CONF-9506290--1

BNL-61418

# Parallel Computation With the Spectral Element Method

Hong Ma[†]

Department of Applied Science
Analytical Sciences Division, Bldg. 490D
Brookhaven National Laboratory
Upton, NY 11973, U.S.A.

## ABSTRACT

Spectral element models for the shallow water equations and the Navier-Stokes equations have been successfully implemented on a data parallel supercomputer, the Connection Machine model CM-5. The nonstaggered grid formulations for both models are described, which are shown to be especially efficient in data parallel computing environment.

## 1 Introduction

The development of the spectral element algorithms and their applications in solving incompressible flow problems [1, 2, 4, 5, 6] have demonstrated the enormous potential of the p-type of finite element method in improving computational efficiency. Like the spectral method, the spectral element method uses high-order polynomials as trial functions, but like the finite element method, it decomposes the computational domain into many elements and defines local trial functions. The hybrid character of the spectral element method enables it to overcome the shortcomings of both the spectral method and the finite element method but still retains their advantages. Since the trial functions of the spectral element method are local, it can handle complex geometry easily. On the other hand, it still is a high-order weighted residual method, so the exponential convergence rate is achieved as the order of polynomials is increased. Therefore, it is more efficient than low-order numerical methods, such as the finite difference and finite element methods.

Another important aspect which greatly enhances the computational efficiency of the spectral element method is the natural fit of this method to parallel computing. The main difference between the spectral element method and the spectral multi-domain method is that the $C^0$ and $C^1$ boundary conditions at the interface of the elements have

[†]This work received support from The U. S. Department of Energy under contract No. DE-AC02-76CH00016, and from The National Science Foundation through grant OCE-9312324. Most of the computation was performed on the CM-5 at ACL of Los Alamos National Laboratory.

to be enforced explicitly in the latter. By contrast, the spectral element method uses the variational principle to guarantee $C^0$ and $C^1$ (weakly) continuity at the interface, therefore, parallel algorithms can be implemented conveniently. Fischer and Patera [1] successfully implemented their spectral element model on MIMD (Multiple Instructions, Multiple Data) type of computers.

This paper is about the development and application of the parallel spectral element models for the Navier-Stokes equations and the shallow water equations with nonstaggered grid formulations. The performance of these models on the Connection Machine system, which is basically a SIMD (Single Instruction, Multiple Data) type of architecture, is analyzed.

## 2 Governing Equations

In the present work, we study irrotational and rotational flows, which are governed by the following two sets of equations, respectively:

Incompressible Navier-Stokes Equations:

$$\frac{\partial \mathbf{u}}{\partial \mathrm{t}} + (\mathbf{u} \cdot \nabla)\mathbf{u} = -\nabla \mathrm{p} + \frac{1}{\mathrm{R_e}} \nabla^2 \mathbf{u} \tag{2.1}$$

$$\nabla \cdot \mathbf{u} = 0 \tag{2.2}$$

where $\mathbf{u} = (\mathrm{u}, \mathrm{v}, \mathrm{w})$ is the velocity vector ($u$, $v$, and $w$ are the velocity components in the $x$, $y$, and $z$ directions, respectively); p is the pressure; $R_e = UL/\mu$ is the Reynolds number, which is based on the characteristic values of the velocity ($U$), the spatial scale ($L$), and the viscosity parameter ($\mu$) of the fluid.

Shallow Water Equations:

$$\frac{\partial \mathbf{u}}{\partial \mathrm{t}} + (\mathbf{u} \cdot \nabla)\mathbf{u} + \omega \times \mathbf{u} = -\nabla \mathrm{h} + \frac{1}{\mathrm{R_e}} \nabla^2 \mathbf{u} \tag{2.3}$$

$$\frac{\partial h}{\partial \mathrm{t}} + (\mathbf{u} \cdot \nabla)(1 + \mathrm{h}) + (1 + \mathrm{h}) \nabla \cdot \mathbf{u} = 0 \tag{2.4}$$

where $\mathbf{u} = (\mathrm{u}, \mathrm{v}, 0)$, $\omega = (0, 0, \mathrm{y})$, and h is the sea-surface elevation. The equatorial nondimensionalization scheme [3] was used in deriving the above shallow water equations.

