Joint International Topical Meeting on Mathematics & Computation and Supercomputing in Nuclear Applications (M&C + SNA 2007)
Monterey, California, April 15-19, 2007, on CD-ROM, American Nuclear Society, LaGrange Park, IL (2007)

COMPARISON OF LINEAR AND QUADRATIC DISCONTINUOUS SPATIAL FINITE ELEMENT METHODS FOR PARALLEL S_N TRANSPORT ON TRIANGLES

Jae H. Chang and James S. Warsa
Los Alamos National Laboratory
P.O. Box 1663, MS D409, Los Alamos, NM 87545, USA
jhchang@lanl.gov; warsa@lanl.gov

ABSTRACT

We compare the performance of linear and quadratic discontinuous finite elements methods spatial discretizations (LDFEM and QDFEM) of the S_N transport equation on triangles in two-dimensional, axisymmetric coordinate system (*r-z*). Numerical results reveal that both unlumped and fully lumped LDFEM are second order accurate as expected. The QDFEM exhibits third order accuracy. Because the QDFEM is third order accurate, it obtains same level of error reduction as LDFEM with a smaller mesh, i.e., coarser mesh cells. Therefore, QDFEM requires less memory and solves the problem faster than LDFEM. Numerical results show that, for optically thick problems, parallel block-Jacobi algorithm reduces CPU times when compared to full parallel sweeps for both LDFEM and QDFEM.

Key Words: discontinuous finite element methods, mass and gradient lumping, parallel block-Jacobi

1. INTRODUCTION

Linear finite elements are often chosen because of their low computational cost and second order accuracy. However, LDFEM discretization can produce un-physical negative fluxes. Therefore LDFEM is often lumped in some way in order to reduce the stencil of the discretization to improve solution positivity while maintaining second order accuracy, albeit with an increased error relative to the unlumped method. Quadratic elements are third order accurate and should be less likely to produce negative fluxes, as it is the case in one dimension, an important consideration because the method can not be lumped in *r-z* geometry.

There is a complicated relationship betwixt the numerical properties of the various methods as well as their impact on computational effort. For example, a natural question to ask is how does the computational effort required to achieve a desired level of accuracy or to ensure positivity depend on the choice of finite element. Another example would be to ask how the choice of finite elements affects the linear algebraic properties of the discretized operator and how is it reflected in the convergence rate and computational effort of a particular iterative method. Another question is how the properties impact scaling of a parallel implementation. The answers to these and other questions are likely to be problem dependent and can therefore never be answered definitively. However, we can begin to make progress toward answering some of these important questions. Our purpose here, for example, will be to present measurements of the

order of accuracy as a function of problem size and computational effort for LDFEM and QDFEM spatial discretizations, for a limited set of problems. We will compare the methods with respect to parallel scaling and two iterative solution techniques, full parallel sweeps and parallel block-Jacobi [1]. We leave consideration of positivity and other numerical properties for future work.

2. THE S_N EQUATIONS

If we assume a properly normalized quadrature set with N_a ordinate directions, the two-dimensional S_N transport equation in axisymmetric coordinates is

$$\frac{\mu_m}{r}\frac{\partial(r\psi_m)}{\partial r}+\xi_m\frac{\partial\psi_m}{\partial z}-\frac{1}{r}\frac{\partial(\eta\psi_m)}{\partial\omega}+\sigma_t(r,z)\psi_m(r,z)=\frac{\sigma_s(r,z)}{4\pi}\sum_{m'=1}^{N_a}w_{m'}\psi_{m'}(r,z)+Q_m(r,z). \quad (1)$$

The angular derivative in Eq. (1) must be defined in terms of the quadrature before we discretized the equation in curvilinear coordinate system. We use angular weighted diamond difference (WDD) formulation defined by Morel and Montry [2].

