

## Note

# Plasma Equilibrium Calculations by Line Successive over Relaxation

### I. INTRODUCTION

Line successive over relaxation (LSOR) is an iterative method for solving elliptic differential equations [1]. LSOR takes advantage of the CRAY vector capabilities as compared to the point successive over relaxation (SOR) method, which does not vectorize. The substantial advantages of LSOR on a vectorizing machine are not well known, except in the field of aerodynamics [2, 3]. By minor modification of the traditional SOR elliptic equation solver, we find that in certain coordinates an increase of a factor of two or greater in convergence time can be realized.

As a model problem for comparison of SOR and LSOR, the numerical solution of Poisson's equation will be reviewed in Section II. In Section III, we discuss the decreased computation time on the National Magnetic Fusion Energy Computer Center (NMFEECC) CRAY computers found with LSOR applied to the iterative solution of plasma equilibria. In Section IV, the conditions for which LSOR is most useful are summarized.

### II. SOLUTION OF POISSON'S EQUATION BY SOR AND LSOR

Centered finite differencing of Poisson's equation

$$\nabla^2 P = S, \tag{1}$$

for a  $10 \times 10$  cylindrical  $[r(i), \theta(j)]$  grid leads to

$$\begin{aligned} & \frac{r}{36(r^2 + 1)} [P_{i+1,j} - P_{i-1,j}] + \frac{r^2}{2(r^2 + 1)} [P_{i+1,j} + P_{i-1,j}] \\ & + \frac{1}{2(r^2 + 1)} [P_{i,j+1} + P_{i,j-1}] - P_{ij} = \frac{S_{ij}r^2}{162(r^2 + 1)}. \end{aligned} \tag{2}$$

In the SOR technique, the  $n$ th iteration of  $P$  is calculated from

$$P_{ij}^n = P_{ij}^{n-1} + \omega R_{ij}. \tag{3}$$

Here the residual  $R(i, j)$  is

$$R_{ij} = \frac{r}{36(r^2 + 1)} [P_{i+1,j}^{n-1} - P_{i-1,j}^n] + \frac{r^2}{2(r^2 + 1)} [P_{i+1,j}^{n-1} + P_{i-1,j}^n] + \frac{1}{2(r^2 + 1)} [P_{i,j+1}^{n-1} + P_{i,j-1}^n] - P_{ij}^{n-1} - \frac{S_{ij}^{n-1} r^2}{162(r^2 + 1)}. \quad (4)$$

Varying the SOR parameter  $\omega$  optimizes convergence time.

In the LSOR [4] method, we write the set of finite differenced equations for all points on the  $(i, j)$  grid in matrix form

$$\begin{bmatrix} D_1 & U_2 & & & \\ L_1 & D_2 & U_3 & & \\ & L_2 & D_3 & U_4 & \\ & & & \ddots & \end{bmatrix} P = S. \quad (5)$$

Here  $P = (P_1, P_2, \dots)$  and each vector  $P_j$  represents a whole row of grid points. Each row  $P_j$  can be found from

$$P_j^T = D_j^{-1} [S_j - L_j P_{j-1}^n - U_{j+1} P_{j+1}^{n-1}], \quad (6)$$

via a tridiagonal matrix inversion. The vector  $P_j^T$  is used to increment the  $(n-1)$ th vector  $P_j^{n-1}$ ,

$$P_j^n = (1 - \omega) P_j^{n-1} + \omega P_j^T. \quad (7)$$

The point SOR method uses the advanced  $(k+1)$  values at two neighboring points  $(i-1, j)$  and  $(i, j-1)$ . Line SOR uses the advanced  $(k+1)$  values at three neighboring points and so slightly improves the convergence rate on a scalar computer [4].

To compare the speed of SOR and LSOR on a vectorizing computer, solutions of Poisson's equation with a nonlinear source

$$\nabla^2 P = P^2 \exp(P^2), \quad (8)$$

were obtained in a cylindrical geometry with cyclic boundary conditions. The equation was solved with tridiagonal solution for rows at constant radius and also for rows at constant  $\theta$  (Fig. 1). It was found that LSOR with tridiagonal solution along  $\theta$  (at constant radius) was fastest. This converged ten times faster than SOR when sweeping radial grid points at constant  $\theta$ . These conclusions also hold for a stationary linear source.