## 3 Spectral Element Discretizations

The basis sets used in the present work are as follows:

$$\psi^e_{lm}(\xi, \eta) = h_l(\xi) h_m(\eta) \qquad l, m \in \{0, 1, ...N\}^2 \quad (2D) \tag{3.1}$$

$$\psi^e_{lmn}(\xi, \eta, \zeta) = h_l(\xi) h_m(\eta) h_n(\zeta) \qquad l, m, n \in \{0, 1, ...N\}^3 \quad (3D) \tag{3.2}$$

where $h_i(s)$ are the Gauss-Lobatto-Legendre polynomials.

If we use a single subscript, $q$ ($q \in \{1, 2, ..., (N+1)^d\}$), the mapping between a macroelement, $\Omega_e$, and its phase domain, $\hat{\Omega}_e$, can be expressed as:

$$\mathbf{x} = \mathbf{x}_\mathrm{q} \psi^e_\mathrm{q}(\xi) \tag{3.3}$$

Where $\mathbf{x} \in \Omega^e$ and $\xi \in \hat{\Omega}_e$.

Let solution $\mathbf{u}$ at time $n\Delta t$ on each subdomain $\Omega^e$ be expanded as:

$$\mathbf{u}^e(\mathrm{x}, \mathrm{n}\Delta\mathrm{t}) = \mathbf{u}_\mathrm{q}^e(\mathrm{n}\Delta\mathrm{t})\psi_\mathrm{q}^e[\xi(\mathrm{x})] \tag{3.4}$$

where $f_q^e(t)$ is the value of function $f$ at the collocation point $\mathbf{x}_\mathrm{q} \in \Omega^e$ at time t.

We use a semi-implicit temporal discretization scheme for the Navier-Stokes equations, i.e., the advection terms are treated with the third-order Adams-Bathforth scheme, and the mixing terms with the Crank-Nicolson scheme. Then, the resulting equations are:

$$\frac{\mathbf{u}^* - \mathbf{u}^\mathrm{n}}{\Delta\mathrm{t}} = \sum_{\mathrm{i}=0}^{2} \alpha_i(\mathbf{u}^{\mathrm{n-i}} \cdot \nabla)\mathbf{u}^{\mathrm{n-i}} \tag{3.5}$$

$$\frac{\mathbf{u}^{\mathrm{n+1}} - \mathbf{u}^*}{\Delta\mathrm{t}} = -\nabla p^{n+\frac{1}{2}} + \frac{1}{R_e}\nabla^2\mathbf{u}^{\mathrm{n+1}} \tag{3.6}$$

$$\nabla^2 p^{n+\frac{1}{2}} = \frac{1}{\Delta t}\nabla \cdot \mathbf{u}^* \tag{3.7}$$

where $\alpha_0 = -\frac{23}{12}$, $\alpha_1 = \frac{4}{3}$, and $\alpha_2 = -\frac{5}{12}$.

Applying the above spectral element discretization scheme to the weak forms of the Navier-Stokes equations, we derive the following matrix formulae:

$$[B][\mathbf{u}^*] = [\mathrm{B}][\mathbf{u}^\mathrm{n}] + \Delta\mathrm{t}\sum_{\mathrm{i}=0}^{2}\alpha_\mathrm{i}\sum_{\mathrm{j}=1}^{\mathrm{d}}[\mathrm{C}_\mathrm{j}^{\mathrm{n-i}}][\mathrm{u}_\mathrm{j}^{\mathrm{n-i}}] \tag{3.8}$$

$$[B][\mathbf{u}^{\mathrm{n+1}}] = [\mathrm{B}][\mathbf{u}^*] - \Delta\mathrm{t}\left\{\mathbf{D}^\mathrm{T}[\mathrm{p}^{\mathrm{n+\frac{1}{2}}}] + \frac{1}{\mathrm{R_e}}[\mathrm{A}][\mathbf{u}^{\mathrm{n+1}}]\right\} \tag{3.9}$$

$$[A][p^{n+\frac{1}{2}}] = -\frac{1}{\Delta t}[\mathbf{D}] \cdot [\mathbf{u}^*] \tag{3.10}$$

where $[A]$ and $[B]$ are the stiffness matrix and mass matrix, respectively; $[\mathrm{C}_\mathrm{j}^\mathrm{n}]$ and $[\mathrm{D}]$ are generalized vectors whose $m^{th}$ components, $[C_{j,m}^n]$ and $[D_m]$, are the advection matrix and gradient matrix, respectively.