In cylindrical coordinates, the quadrature is arranged on levels of constant ξ. In two dimensions, there are N levels for an S_N quadrature and the azimuthal angle ω for level ξ_l is discretized into angular intervals in order to approximate the derivative with respect to ω using a weighted difference of the angular fluxes at the edges of the angular interval. In other words, for an quadrature point m on level l, we make the following approximation

$$\frac{\partial(\eta\psi)}{\partial\omega}\approx\frac{1}{w_m}\left(\alpha_{m+1/2}\psi_{m+1/2}-\alpha_{m-1/2}\psi_{m-1/2}\right). \quad (2)$$

A weighted average of the angular interval edge angular fluxes,

$$\psi_m=(1-\tau_m)\psi_{m-1/2}+\tau_m\psi_{m+1/2}, \quad (3)$$

then defines the angular flux for quadrature point m. In an implementation, the WDD weighted average is used to write $\psi_{m+1/2}$ in terms of $\psi_{m-1/2}$ and ψ_m in the angular derivative approximations. The angular fluxes for each point m can then be computed sequentially (for each level ξ_l) using the previously calculated angular cell interval angular flux $\psi_{m-1/2}$.

The WDD approximation is completed by defining weights for both derivative approximations and quadrature point weighted averages. For every quadrature point on level ξ_l, with quadrature weights that sum to C_l on that level the definitions are given by the following equations

$$\alpha_{m+1/2}=\alpha_{m-1/2}+\mu_m w_m, \quad \alpha_{1/2}=0, \quad (4)$$

$$\omega_{m+1/2} = \omega_{m-1/2} - \pi C_l w_m, \quad \omega_{1/2} = \pi, \tag{5}$$

$$\mu_{m+1/2} = \left(1-\xi_l^2\right)^{1/2} \cos\left(\omega_{m+1/2}\right), \tag{6}$$

$$\tau_m = \frac{\left(\mu_m - \mu_{m-1/2}\right)}{\left(\mu_{m+1/2} - \mu_{m-1/2}\right)}. \tag{7}$$

In order to start the WDD approximation, starting direction fluxes, with zero quadrature weight, must be calculated at the boundary of the first angular interval. For every ξ level, the starting direction solution is computed in Cartesian coordinates for the direction vector $\hat{\Omega}_S = \left(\mu_S, 0, \xi_l\right)$, with $\mu_S = -1\left(1-\xi_l^2\right)^{1/2}$ ($\mu_S < 0$ is chosen arbitrary). In two dimensions, the equation for the starting direction angular flux, $\psi_{S,l}\left(r,z,\hat{\Omega}_S\right)$, is

$$\mu_S \frac{\partial \psi_{S,l}}{\partial r} + \xi_l \frac{\partial \psi_{S,l}}{\partial z} + \sigma_t\left(r,z\right)\psi_{S,l} = \frac{\sigma_s\left(r,z\right)}{4\pi}\sum_{m'=1}^{N_a} w_{m'}\psi_{m'}\left(r,z\right) + Q\left(r,z,\hat{\Omega}_S\right). \tag{8}$$

3. DISCONTINUOUS FEM DISCRETIZATION

We start the description of the DFEM spatial discretization in a general spatial coordinate system by writing the S_N equations in the form

$$\hat{\Omega}_m \bullet \nabla \psi_m + \sigma_t \psi_m\left(\mathbf{r}\right) = S_m\left(\mathbf{r}\right), \tag{9}$$

where $S_m\left(\mathbf{r}\right)$ represents the scattering source plus inhomogeneous sources. The angular flux $\psi_m\left(\mathbf{r}\right)$ is first expanded in a nodal basis on a *p*-node element $\mathbb{E}$

$$\hat{\psi}_m\left(\mathbf{r}\right) = \sum_{j=1}^{p} \psi_{m,j} B_j\left(\mathbf{r}\right), \qquad \mathbf{r} \in \mathbb{E}\backslash\partial\mathbb{E}. \tag{10}$$

To introduce the upwind discontinuous approximation, we first define the indexing function for the k^{th} "face" of an element, $\partial\mathbb{E}_k$,

$$v\left(k\right) = \left\{ j \mid \mathbf{r}_j \in \partial\mathbb{E}_k \right\}, \tag{11}$$

or the set of all vertices on $\partial\mathbb{E}_k$. The up-winded angular flux on *k* is then defined by

$$\tilde{\psi}_m^{(k)}(\mathbf{r}) = \begin{cases} \psi_{m,v(k)}^{B} B_{v(k)}(\mathbf{r}), & \mathbf{r} \in \Gamma \setminus \partial\Gamma \\ \psi_{m,v(k)}^{INC} B_{v(k)}(\mathbf{r}), & \mathbf{r} \in \partial\Gamma \end{cases} \tag{12}$$

where

$$\psi_{m,i}^{B} = \begin{cases} \psi_{m,i,} & \left(\hat{n}_k \bullet \hat{\Omega}_m\right) > 0, \\ \psi_{m,i,}(k), & \left(\hat{n}_k \bullet \hat{\Omega}_m\right) > 0. \end{cases} \tag{13}$$

