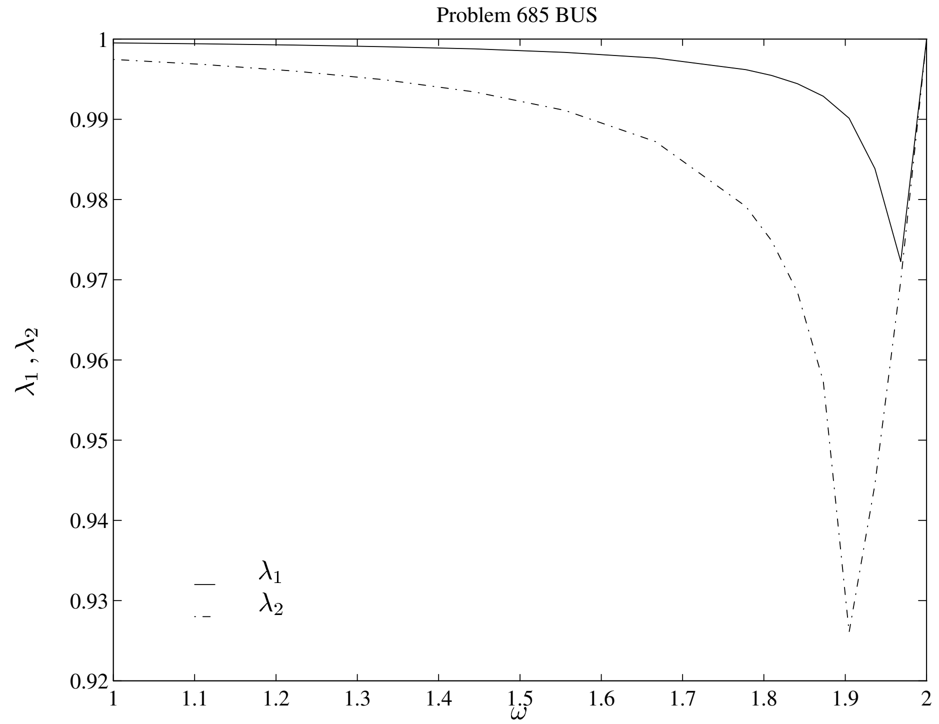
TABLE VII
Problem 685 BUS: number of iterations.

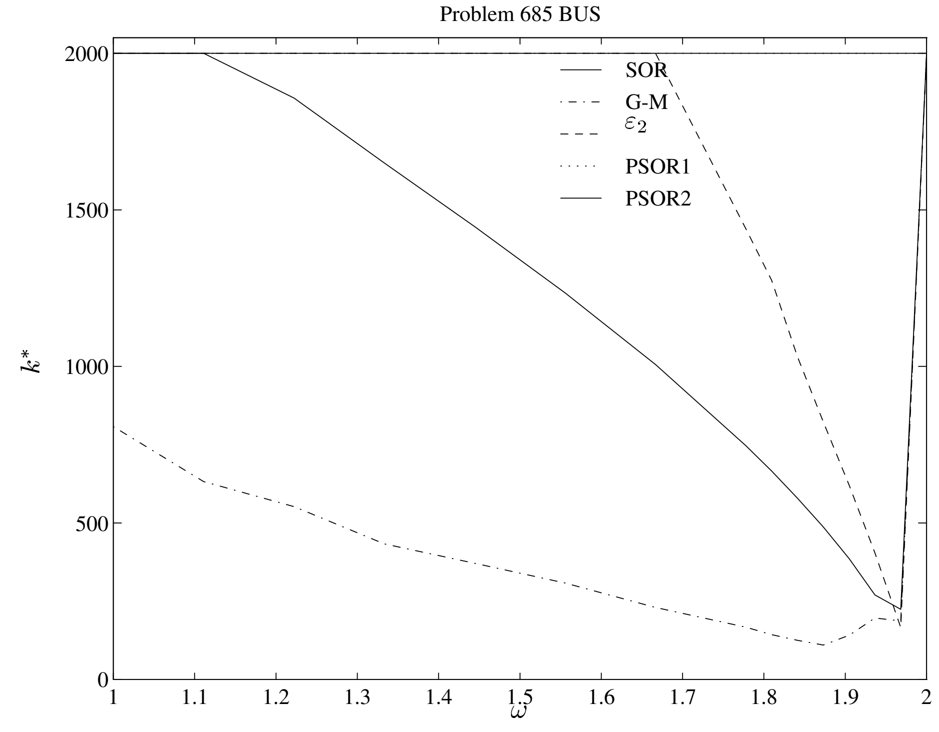
ω	SOR	G-M	ε_2	PSOR1	PSOR2
1.1066	2000	614	2000	2000	2000
1.4762	1479	384	2000	2000	2000
1.6394	1064	251	2000	2000	2000
1.8525	547	117	952	2000	2000
1.9590	171	163	165	2000	2000

TABLE VIII
Problem 685 BUS: flops counting.