In both cylindrical and rectangular coordinates LSOR was significantly faster than SOR for computations requiring many iterations. In cylindrical coordinates, with an

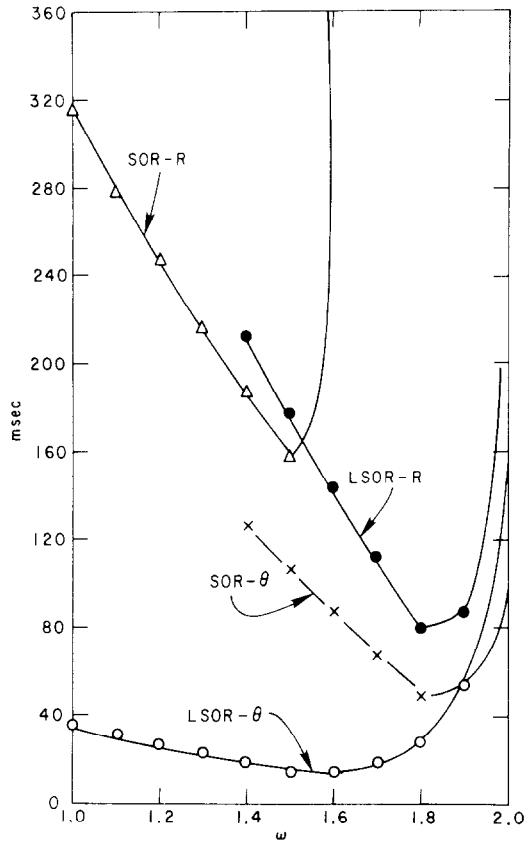


FIG. 1. Cpu convergence time to solve Poisson's equation by LSOR and SOR methods as a function of over relaxation parameter  $\omega$ . For LSOR-R and SOR-R the tridiagonal solution is obtained at constant  $\theta$ . For LSOR- $\theta$  and SOR- $\theta$  the tridiagonal solution is obtained at constant radius.

error criterion requiring at least 400 SOR iterations for convergence, LSOR was much faster than SOR (Fig. 1). LSOR and SOR were found to be equally fast when only ten SOR iterations were needed. Imposed up-down symmetry would further speed LSOR convergence. The LSOR method of solving Poisson's equation converges fastest when the solution is slowly varying in the direction along which the tridiagonal solution is found. The problem is thereby quickly reduced to quasi-one-dimensional.

## III. SOR AND LSOR SOLUTIONS OF PLASMA EQUILIBRIA

Replacing SOR and LSOR in the princeton equilibrium code EQ accelerates convergence for a typical plasma fixed boundary equilibrium. EQ solves the Grad-Shafranov equation [5, 6] for the poloidal flux function  $\chi$  which is derived from the equilibrium force balance between the magnetic and kinetic pressure in tokamak plasmas. This equation has a nonlinear source since the pressure and toroidal field functions are dependent on  $\chi$ .

The equation is solved in magnetic flux coordinates by the method of DeLucia, Jardin, and Todd [7]. The equation is solved by iteration for the poloidal flux function. This is used to compute a new magnetic coordinate system via the Jacobian constraint as well as by matching the total measured current and central plasma pressure. Then the Grad-Shafranov equation is solved again in the new coordinate

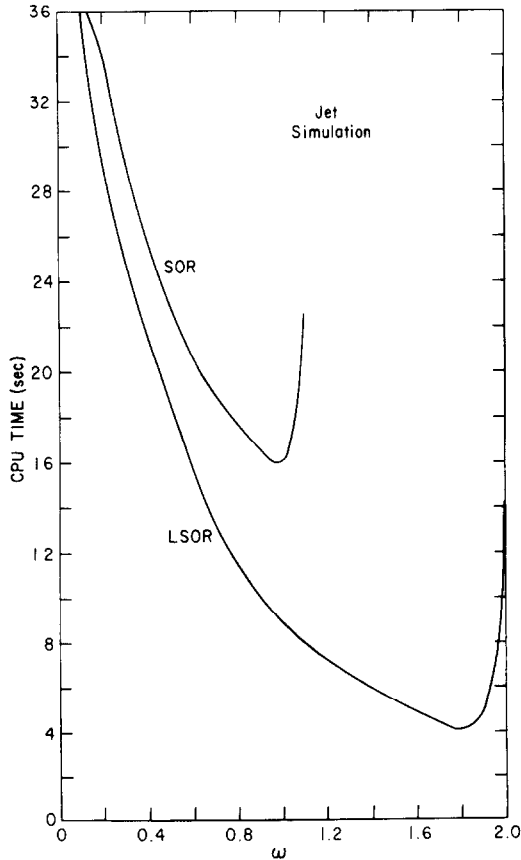


FIG. 2. Computational (cpu) convergence time for EQ for a JET simulation for SOR and LSOR as a function of relaxation parameter  $\omega$ .

system. The coordinate readjustment and the iterative Grad–Shafranov solution continue until the error criterion is no longer exceeded.