$\psi_{m,i}^{INC}$ is specified by boundary conditions for the problem domain Γ at vertex i and $i(k)$ refers to the vertex in the element sharing face k across vertex i, with N_f being the number of faces of an element such that

$$\partial\mathbb{E} = \bigcup_{k=1} \partial\mathbb{E}_k. \tag{14}$$

The weak form of Eq. (9) is then constructed for each basis function $B_i(\mathbf{r})$

$$\begin{aligned} \sum_{k=1}^{N_f} \int_{\partial\mathbb{E}_k} \left(\hat{n}_k \bullet \hat{\Omega}_m\right) B_i(\mathbf{r}) \tilde{\psi}_m^{(k)}(\mathbf{r})\, dA - \int_{\mathbb{E}} \left(\hat{\Omega}_m \bullet \nabla B_i(\mathbf{r})\right) \hat{\psi}_m(\mathbf{r})\, dV \\ + \sigma_t(\mathbf{r}) \int_{\mathbb{E}} B_i(\mathbf{r}) \hat{\psi}_m(\mathbf{r})\, dV = \int_{\mathbb{E}} B_i(\mathbf{r}) \hat{S}_m(\mathbf{r})\, dV, \end{aligned} \tag{15}$$

where $S_m(\mathbf{r})$ has been expanded in the same nodal basis as $\psi_m(\mathbf{r})$, that is,

$$\hat{S}_m(\mathbf{r}) = \sum_{j=1}^{p} \hat{S}_{m,j} B_j(\mathbf{r}). \tag{16}$$

The fully discrete equation for a ξ level l in axisymmetric coordinates is given by

$$\begin{aligned} \sum_{k=1}^{N_f} \left(\hat{\Omega}_m \bullet \bar{\mathbf{N}}^{(k)}\right) \tilde{\Psi}_m^{(k)} - \left(\hat{\Omega}_m \bullet \bar{\mathbf{L}}\right) \Psi_m - \frac{1}{w_m} \alpha_{m+1/2} \frac{1}{\tau_m} \mathbf{P} \Psi_m + \sigma_t \mathbf{M} \Psi_m \\ = \frac{1}{w_m} \left[\alpha_{m+1/2} \frac{(1-\tau_m)}{\tau_m} + \alpha_{m-1/2} \right] \mathbf{P} \Psi_{m-1/2} + \mathbf{MS}, \end{aligned} \tag{17}$$

where the operators are defined for row i and column j or column $v(k)$

$$\bar{\mathbf{N}}_{i,v(k)}^{(k)} = \int_{\partial\mathbb{E}_k} \hat{n}_k r B_i(\mathbf{r}) B_{(k)}(\mathbf{r})\, dA, \tag{18}$$

$$\bar{\mathbf{L}}_{i,j} = \int_{\mathbb{E}} r\left[\nabla B_i(\mathbf{r})\right] B_j(\mathbf{r})\, dV, \tag{19}$$

$$\mathbf{M}_{i,j} = \int_{\mathbb{E}} r B_i(\mathbf{r}) B_j(\mathbf{r})\, dV, \tag{20}$$

$$\mathbf{P}_{i,j} = \int_{\mathbb{E}} B_i(\mathbf{r}) B_j(\mathbf{r})\, dV. \tag{21}$$

We incorporate the standard linear and quadratic basis functions for triangles [3].

