INL/CON-11-20999
PREPRINT

Parallelization of the Red-Black Algorithm on Solving the Second-Order P_N Transport Equation with the Hybrid Finite Element Method

2011 ANS Annual Meeting

Yaqi Wang
Cristian Rabiti
Giuseppe Palmiotti

June 2011

The INL is a
U.S. Department of Energy
National Laboratory
operated by
Battelle Energy Alliance

Parallelization of the Red-Black Algorithm on Solving the Second-Order P_N Transport Equation with the Hybrid Finite Element Method

Yaqi Wang, Cristian Rabiti, Giuseppe Palmiotti

Idaho National Laboratory, P.O. Box 1625, Idaho Falls, ID 83415-3870
Email: Yaqi.Wang@inl.gov,Cristian.Rabiti@inl.gov,Giuseppe.Palmiotti@inl.gov

INTRODUCTION

The Red-Black algorithm has been successfully applied on solving the second-order parity transport equation with the P_N approximation in angle and the Hybrid Finite Element Method (HFEM) in space, i.e., the Variational Nodal Method (VNM) [1,2,3,4,5]. Any transport solving techniques, including the Red-Black algorithm, need to be parallelized in order to take the advantage of the development of supercomputers with multiple processors for the advanced modeling and simulation. To our knowledge, an attempt [6] was done to parallelize it, but it was devoted only to the z axis plans in three-dimensional calculations. General parallelization of the Red-Black algorithm with the spatial domain decomposition has not been reported in the literature. In this summary, we present our implementation of the parallelization of the Red-Black algorithm and its efficiency results.

RED-BLACK ALGORITHM AND ITS PARALLELIZATION

Because the parallel communications happen only on the level of the in-group solving, we can present the implementation in the context of the steady-state one-speed fixed source transport equation,

$$L\Psi = s, \tag{1}$$

where L is the transport operation including the streaming, collision and in-group scattering. Ψ is the angular flux and s is the source including the external, fission and cross-group scatterings. Given a mesh $\mathbb{T}$ covered with non-overlapping cells, with P_N and HFEM the equation can be transformed into a formula,

$$\mathbf{j} = \mathbf{R}\mathbf{j} + \mathbf{s}_{RB}, \tag{2}$$

where $\mathbf{j}$ is the partial currents with all spatial and angular moments defined on the cell interfaces. $\mathbf{R}$ is called the response matrix which depends on the geometry of the mesh and the macroscopic cross sections. $\mathbf{s}_{RB}$ is the result of the linear operation of the source term s in the Eq. (1). The details on how these response matrices and the red-black source are constructed can be found in Ref [3].

Consider any cell K in the mesh $\mathbb{T}$, Eq. (2) is

$$\mathbf{j}_K^{out} = \mathbf{R}_K \mathbf{j}_K^{in} + \mathbf{s}_{K,RB}, \tag{3}$$

where $\mathbf{j}_K^{out}$, $\mathbf{j}_K^{in}$ are out-going and in-coming partial currents defined on all the faces of the cell. The number of angular moments is $(P_N+1)^2/4$ in 2D and $P_N*(P_N+1)/2$ in 3D, where P_N is an odd number. The number of spatial moments is p+1 for a line segment in 2D and $(p+1)^2$ for a quadrilateral face in 3D when the product shape function is used. The equation states that given the incoming partial currents of a cell, its out-going partial current can be calculated with the cell response matrix.

When a face E of a cell is part of the vacuum boundary, the in-coming and out-going partial current have the following relation,

$$\mathbf{j}_E^{out} = \mathbf{N}\mathbf{j}_E^{in}, \tag{4}$$

where $\mathbf{N}$ is a constant matrix irrelevant of the orientation of the face. When a face E of a cell is part of the reflecting boundary, the in-coming and out-going partial current have the following relation,

$$\begin{aligned} \mathbf{j}_{E,e}^{out} &= \mathbf{j}_{E,e}^{in} \\ \mathbf{j}_{E,o}^{out} &= -\mathbf{j}_{E,o}^{in} \end{aligned}, \tag{5}$$

where the subscript e and o represent the even and odd moments in azimuthal angle. Eq. (2) through (5) give the complete equation set with the vacuum and reflecting boundary conditions in terms of interface partial currents. For a example of a 2D Cartesian mesh with uniform P_3 and linear expansion, the length of the solution vector is the number of elements times 4 faces, 4 angular moments, and 2 spatial moments.

When solving Eq. (2), we first paint all cells in the mesh with colors so that every pair of cells sharing a non-zero interface has different colors. For a regular Cartesian geometry, two colors, red and black, will be sufficient, which is how the name of the algorithm comes from. Note that we want to use a minimum number of colors to paint the domain. Then, we can loop through all colors with one red-black iteration to update the local out-going partial currents of all cells with the same color. We continue the red-black iterations until the convergence criteria is reached. This solving technique is referred as the Red-Black algorithm.

We immediately can identify the few tasks needed for parallelization with the spatial domain decomposition:

1. Painting the domain. This can be done on the master processor before the mesh is distributed into processors if there is no later mesh modifications such as refinement. Repainting the mesh with all processors

participating is still trivial if the optimal painting is not our objective;

2. The variable that need to be communicated are the partial currents defined over the processor interfaces;

3. Communication needs to be done after each color is swept, if we want to preserve the serial Red-Black iteration. After a sweep of one color is finished, each processor sends the solutions on the processor interfaces connected to the cells with the color to its neighboring processors possibly with N_{color} *-1* slices and it receives solutions from the neighboring processors.