In EQ, the equilibrium solution is obtained in a nonorthogonal cylindrical geometry  $(\theta, r)$  in the poloidal plane. The convergence time required for EQ with SOR was the same whether  $\theta$  or  $r$  is the direction of successive sweeps. However, the convergence time for EQ with LSOR is about three times faster for tridiagonal solutions along  $\theta$  rather than along  $r$  for a typical poloidal divertor experiment (PDX) plasma simulation.

Figure 2 presents the total cpu time required for convergence of EQ as a function of the relaxation parameter. The results of SOR and LSOR calculations are shown for a joint European torus (JET) simulation. We find that optimized LSOR reduces cpu time required for convergence. The LSOR JET simulation convergence is faster than SOR by a factor of four, while a LSOR PDX simulation (not shown) converged faster by a factor of two. The JET simulation represents a noncircular, low aspect ratio, high current plasma. This is an equilibrium with steeper gradients, a more nonlinear grid and requires a more time-consuming calculation. For this case LSOR is faster than SOR at all values of  $\omega$ .

Another approach to obtaining a fast accurate two-dimensional plasma MHD equilibrium is that of Lao, Hirshman, and Wieland [8]. They developed a variational moments method which takes about 0.2 sec of CRAY time to compute an equilibrium with a relative error of  $10^{-3}$  for three amplitude functions. For a circular PDX-type simulation, EQ with SOR iteration and error criterion

$$\frac{\text{Max}_{ij} |\psi^n - \psi^{n-1}|}{\langle |\psi^n| \rangle_{ij}} \leq 10^{-5}, \quad (9)$$

converges in 9 sec of CRAY time for 80 poloidal points and imposed up-down symmetry. The toroidal flux function is  $\psi$ . EQ with LSOR obtains the same solution in 4.5 sec for this fixed boundary case.

The LSOR-modified EQ code runs more quickly than EQ with SOR. It is potentially more accurate than the Lao–Hirshman–Wieland moments code for high beta equilibria having steep gradients, since it computes with all Fourier amplitude functions. When the distortion of flux surfaces is not large, the Lao–Hirshman–Wieland code will be faster.

#### IV. CONCLUSION

On scalar computing machines, LSOR is only slightly faster than SOR [9]. LSOR converges in fewer iterations than point SOR but each iteration may take longer because of the implicit tridiagonal solution required. SOR is an adequate iterative method for solution of elliptic equations when the problem (a) has no axis of

symmetry, (b) does not require a great many iterations, or (c) must be solved on a scalar computer. On a vectorizing machine problems requiring many iterations of a nonlinear elliptic equation can often be solved faster by LSOR.

#### ACKNOWLEDGMENTS

The authors wish to thank C. Singer and D. Post for direction, S. C. Jardin for suggesting that cylindrical geometry might be important for fast LSOR convergence, R. Hulse for the tridiagonal solution of a system with cyclic boundary conditions, and M. F. Reusch for valuable discussions. This work was supported by the U. S. Department of Energy Contract DE-AC02-76-CHO-3073.

#### REFERENCES

1. F. F. AMES, "Numerical Methods for Partial Differential Equations," p. 146, Nelson, London, 1969.
2. D. REDHED, A. W. CHEN, AND S. G. HOTOVY, *A.I.A.A. J.* **17** (1) (1979), 98.
3. J. C. SOUTH, JR., J. D. KELLER, AND M. M. HAFEZ, *A.I.A.A. J.* **18** (7) (1980), 786.
4. P. J. ROACHE, "Computational Fluid Dynamics," p. 119, Hermosa, Albuquerque, N. Mex., 1976.
5. H. GRAD AND H. RUBIN, Hydromagnetic Equilibria and Force Free Fields, in "Proceedings of the Second United Nations International Conference on Peaceful Uses of Atomic Energy, Geneva," Vol. 31, p. 191, United Nations Publications, Geneva, 1958.
6. V. D. SHAFRANOV, Plasma Equilibrium in a Magnetic Field, in "Reviews of Plasma Physics" (M. A. Leontovich, Ed.), Vol. 2, p. 103, Plenum, New York, 1966.
7. J. DELUCIA, S. C. JARDIN, AND A. M. M. TODD, *J. Comput. Phys.* **37** (1980), 183.
8. L. L. LAO, S. P. HIRSHMAN, AND R. M. WIELAND, *Phys. Fluids* **24** (1981), 1431.
9. Y. PAO AND R. J. DOUGHERTY, Boeing Scientific Research Laboratories Report D1-82-0822, 1969.

RECEIVED: May 3, 1983; REVISED: July 19, 1983

M. H. REDI AND D. A. LARRABEE

*Plasma Physics Laboratory, Princeton University  
Princeton, New Jersey 08544*