EQUADIFF 10

Edik A. Ayrjan; Shura Hayryan; Chin-Kun Hu; Imrich Pokorný; Igor V. Puzynin

Parallel realization of the finite difference method solution of the Poisson-Boltzmann equation

In: Jaromír Kuben and Jaromír Vosmanský (eds.): Equadiff 10, Czechoslovak International Conference on Differential Equations and Their Applications, Prague, August 27-31, 2001, [Part 2] Papers. Masaryk University, Brno, 2002. CD-ROM; a limited number of printed issues has been issued. pp. 17--20.

Persistent URL: http://dml.cz/dmlcz/700377

Terms of use:

© Institute of Mathematics AS CR, 2002

Institute of Mathematics of the Academy of Sciences of the Czech Republic provides access to digitized documents strictly for personal use. Each copy of any part of this document must contain these *Terms of use*.

This paper has been digitized, optimized for electronic delivery and stamped with digital signature within the project *DML-CZ: The Czech Digital Mathematics Library* http://project.dml.cz

Equadiff 10, August 27–31, 2001
Prague, Czech Republic

Equadiff 10 CD ROM
Papers, pp. 17–20

Parallel realization of the finite difference method solution of the Poisson-Boltzmann equation

Edik A. Ayrjan[1], Shura Hayryan[2], Chin-Kun Hu[3], Imrich Pokorný[4] and Igor V. Puzynin[5]

[1] Laboratory of Information Technology,
JINR, Dubna, 141980 Moscow region, Russia
Email: ayrjan@cv.jinr.ru
[2] Institute of Physics,
Academia Sinica, Nankang, Taipei 11529, Taiwan
Email: shura@stp1.phys.sinica.edu.tw
[3] Institute of Physics,
Academia Sinica, Nankang, Taipei 11529, Taiwan
Email: huck@phys.sinica.edu.tw
[4] Department of Metal Forming FM, TU in Košiciach,
Letná 9, 042 00, Košice, Slovakia
Email: Imrich.Pokorny@tuke.sk
[5] Laboratory of Information Technology,
JINR, Dubna, 141980 Moscow region, Russia
Email: puzynin@jinr.ru

Abstract. We demonstrate the parallel realization of finite difference solution of one problem by successive overrelaxation method (SOR) for the small number of processors. The solution was realized on supercomputer SPP2000 (Dubna, Russia).

MSC 2000. 65Y05, 65M06, 65M55

Keywords. Parallel programming, finite difference, successive overrelaxation method, Poisson-Boltzmann equation

The paper is related to with the elaboration of effective numerical algorithms for the multigrid solution of the following nonlinear Poisson-Boltzmann equation (PB) in the complex area:

$$\nabla[\varepsilon(\mathbf{r})\nabla\phi(\mathbf{r})] - \varepsilon(\mathbf{r})\kappa(\mathbf{r})^2 \sinh[\phi(\mathbf{r})] + \frac{4\pi\rho(\mathbf{r})}{kT} = 0.$$

This is the preliminary version of the paper.

This equation describes a model of the energy of the electrostatic field produced by the electrical charge placed at the center of some atoms of the protein molecule, where $\phi(\mathbf{r})$ is the dimensionless electrostatic potential in units of kT/q (k is the Boltzmann constant, T is the absolute temperature, and q is the charge on a proton), $\varepsilon(\mathbf{r})$ is the dielectric constant, and $\rho(\mathbf{r})$ is the fixed charge density (in proton charge units). The term $\kappa = 1/\lambda$, where λ is the Debye length. The variables $\phi, \varepsilon, \kappa$, and ρ are all functions of the position vector $\mathbf{r}$.

We solve this problem by the multigrid finite difference successive overrelaxation method (SOR). Iterations are stopped when the relative error in solution is less than some δ. Computations are organized using the multigrid approach by using the sequence of grids. This method provides a quick convergence as well as a better control and analysis of obtained approximations. The first problem is solved on the mesh with step size h=1A by SOR iteration process. The obtained solution is interpolated on the mesh with a half grid size and a new iteration is performed. Similarly the solution on the fine grid with a mesh size of 0.25A is obtained.

The program PBSOLVE is constructed for numerical solution of the linearized Poisson - Boltzmann equation. Finite difference discretization and SOR iterations [1] on the sequence of grids (multigrid) are used to obtain the approximate electrostatic potential on the grid. The error control is provided by comparison of solutions on nested grids. A parallel version of the PBSOLVE program was written in the programming language FORTRAN77 by using MPI (Message Passing Interface) [6], and we used the multiprocessor computer system *SP2000* with 8 processors.

The improvement of the program is related to the treatment of the solute-solvent interface boundary. Two ways are possible to improve the mapping of the interface boundary onto the grid. The former is a multilevel mesh refinement technique [2] . The method is intended for accurate electrostatic calculations over domains with local mesh refinement patches. The latter is the exploitation of well-developed finite [3] and boundary element [4] techniques to complex molecular surface. To account exactly the behavior of the potential on the infinity, the coupled boundary integral finite elements formulations of the problem [5] will be elaborated.

