Experiments with MRAI time stepping schemes on a distributed memory parallel environment

Mikhail A. Botchev

Mathematical Institute, Utrecht University,
P.O.Box 80.010, 3508 TA Utrecht, the Netherlands.
botchev@@math.ruu.nl

Presenting author: Mikhail A. Botchev

E-mail: botchev@@math.ruu.nl

tel.: (+31 30) 253 46 30
fax: (+31 30) 251 83 94

Botchev, M.A. (1998) Experiments with MRAI time stepping schemes on a distributed memory parallel environment. In: Proceedings of the International Conference on "High-performance computing and networking" (HPCN'98), Aplril 21-23, 1998, Amsterdam, the Netherlands. pp. 872-874. Lecture Notes in Computer Science 1401. Springer Verlag. ISSN 0302-9743

Experiments with MRAI time stepping schemes on a distributed memory parallel environment

Mikhail A. Botchev

Mathematical Institute, Utrecht University,
P.O.Box 80.010, 3508 TA Utrecht, the Netherlands.
botchev@@math.ruu.nl

Abstract. Implicit time stepping is often difficult to parallelize. The recently proposed Minimal Residual Approximate Implicit (MRAI) schemes [2] are specially designed as a cheaper and parallelizable alternative for implicit time stepping. A several GMRES iterations are performed to solve approximately the implicit scheme of interest, and the step size is adjusted to guarantee stability.

A natural way to apply the approach is to modify a given implicit scheme in which one is interested. Here, we present numerical results for two parallel implementations of MRAI schemes. One is based on the simple Euler Backward scheme, and the other is the MRAI-modified multistep ODE solver LSODE.

On the Cray T3E and IBM SP2 platforms, the MRAI codes exhibit parallelism of explicit schemes. The model problem under consideration is the 3D spatially discretized heat equation. The speed-up results for the Cray T3E and IBM SP2 are reported.

1 MRAI time stepping

Assume that to solve a system of N ODE's $\boldsymbol{y}' = \boldsymbol{f}(t, \boldsymbol{y})$, we are interested in some implicit time stepping, for instance, the Euler Backward (EB) scheme

$$\boldsymbol{y}^{n+1} - \boldsymbol{y}^n = \tau \boldsymbol{f}(t_{n+1}, \boldsymbol{y}^{n+1}). \tag{1}$$

This nonlinear system in $\boldsymbol{y}^{n+1}$ is usually linearized, and the corresponding Jacobian linear system $(I - \tau J)(\boldsymbol{y}^{n+1} - \boldsymbol{y}^n) = \tau \boldsymbol{f}(t_{n+1}, \boldsymbol{y}^n)$, $J = \frac{\partial \boldsymbol{f}}{\partial \boldsymbol{y}}(t_{n+1}, \boldsymbol{y}^n)$ is solved approximately. In Newton's process this procedure is repeated.

The basic idea in the MRAI time stepping [2] is very simple: at each time step, we solve the Jacobian system approximately with k steps of GMRES [7]. The number of iterations k is fixed and taken small (say 5). MRAI scheme is an approximation for an implicit scheme and therefore it is not unconditionally stable. A step size control for stability is proposed in [2]; it is based on the information delivered by the GMRES process.

In MRAI schemes one does not control the residual reduction achieved in GMRES, and the number of iterations k is kept fixed. This makes the overall scheme quite simple.

2 Parallelization of MRAI and numerical experiments

It is well known how to parallelize the conjugate gradient type iterative methods (as GMRES) (see e.g. [9, 1, 5]). In our experience it turns out that, on the platforms as the IBM SP2 and Cray T3E, there is no need in modifications proposed in [5, 1].

As a model problem we take a spatially discretized 3D heat equation. (This model problem is used in [8].) The standard 7-point stencil finite difference discretization on the spatial grid $40 \times 40 \times 40$ leads to the system of size $N = 64\,000$. The numerical integration is done for $t \in [0, 0.7]$.

In our tests, we use two experimental MRAI codes. The first one is based on the simple Euler Backward scheme (we refer to the code as EB/MRAI), the second is the MRAI-modified stiff ODE solver LSODE (the LSODE/MRAI code). In [2], the performance of the LSODE/MRAI code was tested and compared with the RKC [8] and VODPK [3] codes. For the model problem under consideration, the EB/MRAI code gives the CPU time gain factor 3.2 with respect to the Euler Forward scheme. Both codes use matrix free Jacobian evaluation (see e.g. [6, 4]). Number of GMRES steps was always $k = 5$. The both tolerance parameters `atol` and `rtol` in the LSODE/MRAI code were taken 10^{-3}. In the EB/MRAI code the step size was chosen automatically on the base of the MRAI stability control [2].

In parallel versions of the code, we used the MPI communication library. In fact, it appears that on both the IBM SP2 and Cray T3E platforms these MRAI codes possess parallelism of explicit schemes, i.e. the speed-up is restricted *only* by evaluations of $\boldsymbol{f}$.

