PARALLEL PERFORMANCE OF POWER-METHOD PREDICTION OF OPTIMAL SOR PARAMETERS

J. M. MCDONOUGH

* University of Kentucky, Departments of Mechanical Engineering and Mathematics, Lexington, KY, USA 40506
e-mail: jmmcd@uky.edu, web page: http://www.engr.uky.edu/~acfd/

Key words: Power method parallelization, optimal SOR parameters

Abstract. This extended abstract provides background and motivation for parallelizing the power method in order to efficiently compute optimal iteration parameters for SOR.

1 INTRODUCTION

It is well known that successive overrelaxation (SOR) performs efficiently in solving Poisson equations, e.g., \( \Delta p = f(x) \) (as frequently occur in incompressible CFD), only if the optimal iteration parameter (often denoted \( \omega_b \)) is employed. Moreover, \( \omega_b \) can be analytically calculated essentially only for Dirichlet problems on rectangular domains. This is because the formula for this parameter,

\[
\omega_b = \frac{2}{1 + \sqrt{1 - \rho(B)^2}},
\]

contains the spectral radius \( \rho(B) \) of the Jacobi iteration matrix \( B \) (see Young [1]); and this can be derived analytically for only Poisson/Dirichlet problems posed on rectangles.

Because SOR is not only a simple and relatively efficient algorithm, but also because it is easily parallelized (see Tang et al. [2]), it would be useful to be able to numerically (and automatically) calculate \( \rho(B) \) for general problems not admitting an analytical formula. In particular, for projection-like methods often employed in CFD, at least some boundary conditions are not of Dirichlet type, and domains are often more general than simple rectangles; hence, there is no formula for predicting \( \rho(B) \). Here, we investigate use of the power method for this purpose.

The power method is a basic fixed-point iteration for determining the eigenvector, say \( X \), corresponding to the eigenvalue, \( \lambda \), of maximum modulus (hence, the spectral radius) of a finite-dimensional matrix \( A \); i.e.,

\[
X^{(n+1)} = AX^{(n)},
\]

modulo required normalizations (to prevent floating-point overflow), with parenthesized superscripts being iteration counters. The corresponding eigenvalue \( \lambda \) can be calculated in at least two different ways. Here, we employ the Rayleigh quotient given as

\[
\lambda^{(n+1)} = \frac{\langle AX^{(n)}, X^{(n)} \rangle}{\langle X^{(n+1)}, X^{(n+1)} \rangle},
\]

with \( \lambda^{(n+1)} \) indicating the advanced iterate of the eigenvalue and with \( \langle \cdot, \cdot \rangle \) denoting inner product. In our present context, the matrix \( A \) will be the Jacobi iteration matrix \( B \) associated with analysis of the SOR method, and \( \lambda \) will be \( \rho(B) \). Parallelization is implemented, via OpenMP, to improve efficiency of using Eqs. (2),(3) to numerically determine \( \rho(B) \) used in Eq. (1).
2 ANALYSIS/RESULTS

For this brief abstract we limit work to Dirichlet cases for which exact results are known, permitting validation of the power method code; in particular, for Poisson’s equation discretized with second-order centered differencing employing uniform grid spacing $h$ on the unit square, it is easily found (see [1]) that $\rho(B) = \cos \pi h$. It can be seen from Eqs. (2) and (3) that the main contributions to the arithmetic are a sparse matrix multiply and two inner-product calculations, each requiring $O(N)$ operations for $X \in \mathbb{R}^N$. This can be parallelized in line-by-line fashion using Parallel DO Loops in OpenMP Fortran for the matrix multiply. Parallelization of the inner-products has not yet been performed.

In the left-hand figure we provide validation of use of the power-method algorithm, and the present coding of it, for finding $\rho(B)$. The points correspond to the exact $\rho(B)$ plotted against the calculations for a range of $N$ from $101^2$ to $1001^2$. Weak speedups for 1 to 12 cores (after which efficiency deteriorated) are summarized in the right-hand figure for $N = 301^2$ points with run time for a single core being $\sim 16.8$ sec on a typical Intel processor. We remark that “estimated” speedups attempt to account for the fact that nearly $2/3$ of the total arithmetic has yet to be parallelized at this time; but these may be overly optimistic.

3 SUMMARY/CONCLUSIONS

In this abstract we have demonstrated validity and estimated parallel speedups when using a power-method algorithm for predicting optimal SOR iteration parameters. At the conference presentation, and in the conference paper, we will provide details for non-Dirichlet problems and larger numbers of discrete finite-difference points.

REFERENCES