4. GENERALIZED LUMPING

Lumping may be necessary for optically thick cells. Mass matrix lumping is simply defined by setting

$$\mathbf{M}_{i,j} = \delta_{i,j} \int_{\mathbb{E}} r B_i(\mathbf{r}) dV, \tag{22}$$

where $\delta_{i,j}$ is the Kronecker delta function. The mass matrix cannot be lumped on quadratic triangular finite elements. Furthermore, lumping of the gradient term may improve the solution positivity if done carefully to preserve the diffusion limit and retain second order spatial convergence. We incorporate a generalized lumping of the gradient term described by Morel and Warsa for quadrilaterals [4], which results in the twice-integrated-by-parts gradient term

$$\sum_{k=1}^{N_f} \left(\psi^B_{m,i} - \psi_{m,i}\right) \delta_{i,v(k)} \int_{\partial\mathbb{E}_k} \left(\hat{n}_k \bullet \hat{\Omega}_m\right) r B_i(\mathbf{r})\, dA + \int_{\mathbb{E}} B_i(\mathbf{r}) \left(\hat{\Omega}_m \bullet \nabla\left[r\hat{\psi}_m(\mathbf{r})\right]\right) dV, \tag{23}$$

where ∇ is again the Cartesian gradient. Noting that $\hat{\Omega}_m \bullet \nabla r = \mu_m$ and $\hat{\Omega}_m \bullet \nabla\left(r^2\right) = 2\mu_m r,$ we can re-write the last term in Eq. (23) as

$$\begin{aligned} \int_{\mathbb{E}} B_i(\mathbf{r}) \left(\hat{\Omega}_m \bullet \nabla\left[r\hat{\psi}_m(\mathbf{r})\right]\right) dV &= \int_{\mathbb{E}} r B_i(\mathbf{r}) \left(\hat{\Omega}_m \bullet \nabla \hat{\psi}_m(\mathbf{r})\right) dV \\ &\quad + \mu_m \int_{\mathbb{E}} B_i(\mathbf{r}) \hat{\psi}_m(\mathbf{r})\, dV. \end{aligned} \tag{24}$$

Therefore, the mass and gradient lumped version of Eq. (17) is given by substituting Eq. (20) with Eq. (22) and Eqs. (18), (19), and (21) with

$$\bar{\mathbf{N}}^{(k)}_{i,v(k)} = \delta_{i,v(k)} \int_{\partial\mathbb{E}_k} \hat{n}_k r B_i(\mathbf{r})\, dA, \tag{25}$$

$$\bar{\mathbf{L}}_{i,j} = \int_{\mathbb{E}} r\, B_i(\mathbf{r}) \left[\nabla B_j(\mathbf{r}) \right] dV, \tag{26}$$

$$\mathbf{P}_{i,j} = \delta_{i,j} \int_{\mathbb{E}} B_i(\mathbf{r}) dV. \tag{27}$$

5. NUMERICAL RESULTS

In this section, we examine the performance of the unlumped and lumped LDFEM and QDFEM methods, which are incorporated in a parallel, S_N transport code. We use an isotropic manufactured solution on the domain $r \in [0,1]$ cm and $z \in [0,1]$ cm:

$$\psi(r,z) = 4\pi \sin(\pi z) \left[\sin(\pi r) + (1-r) \right]. \tag{28}$$

This solution satisfies the cylindrical symmetry condition at $r = 0$ and vacuum conditions on the remaining domain boundaries.

We first determine the order of accuracy for a pure absorber transport problem: σ_t = 1 cm^{-1}, σ_a = 1 cm^{-1}, a constant isotropic distributed source of 1 cm^{-3}/s, and S4 level symmetric quadrature. We solved the transport problem using source iteration with tolerance set to 10^{-12}. In Fig. 1, we plot the L_2 error of the cell-averaged scalar flux as a function of number of cells (1/h) for LDFEM, lumped LDFEM, and QDFEM. Both LDFEM and lumped LDFEM are second order accurate while the QDFEM exhibits third order accuracy. Although QDFEM requires twice the number of unknowns as LDFEM, QDFEM requires less memory than LDFEM for a given level of accuracy. For this problem lumped LDFEM requires a mesh size of 256 x 256 cells for an error of 10^{-5} while QDFEM requires only a 16 x 16 mesh. Therefore, QDFEM requires 128 times less memory than LDFEM. Furthermore, to reduce the error to 10^{-6}, QDFEM needs 512 times less memory than LDFEM methods.

Next, we compare LDFEM, lumped LDFEM and QDFEM with respect to parallel scaling. All the properties are identical as above except we increased the scattering ratio to 0.9(σ_s = 0.9 cm^{-1}) and we ran on a 1024 x 1024 mesh. In Table I, we show the number of iterations and CPU times for various number of processors for each spatial discretization. Table I includes both full parallel sweeps and parallel block-Jacobi algorithms.

