Performance Analysis of Block Jacobi Preconditioning Technique Based on Block Broyden Method*

Peng Jiang[1], Geng Yang[1], and Chunming Rong[2]

[1] School of Computer Science and Technology, P.O.Box 43, Nanjing University of Posts and Telecommunications, 210003, Nanjing, China
alice20006@hotmail.com, yangg@njupt.edu.cn
[2] Department of Electrical and Computer Engineering, University of Stavanger, Norway
chunming.rong@uis.no

Abstract. The Block Jacobi preconditioning technique based on Block Broyden method is introduced to solve nonlinear equations. This paper theoretically analyzes the time complexity of this algorithm as well as the unpreconditioned one. Numerical experiments are used to show that Block Jacobi preconditioning method, compared with the unpreconditioned one, has faster solving speed and better performance under different dimensions and numbers of blocks.

1 Introduction

In the past few years, a number of books entirely devoted to iterative methods for nonlinear systems have appeared. The Block Broyden Algorithm was proposed and analyzed in References [1, 2]. However, the convergence speed of this algorithm is affected to some extent, for the information among the nodes is always lost. Hence, seeking for proper preconditioning methods[3] is one of the effective ways to solve this problem. Some preconditioners have been proposed and discussed in Reference [4].

This paper introduces Block Jacobi preconditioning technique based on Block Broyden method and we name this method as BJBB. It also analyzes time complexity and applies BJBB method to nonlinear systems arising from the Bratu problem.

2 Block Jacobi Method Based on Block Broyden Algorithm

2.1 General Remarks

In the following discussion, we are concerned with the problem of solving the large system of nonlinear equations as (1):

$$F(x) = 0 . \tag{1}$$

* This work was supported in part by the Natural Science Foundation of Jiangsu Province under Grant No 05KJD520144 and the Foundation of the QingLan Project (KZ0040704006).

V.N. Alexandrov et al. (Eds.): ICCS 2006, Part I, LNCS 3991, pp. 794 – 797, 2006.

where $F(x) = (f_1, \cdots . f_n)^T$ is a nonlinear operator from R^n to R^n, and $x^* \in R^n$ is an exact solution. Suppose that the components of x and F are divided into q blocks:

$$F = \begin{pmatrix} F_1 \\ \vdots \\ F_q \end{pmatrix} \qquad x = \begin{pmatrix} x_1 \\ \vdots \\ x_q \end{pmatrix}$$

We consider the Block Jacobi preconditioning technique based on Block Broyden method as follows:

- **Algorithm 2.1.1** BJBB Method

1. Let x^0 be an initial guess of x^*, and B^0 an initial block diagonal approximation of $J(x^0)$. Calculate $r^0 = F(x^0)$.
2. For k = 0, 1…until convergence:
   2.1 Solve $B^k s^k = -r^k$:
      2.1.1 Calculate the Block Jacobi preconditioner M:
         If the index set $S = \{1, \cdots n\}$ is partitioned as $S = U_i S_i$ with the sets S_i mutually disjoint, then the elements $m_{i,j}$ of preconditioner M is:

$$m_{i,j} = \begin{cases} a_{i,j} & \textit{if and } j \textit{ are in the same index subset} \\ 0 & \textit{otherwise} \end{cases}$$

      2.1.2 Calculate the inverse of the preconditioner.
      2.1.3 Transform the linear system as $\mathrm{M}^{-1} B^{\mathrm{k}} s^k = -\mathrm{M}^{-1} r^k$ and solve it by Jacobi method.
   2.2 Update the solution $x^{k+1} = x^k + s^k$.
   2.3 Calculate $r^{k+1} = F(x^{k+1})$. If r^{k+1} is small enough, stop.
   2.4 Calculate $(s^k)^T s^k$ and update B^{k+1} by

$$B_i^{k+1} = B_i^k + \frac{r_i^{k+1}(s_i^k)^T}{(s^k)^T s^k} \quad . \tag{2}$$

   Then set $k = k + 1$, and go to step 2.