A natural way of parallelization is to divide the area into the p peaces by planes (we used horizontal ones because we used FORTRAN), where p is the number of used processors. First, the processors work step by step.

The other way of parallelization might be rearrange of cycles [7].

We obtain our results for the method inaccuracy 1, 0.25, and 0.0625; calculation inaccuracy - 10^{-4}; and the best time ratio is for the 3 processors (2.82). This ratio decreases for more than 3 processors because the number of data transmits increases. This result we obtained when our program had worked in the non-solo regime. For the solo regime we obtained the following interesting result (see Tab. 1. for the molecule with 295 atoms):

For the purpose of this paper we take the cubic solution area with the three equidistant grids (number of knots was 41, 81, and 161).

Our parameters and calculate times are shown in the following table:

N. of proc.	1	2	3	4	5	6	7	8
Ast. time*	850.98	461.94	331.93	263.50	210.89	191.80	169.29	172.86
User time*	848.88	459.43	328.82	261.10	208.47	188.94	165.81	155.74
** 1st grid	97	97	97	96	96	95	95	95
** 2nd grid	165	165	163	166	165	166	167	167
** 3rd grid	309	308	311	312	312	316	316	319
Time rel.	1.00	1.84	2.56	3.23	4.05	4.44	5.03	4.92

* The time is in sec.
** Number of iterations.

Tab. 1.

The astronomic time of calculation is shown in the second line of Tab. 1. This time was obtained by MPI function *MPI_Wtime()* which show the time between the start and the end of calculation by clock. Also this time is not a real calculation time of processors. A bit close to such time is a user time. We can obtain the user time by the function *itimes(i)* which was written by Sapozhnikov A. P. and Sapozhnikova T. F. in the programming language C. The difference between these times is based on the fact that our processor works with other system processes and sometimes it must wait.

Numbers of iterations for our grides are shown in the 4th, 5th, and 6th lines respectively. Let the number of iterations for one processor be n. By our method the numbers of iterations for p processors $(p \neq 1)$ can be at most $n+p-1$ for the good convergence. In the real process this number sometimes is decreased. It is natural because when the last processor makes i^{th} iteration the previous processors make $(i+j)^{th}$ iteration, where j is a difference of order numbers of the last processor and the actual one.

The result of solving process for one processor can be different from the result for several processors:

– by the reason of previous paragraph (some part of data is from the highest iteration - for each processor except the last),
– if the process converges slower for some data which are not in the last processor working area - the number of iterations can decrease.

By the last line of Tab. 1. we can see that the ratio of times is increasing.

This ratio is shown graphically on the following figure 1.

The correctness of the algorithm was tested by solving the PB equation for a single spherical charge for which the exact solution is known.

The performance of the algorithm was tested on the small peptide Metenkephalin.

Acknowledgment.
This research was supported by the grant VEGA 2/7192/20 and the grant RFFI 01-01-00726.

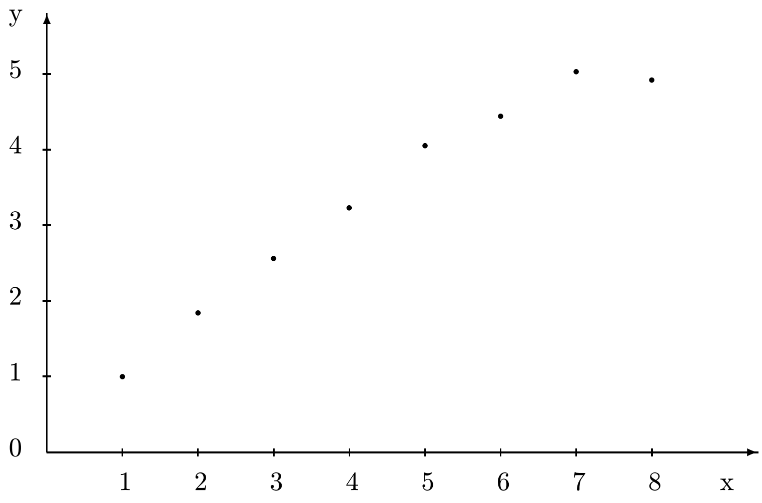

Fig. 1.

References

1. L. A. Hageman and D. M. Young, *Applied iterative methods*, Academic Press, New York, 1981.
2. D. Bai and A. Brandt, *SIAM J. Sci. Stat. Comput.*, **8**, 109, 1987.
3. O. C. Zienkiewicz, *The finite element method*, London, McGraw-Hill, 1977.
4. C. A. Brebbia, J. C. F. Telles, and L. C. Wrobel, *Boundary element techniques: theory and applications in engineering*, Springer-Verlag, 1984.
5. E. A. Ayrjan, E. P. Zhidkov a.o., *Numerical algorithms for accelerators magnetic systems calculation*, Particle Phys. Nucl, 1991.
6. M. Snir, S. W. Otto, S. Huss-Lederman, D. W. Walker, and J. Dongarra, *MPI: The complete reference*, The MIT Press, Massachusetts Institute of Technology, Cambridge (USA), 1997.
7. L. Gregushova, *Method of paralleling of cycles for multiconveier numerical systems*, (Russian), JINR, p11-87-900, Dubna, 1987.