Except for one processor (serial), where full sweeps are identical to block-Jacobi, the full parallel sweep algorithm is twice as fast as parallel block-Jacobi algorithm for all three DFEM methods. It should be noted that the number of iterations stays constant for the full parallel sweep algorithm when the number of processors increase. However, for the parallel block-Jacobi algorithm, the number of iterations increases as the number of processors increases. For both parallel algorithms the speed-up is approximately 1.6, even though number of iterations increases for the parallel block-Jacobi algorithm.

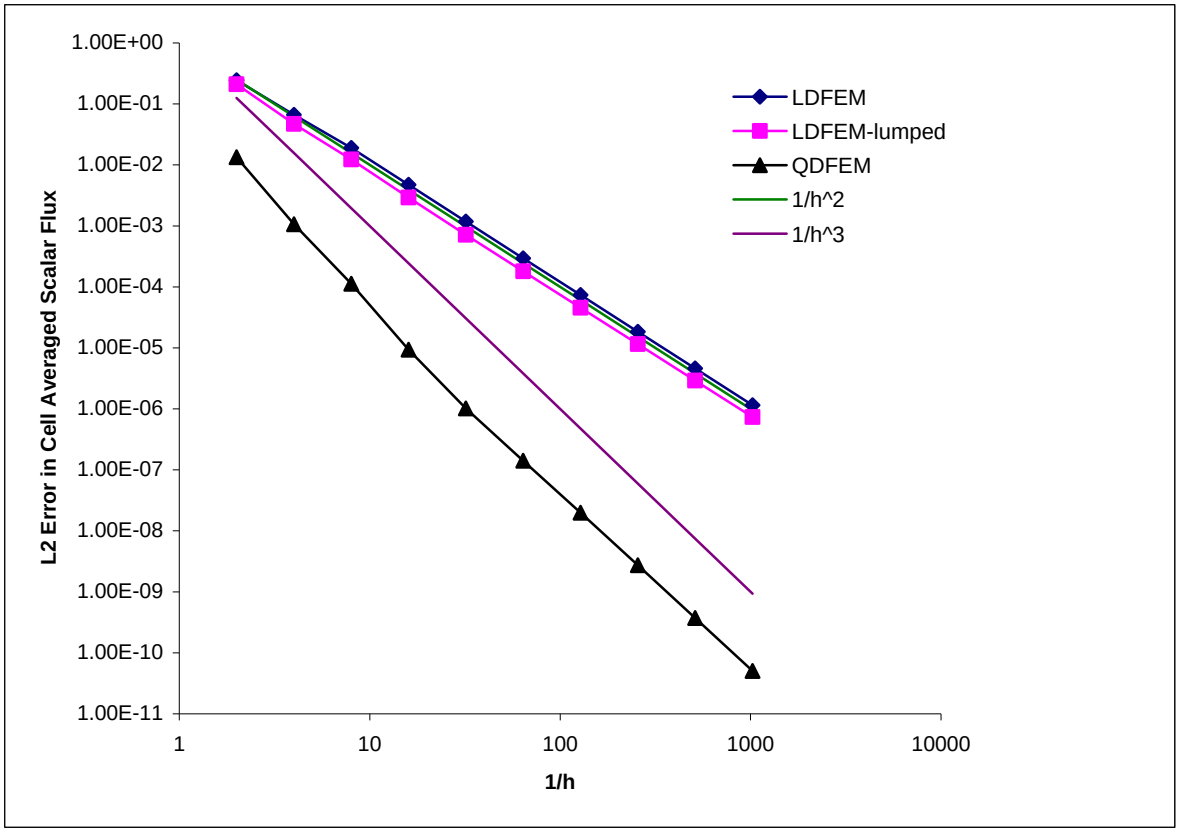

Figure 1. Pure absorber convergence plot.

