Exploiting Symmetry in Parallel Computations for Structural Biology

Ioana M. Boier Martin[1,2] and Dan C. Marinescu[1]

[1] Purdue University, West Lafayette, IN 47907
[2] Current address: Indiana University, South Bend, IN 46634

Abstract. In this paper we outline a technique to carry out parallel computations for objects which exhibit a high degree of symmetry. We describe an algorithm used for the computation of the envelope of a spherical virus with icosahedral symmetry and discuss its performance. The algorithm is quite general and it is advantageous if folding back is computationally less intensive than the original computation.

1 Introduction

The 3-D atomic structure determination of viruses is a computationally intensive area of research which requires the fusion of biology and high performance computing. Parallel computers are needed for the data acquisition, data analysis, and the modeling phase of structure determination of large macromolecules such as viruses.

While the structure determination of proteins which contain tens to hundreds of residues is common today, the determination of the structure of viruses is still very challenging. Viruses are fairly large proteins, typically they contain tens of thousands of residues and have millions of atoms. Spherical viruses are of special interest. They consist of a large number of identical, asymmetric units related by symmetry.

In this paper we investigate means to exploit this symmetry to reduce the amount of computations required for the structure determination of spherical viruses. We discuss parallel algorithms used in structure determination and examine an idea of Michael Rossmann [3] to exploit the symmetry of spherical viruses for the computations needed for phase refinement and extension [1].

2 Electron Density Averaging for Computing the Molecular Envelope of a Virus

In 1963 Michael Rossmann and David Blow suggested the Molecular Replacement Method for determining the phase of the structure factors in X-ray crystallography [2]. One starts with a low resolution model of a virus, e.g. hollow spheres, and the measured amplitudes of the structure factors. The symmetry of the virus is used to replace the electron density at every point with an averaged

one among all points related by non-crystallographic symmetry in an iterative process known as phase refinement and extension [1].

In this paper we compare the single and double interpolation algorithms for averaging suggested by Michael Rossmann [3]. The basic idea of the double interpolation algorithm is straightforward and can be applied to other problems exhibiting a high degree of symmetry: carry out the required computations only for the unit repeating itself throughout the data space, then propagate the results to the entire data space by folding back the points throughout the space to the unit that has been updated. This algorithm is advantageous if folding back requires fewer operations than the original computation.

We do expect that the double interpolation algorithm converges slower than the single interpolation. For a sequential algorithm, the double interpolation is better than the single interpolation, but this is not necessarily true for a parallel implementation of the algorithm.

3 Implementation and Results

We report here the results obtained for a parallel implementation of the double interpolation algorithm described in the previous section. First, we observe that one needs to carry out averaging for a region slightly larger than the asymmetric unit. Figure 1 shows that folding back a grid point A_{out} outside the asymmetric unit may map into a point A_{in} inside the asymmetric unit. To determine the electron density for A_{in} may require the knowledge of the electron density of a grid point A' outside the asymmetric unit, but inside the border region.

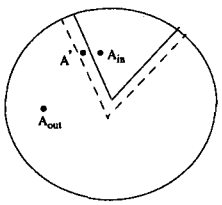

Fig. 1. The need to carry out the averaging process for the border region outside the asymmetric unit.

With this observation the algorithm for the double interpolation consists of three phases:

Phase 1. Identify all protein points inside the asymmetric unit and the border region.
Phase 2. Carry out the averaging for all the grid points marked as being inside.
Phase 3. Fold back all grid points marked as being outside.

To parallelize this algorithm, we use a strategy of dividing the entire data space into Data Allocation Units (DAUs) and carry out the computation required at each phase for all the DAUs assigned to a given Processing Element, PE. Unfortunately, there are two significant sources of inefficiencies for the parallel double interpolation algorithm. First, each phase of the algorithm has to be preceded by a load balancing computation and data has to be redistributed accordingly. The second source of inefficiency is the global synchronization required before the beginning of phases 2 and 3. By comparison, the single interpolation algorithm is embarrassingly parallel, each node is assigned only once the workload and no synchronization is necessary.