The total number of collective communications will be equal to the number of iterations times the number of colors. The number of communications can be minimized with the minimum painting colors. Memory needs to be allocated in the way that all these communications can be done in place, i.e., no communication buffers are required.

Implementing-wise, each face is defined with an orientation, and $\mathbf{j}^+$ and $\mathbf{j}^-$ are defined over all faces with respect to these orientations. Only on the processor interfaces, $\mathbf{j}^+$ is defined as $\mathbf{j}^{out}$ and $\mathbf{j}^-$ is defined as $\mathbf{j}^{in}$ regardless of the orientation of the face. The solutions on the processor interfaces are allocated so that all variables with the same color are contiguous. When loop through all elements is carried out, the faces and its location connecting the elements can be obtained as the local face solution vectors. Note that no continuity constrain is put on the local face solutions; we refer to this type of solution as the L2 variable comparing the variable in the continuous space. This implementation can be integrated into a mesh framework so that the variable initialization and communication can be done in one subroutine call and it can be possibly used in other applications.

RESULTS AND DISCUSSIONS

The idea presented in the above section is implemented with modern Fortran at INL in the code INSTANT [8] . INSTANT includes a mesh handler to facilitate calculations on triangular, hexagonal and Cartesian geometries in both 2D and 3D and has been parallelized with the domain decomposition with MPI. The simple 2D 2-group Wagner benchmark problem is used to test the implementation, which can be found in Ref [7]. 36*36 0.5cm*0.5cm cells with quadratic shape functions are used to discretize the problem. Calculations with the standard power iteration and the red-black algorithm as the in-group solver have been performed with different number of processors. We carried out the calculations on a desktop with 12 Intel X5680 processors at 3.33GHz and MPICH2 with Windows XP Pro. Two calculations with P_9 and P_{15} were performed.

K_{eff} with P_{15} is 0.9015008 and 0.9014694 with P_9. With P_9, there are a total of 803*2 collective communications in 16 power iterations while there are 876*2 collective communications in 16 power iterations with P_{15}. The computing times are summarized in Table 1 and Table 2. Note that the time spent on assembling all the response matrices is not included in these times. However, if the number of response matrices is equal to the number of cells, there will be perfect scaling on the flops of this part.

We can see the very good scaling with the domain decomposition on this platform. It is clear that the fraction of time spent in the communications increases with the increasing number of processors. One option to reduce their cost is to move the communications out of the color loop.

ACKNOWLEDGMENTS

Prepared for the U.S. Department of Energy through the INL LDRD Program under DOE Idaho Operations Office Contract DE-AC07-05ID14517.

REFERENCES

1. I. Dilber and E. E. Lewis, "Variational Nodal Methods for Neutron Transport". *Nucl. Sci. Eng.*, **Vol. 91**, p. 132 (1985).

2. C. B. Carrico, E. E. Lewis and G. Palmiotti, "Three-Dimensional Variational Nodal Transport Methods for Cartesian, Triangular, and Hexagonal Criticality Calculations". *Nucl. Sci. Eng.*, **Vol. 111**, p. 168 (1992).

3. G. Palmiotti, E. E. Lewis and C. B. Carrico, "VARIANT: VARIational Anisotropic Nodal Transport for Multidimensional Cartesian and Hexagonal Geometry Calculation". ANL-95/40, Argonne National Laboratory (1995).

4. J. Y. Doriath, "Reactor Analysis Using a Variational Nodal Method Implemented in the ERANOS System". *Proc. Topl. Mtg. Advances in Reactor Physics.*, Knoxville, Tennessee, April 11-15, 1994, p. 20 (1994).

5. E. E. Lewis, and G. Palmiotti, "Red-Black Response Matrix Acceleration by Transformation of Interface Variables". *Nucl. Sci. Eng.*, **Vol. 130**, p181-193 (1998).

6. U. R. Hanebutte, H. S. Khalil, G. Palmiotti, M. Tatsumi, "Development of a Parallelization Strategy for the VARIANT Code," *Trans. ANS* **76**, 159 June 1-5, 1997 Orlando, FL.

7. M. A. Smith, G. Palmiotti, E. E. Lewis and N. Tsoulfanidis, "An Integral Form of the Variational Nodal Method", *Nucl. Sci. Eng.*, Vol. 146, p141-151 (2004).

8. Y. Wang, C. Rabiti, G. Palmiotti, "Krylov Solvers Preconditioned with the Low-Order Red-Black Algorithm for the PN Hybrid FEM for the INSTANT Code", the International Conference on Mathematics and Computational Methods applied to Nuclear Science and Engineering, May 8-12, Rio de Janeiro, Brazil, 2011

Table 1. Computing time on a single desktop with P_9

Number of processors	Total computing time (s)	Speed up
1	35.70	1.00
2	17.98	1.99
3	11.11	3.21
4	8.39	4.26
6	5.93	6.02
9	4.03	8.86

Table 2. Computing time on a single desktop with P_{15}

Number of processors	Total computing time (s)	Speed up
1	216.78	1.00
2	108.57	2.00
3	74.42	2.91
4	54.42	3.98
6	38.11	5.69
9	29.55	7.34