2.2 Time Complexity

From Reference [2], we know that the complexity of Block Jacobi method is:

$$U = \sum_{i=1}^{q} (4n_i - 1 + 2n_i^2) + L(n) + R(n) \quad . \tag{3}$$

here $L(n)$ means the complexity of solving q block linear equations $B_i^k s_i^k = -r_i^k$, and $R(n)$ is the calculative cost in step 2.3, Algorithm 2.1.1. From (3) we can know that the value of U differs in $L(n)$ for various methods.

For unpreconditioned method, we can deduce the value of $L(n)$ as follows:

$$L_n(n) = k_n \times (2\bar{n}^3 + \bar{n}^2 + \bar{n}) \ . \tag{4}$$

where k_n refers to the addition of number of iterations for unpreconditioned method..

For the BJBB method, we can get L(n) as follows:

$$L_p(n) = q \times (\frac{\bar{n}^4}{3} + 2\bar{n}^3 - \frac{\bar{n}^2}{3} - \bar{n}) + k_p \times (2\bar{n}^3 + \bar{n}^2 + \bar{n}) . \tag{5}$$

where k_p refers to the addition of number of iteration for BJBB method.

3 Numerical Experiments

Suppose a nonlinear partial differential equation can be written as

$$\begin{cases} -\Delta u + u_x + \lambda e^u = f, \\ u\,|_{\partial\Omega} = 0 \end{cases} \quad (x, y) \in \Omega = [0,1] \times [0,1] \ . \tag{6}$$

It is known as the Bratu problem and has been used as a test problem by Yang in [2] and Jiang in [4]. In the following tests, we suppose N=110, 150 and 180, giving three grids, M1, M2 and M3, with 12100, 22500, 32400 unknowns, respectively. And we set block number q1=2000, q2=800 and q3=2500 for each grid. Table 1, 2, and 3 show the number of nonlinear iterations, which is denoted by "k" and the sum of numbers of iterations during the i-th nonlinear iteration, which is denoted by "k[i]".

Table 1. Comparison of the total number of iterations in M1, q1

	BJBB	No Preconditioner
k	3950	4499
k[500]	7106	13095
k[1500]	3533	7243

Table 2. Comparison of the total number of iterations in M2, q2

	BJBB	No Preconditioner
k	4335	4713
k[1700]	1088	5601
k[3290]	512	5020

Table 3. Comparison of the total number of iterations in M3, q3

	BJBB	No Preconditioner
k	4581	4884
k[1000]	4149	11499
k[3000]	1735	8994

To judge the performance of each method, we use data shown in Table 1 as an example. During the 1500–th iteration, the following can be known:

$$k_p = 3533,\ k_n = 7243,\ q_1 = 2000,\ \bar{n} = M_1 / q_1 = 12100/2000 = 6$$

According to (4), we get $L_n(n) = 3433182$ for the unpreconditioned method. According to (5), we get $L_p(n) = 3366642$ for BJBB method. Thus we find that $L_p(n) < L_n(n)$, so the performance of BJBB method is much better than the unpreconditioned one.

4 Conclusions

We have proposed Block Jacobi preconditioning technique based on Block Broyden Method for solving nonlinear systems. It shows evidently some advantages to combine Block Broyden Algorithm with preconditioners.

References

1. Yang G, Dutto L, Fortin M: Inexact block Jacobi Broyden methods for solving nonlinear systems of equations. SIAM J on Scientific Computing (1997) 1367-1392
2. Yang Geng: Analysis of parallel algorithms for solving nonlinear systems of equations. Chinese Journal of Computer (2000) 555-777 (in Chinese)
3. M. Benzi: Preconditioning Techniques for Large Linear Systems: A Survey. J Comput Phys (2002) 418-477
4. Peng Jiang, Geng Yang: Performance Analysis of Preconditioners based on Broyden Method. Applied Mathematics and Computation (Accepted for publication)