Table I. Numerical results for σ_t = 1 cm^{-1}, σ_s = 0.9 cm^{-1}, 1024 x 1024 cells

	Full Sweeps						Block-Jacobi					
	LDFEM		LDFEM-lumped		Quadratic		LDFEM		LDFEM-lumped		Quadratic	
Processors	Iterations	CPU(s)	iterations	CPU(s)	iterations	CPU(s)	iterations	CPU(s)	iterations	CPU(s)	iterations	CPU(s)
1	31	2167	31	2371	31	5702	31	2149	31	2367	31	5716
2	31	1219	31	1330	31	3225	72	2431	71	2666	73	6444
4	31	854	31	922	31	2198	85	1501	82	1620	89	4434
8	31	504	31	551	31	1274	108	994	106	1047	119	2789
16	31	331	31	347	31	829	148	665	146	697	156	1825
32	31	190	31	218	31	514	171	392	170	418	171	1023
64	31	143	31	154	31	366	222	246	221	269	223	666
128	31	87	31	94	31	226	291	159	291	168	292	436

In Table II, we show the number of iterations and CPU times for an optically thicker problem, where the total cross section is increased to 100 cm^{-1}. As expected, the number of iterations increased, from 31 to 283 for the full parallel sweeps and CPU times also increased by an order of magnitude. For the parallel block-Jacobi algorithm, the number of iterations also increased. Furthermore, parallel block-Jacobi solves the problem faster than full parallel sweeps for all cases. The speed-ups for full parallel sweeps are again around 1.6 and the parallel block-Jacobi algorithm exhibits near linear speed-ups, even though the number of iterations increases slightly as the number of processors increases.

Table II. Numerical results for σ_t = 100 cm^{-1}, σ_s = 99.0 cm^{-1}, 1024 x 1024 cells

	Full Sweeps						Block-Jacobi					
	LDFEM		LDFEM-lumped		Quadratic		LDFEM		LDFEM-lumped		Quadratic	
Processors	iterations	CPU(s)	iterations	CPU(s)	iterations	CPU(s)	iterations	CPU(s)	iterations	CPU(s)	iterations	CPU(s)
1	283	17627	283	19532	283	>37000	283	17606	283	19399	283	>37000
2	283	10042	283	11832	283	28822	290	9133	290	9994	290	24975
4	283	7674	283	7742	283	18650	295	5060	295	5715	295	14001
8	283	4607	283	4392	283	11406	299	2668	299	2770	299	6750
16	283	2842	283	2969	283	7063	307	1361	307	1473	307	3574
32	283	1648	283	1838	283	4421	312	709	312	761	312	1901
64	283	1230	283	1287	283	3160	321	345	321	387	321	955
128	283	760	283	808	283	1927	334	181	334	194	334	496

6. CONCLUSIONS

We have demonstrated via a manufactured solution that LDFEM and lumped (mass and gradient) LDFEM are second order accurate while QDFEM is third order accurate for the S_N transport equation on triangles in two-dimensional, axisymmetric coordinate system. Our numerical tests show that as the optical thickness for a problem increases the parallel performance of parallel block-Jacobi becomes more effective than full parallel sweeps.

A stated earlier, we leave consideration of positivity and other numerical properties to future work. This includes effects of LDFEM and QDFEM methods on other iterative algorithms like diffusion synthetic acceleration, transport synthetic acceleration, and Krylov methods.

ACKNOWLEDGMENTS

This work was supported by Los Alamos National Laboratory, operated for the US DOE by Los Alamos National Security, LLC, under contract No. DE-AC52-06NA25396.

REFERENCES

1. M. Rosa, J. S. Warsa, and J. H. Chang, "Fourier Analysis of Parallel Inexact Block-Jacobi Splitting with Transport Synthetic Acceleration in Slab Geometry," *Proceedings of Physor-2006, ANS Topical Meeting on Reactor Physics*, Vancouver, Canada, September 10-14 (2006).
2. J. E. Morel and G. R. Montry, "Analysis and Elimination of the Discrete Ordinates Flux Dip," *Transport Theory and Statistical Physics*, **13**, pp. 615-633 (1984).
3. O. C. Zienkiewicz and R. L. Taylor, *The Finite Element Method, Volume 1: The Basis*, Butterworth-Heinemann, Oxford, England (2000).
4. J. E. Morel and J. S. Warsa, "A Lumped Bilinear-Discontinuous S_n Spatial Discretization for R-Z Quadrilateral Meshes," *Trans. Am. Nucl. Soc.*, (2006).