Simple analysis based on the Amdahl's law suggests that for the MRAI schemes the speed-up is of the form

$$S_p = \frac{p}{1 + p(1-f)\alpha_p}$$

where p is number of processors, f is the parallel fraction of computational work in one time step of MRAI, and $\alpha_p = t_p^{\text{comm}}/t_1^{\text{feval}}$ is ratio of interprocessor communication time within one function evaluation to the time for this evaluation on 1 processor. For this model problem, $\alpha_p \approx c_1/p + c_2$ where platform dependent constants c_1 and c_2 are often known or can easily be measured.

In Figure 1, timings and speed-ups of both codes on the Cray T3E and IBM SP2 are presented. As we see, the speed-ups are significantly worse on the IBM SP2, and this is apparently due to the higher latency on the platform.

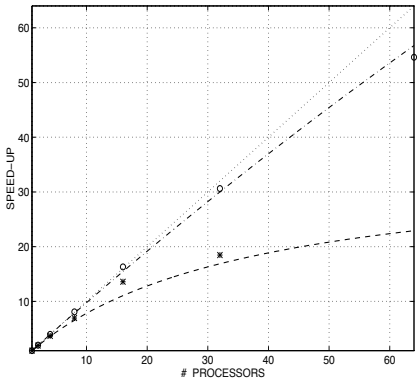
3 Conclusions

It has been shown that on the platforms as the IBM SP2 and Cray T3E the MRAI based codes possess parallelism of explicit schemes. Taking into account that stability properties of MRAI schemes are much better (recall CPU gain factor 3.2 for the 3D model problem), we conclude that MRAI technique is an attractive parallelizable time-stepping strategy.

Acknowledgments Author thanks Henk van der Vorst and Auke van der Ploeg for their valuable suggestions on parallelization issues.

This research was supported by the Netherlands organization for scientific research NWO, project 95MPR04.

EB/MRAI, speed-up



EB/MRAI and LSODE/MRAI, timings

#	EB/MRAI		LSODE/MRAI	
PEs	T3E	SP2	T3E	SP2
1	404.2	426.6	55.8	58.3
2	202.8	221.2	28.0	30.3
4	101.8	115.8	14.0	15.7
8	50.0	61.4	7.0	8.4
16	24.8	31.4	3.4	4.4
32	13.2	23.1	1.8	3.3
64	7.4	—	1.0	—

Fig. 1. Left: Speed-up results for the Cray T3E (predicted: `-·-`, observed: `o`) and IBM SP2 (predicted: `---`, observed: `*`) versus the ideal speed-up (dotted line). Right: timings (sec.) versus number of processors (# PEs)

References

1. Z. Bai, D. Hu, and L. Reichel. A Newton basis GMRES implementation. *IMA J. Numer. Anal.*, 14:563–581, 1991.
2. M. A. Botchev, G. L. G. Sleijpen, and H. A. van der Vorst. Stability control for approximate implicit time-stepping schemes with minimal residual iterations. Technical Report 1043, Department of Mathematics, Utrecht University, Dec. 1997.
3. G. D. Byrne, A. C. Hindmarsh, and P. N. Brown. Vodpk, large non-stiff or stiff ordinary differential equation initial-value problem solver. Available at `http://www.netlib.org`, 1997.
4. T. F. Chan and K. R. Jackson. The use of iterative linear-equation solvers in codes for large systems of stiff IVPs for ODEs. *SIAM J. Sci. Stat. Comput.*, 7(2):378–417, 1986.
5. E. de Sturler and H. A. van der Vorst. Reducing the effect of global communication in GMRES(m) and CG on parallel distributed memory computers. *J. Appl. Num. Math.*, 18:441–459, 1995.
6. C. W. Gear and Y. Saad. Iterative solution of linear equations in ODE codes. *SIAM J. Sci. Stat. Comput.*, 4(4):583–601, 1983.
7. Y. Saad and M. H. Schultz. GMRES: a generalized minimal residual algorithm for solving nonsymmetric linear systems. *SIAM J. Sci. Stat. Comput.*, 7(3):856–869, 1986.
8. B. P. Sommeijer, L. F. Shampine, and J. G. Verwer. RKC: An explicit solver for parabolic PDEs. Technical Report MAS-R9715, CWI Amsterdam, 1997.
9. H. A. van der Vorst and T. C. Chan. Linear system solvers: sparse iterative solvers. In D. E. Keyes, A. Samed, and V. Venkatakrishnan, editors, *Parallel Numerical Algorithms*, volume 4 of *ICASE/LaRC Interdisciplinary Series in Science and Engineering*, pages 91–118, Dordrecht, 1997. Kluwer Academic. See also Utrecht University, Department of Mathematics Preprint No. 869 (`http://www.math.ruu.nl/publications/`).

This article was processed using the LaTeXmacro package with LLNCS style