To compare the single and double interpolation procedures, we used data for the HRV16, the Human Rhinovirus 16. The unit cell for HRV16 has the following dimensions: $a = 362.49$ Å, $b = 347.11$ Å, $c = 334.75$ Å, $\alpha = \beta = \gamma = 90°$. The number of grid points are $nx = 360$, $ny = 352$, and $nz = 336$.

The execution time for several cycles of averaging for the single and the double interpolation method are summarized in Table 1. The results obtained on a 64 nodes partition of a Paragon system are quite disappointing and indicate that the single interpolation takes less time than the double interpolation.

As suggested earlier, the double interpolation is expected to converge slower than the single interpolation and this is confirmed by the results in Figure 2.

	Double Interpolation				Single Interpolation
Cycle	Phase 1	Phase 2	Phase 3	Total	
1	78	37	131	248	132
2	74	39	119	232	127
3	72	39	116	227	127
4	109	41	136	286	130
5	86	40	130	256	131

Table 1. The execution time (seconds) of several cycles of electron density averaging for the single and the double interpolation algorithms.

At the present time, we are implementing an optimized version of the double interpolation algorithm. In this version the first phase of the algorithm and the load balancing procedures for phase two and three are carried out only during the first iteration and their results are reused during subsequent iterations. Even in this case it will be very difficult for the double interpolation to compete with the single interpolation simply because of its considerably slower convergence rate. In addition to the global synchronization required at the end of phase 2, an inherent problem of the parallel double interpolation algorithm is that the size of the problem to be solved in phase 2 is considerably smaller than that of phase 3. In other words, the degree of parallelism and consequently, the optimal number of PEs for phase 2 and 3 are different $P_2 << P_3$.

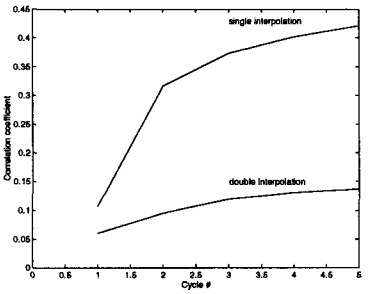

Fig. 2. The convergence rate of the single and the double interpolation method. The convergence rate is determined by the correlation coefficient of the phases of the structure factors calculated during two consecutive iterations.

4 Conclusions

To carry out computations for objects which exhibit a high degree of symmetry one can first transform the asymmetric unit which repeats itself throughout the object and then propagate the results by symmetry. An algorithm based upon these ideas requires (a) to identify the asymmetric unit, (b) to carry out the transformations within the asymmetric unit, and (c) to map every point in the data space back into the asymmetric unit. The more complex the transformations required in step (b) compared with folding back required by step (c) are, the more advantageous the algorithm is.

Acknowledgments

The authors express their gratitude to Michael Rossmann who has provided insight into structural biology computations. This research is supported in part by National Science Foundation grants BIR-9301210 and MCR-9527131, by the Scalable I/O Initiative and by a grant from the Intel Corporation.

References

1. M.C. Cornea-Hasegan, Z. Zhang, R.E. Lynch, D.C. Marinescu, A. Hadfield, J.K. Muckelbauer, S. Munshi, L. Tong and M.G. Rossmann, Phase Refinement and Extension by Means of Non-crystallographic Symmetry Averaging using Parallel Computers, *Acta Cryst.*, **D51**, pp. 749–759, 1995.
2. Rossmann, M.G., The Molecular Replacement Method, *Gordon and Breach*, 1972.
3. Rossmann, M.G., Private Communication, 1995.
4. Rossmann M.G., McKenna R., Tong L., Xia D., Dai J., Wu H., Choi H.K., Marinescu D.C and Lynch R.E., Molecular Replacement, *Proceedings of the CCP4 Study Weekend*, **31**, January - February 1992, edited by E. Dodson, S. Gover and W. Wolf, pp. 33–48, Daresbury, England: SERC.