	Mflops		ratio to SOR(%)		
ω	SOR	G-M	ε_2	PSOR1	PSOR2
1.1066	939.82	30.79	100.87	100.00	100.00
1.4762	695.00	26.04	136.41	135.23	135.23
1.6394	499.98	23.66	189.61	187.97	187.97
1.8525	257.04	21.45	175.56	365.63	365.63
1.9590	80.35	95.60	97.33	1169.59	1169.59
ω	ratio to minimum SOR flops(%)				
1.1066	1169.59	360.11	1179.82	1169.59	1169.59
1.4762	864.91	225.22	1179.82	1169.59	1169.59
1.6394	622.22	147.21	1179.82	1169.59	1169.59
1.8525	319.88	68.62	561.59	1169.59	1169.59
1.9590	100.00	95.60	97.33	1169.59	1169.59

Fig. 5. Problem 685 BUS: Eigenvalue separation and number of iterations of the methods.





7. Summary

We summarise the results presented in §6. The main points are as follows

1. The PSOR2 iteration performed poorly in the test problems used in this paper. The other three methods generally outperformed the basic SOR method with the G-M acceleration showing the most consistent improvements.
2. The G-M, ε_2 and PSOR1 iterations almost invariably reduced the number of iterations required to obtain a specified accuracy when $\omega < \omega_b$. In some cases this reduction was observed for $\omega = \omega_b$.
3. As with the basic SOR method, the G-M, ε_2 and PSOR1 iterations are poor if ω chosen is greater than ω_b.
4. The reduction in the number of iterations using the G-M method is proportional to the separation of λ_1 and λ_2. Since the ε_2 and PSOR1 iterations produce similar behaviour to G-M, we believe this separation also has an influence on the convergence properties of these methods. Note that as the separation between λ_1 and λ_2 decreases all three iterations exhibit similar behaviour to SOR.

The experimental results presented show that the Graves-Morris's acceleration technique is the most attractive of the techniques discussed here, from the point of view both of the overall amount of computational work required and the range of the ω parameter for which the rate of convergence is improved. Though the number of experiments performed was small we believe the results indicate that these accelerations of the SOR method are effective and may be applicable to other systems.

Acknowledgements

We would like to thank Peter Graves-Morris for his valuable comments which helped to improve this paper.

References

1. J. Dancis. The optimal ω is not best for the SOR iteration method. *Linear Algebra and its Applications*, 154-156:819–845, 1991.
2. I.S. Duff, R.G. Grimes, and J.G. Lewis. Users' guide for the Harwell-Boeing sparse matrix collection. Report TR/PA/92/86, CERFACS, October 1992.
3. P.R. Graves-Morris. A review of Padé methods for the acceleration of convergence of a sequence of vectors. Report No. NA 92-31, Department of Mathematics, University of Bradford, October 1992.
4. R.G. Grimes, D.R. Kincaid, and D.M. Young. ITPACK 2.0 user's guide. Report No. CNA-150, Center for Numerical Analysis, University of Texas at Austin, August 1979.
5. L.A. Hageman and D.M. Young. *Applied Iterative Methods.* Academic Press, New York, 1981.
6. D.R. Kincaid and W. Cheney. *Numerical Analysis.* Brooks/Cole, Pacific Grove, 1991.
7. R.S. Varga. *Matrix Iterative Analysis.* Prentice-Hall, Englewood Cliffs, 1962.
8. P. Wynn. On a device for computing the $e_m(S_n)$ transformation. *Mathematical Tables and Aids of Computation*, 10:91–96, 1956.
9. P. Wynn. Acceleration techniques for iterated vector and matrix problems. *Mathematics of Computation*, 16:301–322, 1962.