The boundary conditions used in the above derivations are:

$$\mathbf{u}(\mathrm{x}, \mathrm{t}) = \mathbf{f}(\mathrm{x}, \mathrm{t}) \qquad \mathbf{x} \in \partial\Omega \tag{3.11}$$

$$p(\mathbf{x}, \mathrm{t}) = \mathrm{g_1}(\mathbf{x}, \mathrm{t}) \quad \text{or} \quad \nabla\mathrm{p}(\mathbf{x}, \mathrm{t}) \cdot \mathbf{n} = \mathbf{g_2}(\mathbf{x}, \mathrm{t}) \qquad \mathbf{x} \in \partial\Omega \tag{3.12}$$

where $\partial\Omega$ is the boundary of the computational domain, $\Omega$.

We chose the isoparametric spectral element discretization scheme, namely, using nonstaggered grids, for the present numerical model. The nonstaggered formulation avoids spurious pressure modes as staggered schemes do, and, at the same time, has the advantage that pressure is continuous across boundaries of the spectral elements. Only one set grids is required for both interpolation and quadrature, hence simplifying operations. We find that the nonstaggered grid formulations do not necessarily entail high costs for the iterative pressure solver as one might expect, and the two types of pressure boundary conditions in (3.12) make little difference in terms of computational efficiency.

The discretized shallow water equations have the following form:

$$[B][\mathbf{u}^{n+1}] = [\mathrm{B}][\mathbf{u}^{n}]$$
$$+\Delta t \sum_{i=0}^{2} \alpha_i \left\{ \sum_{j=1}^{2} [\mathrm{C}_j^{n-i}][\mathrm{u}_j^{n-i}] - [\mathbf{D}^{\mathrm{T}}][\mathrm{p}^{n-i}] - \frac{1}{\mathrm{R_e}}[\mathrm{A}][\mathbf{u}^{n-i}] \right\} \tag{3.13}$$
$$[B][h^{n+1}] = [B][h^{n}]$$
$$+\Delta t \sum_{i=0}^{2} \alpha_q \left\{ [D_1][u^{n-i}(1+h^{n-i})] + [D_2][v^{n-i}(1+h^{n-i})] \right\} \tag{3.14}$$

Details of deriving Eqs.(3.13) and (3.14) were given in [3].

# 4 Parallel Algorithm and Implementation

For a distributed-memory, massively parallel computer, which is the chosen architecture for the present models, remote memory access usually is much slower than local memory access. To efficiently implement the spectral element model on this type of computer, we must create a data parallelism which minimizes communication among processing nodes. This goal is achieved through a data-mapping scheme that allows for all the information related to a given spectral element to be collected in the memory of a single processor. On the Connection Machine model CM-5, we pursue data parallelism by designing the layout of the arrays of the spectral element model in such a way that the axes along the number of elements are assigned as parallel dimensions, and those along intra-element degrees of freedom as serial ones. The following is a three-dimensional example in CM Fortran syntax

$$REAL \quad A(N,N,N,K_1,K_2,K_3)$$
$$CMF\$LAYOUT \quad A(:SERIAL,:SERIAL,:SERIAL,:NEWS,:NEWS,:NEWS)$$

where $K_1$, $K_2$, $K_3$ are numbers of elements in each spatial dimension, separately, and $N$ is the order of polynomials.

Basically, there are four types of algebraic computations involved in solving the discretized Navier-Stokes Equations (3.8)-(3.10), when the preconditioned conjugated gradient iteration method is applied:

$$[r] = [A][u] \tag{4.1}$$
$$[s] = [B][u] \quad or \quad [B]^{-1}[u] \tag{4.2}$$
$$[t] = [C^n_{m,j}][u] \tag{4.3}$$
$$\alpha = [u] \cdot [v] \tag{4.4}$$

Operations in (4.1)−(4.3) are all matrix-vector products, except that the matrices in (4.2) and (4.3) are diagonal ones. The elements of $[A]$ and $[B]$ are constants which only need to be calculated once; those of $[C^n_{m,j}]$ are not, and they need to be evaluated at each time step. (4.4) is the inner product of two vectors.

Matrices in (4.1)−(4.3) are global ones, which means that they are the results of the direct stiffness summation. The way in which the direct stiffness summation is

performed can significantly affect the efficiency of the parallel implementation of the model. In serial spectral element algorithms, direct stiffness summation usually is carried out automatically by using local and global node numbering systems. However, in data parallel programs we have to treat direct stiffness summation separately to avoid explicit short messages.

We split the procedure of calculating matrix-vector products into two steps, each of which admits concurrency. At the first step, the matrix-vector products are carried out at the elemental level with, for example, the elemental Laplacian and mass matrices:

$$r^k(i) = \sum_{q=1}^{(N+1)^d} A^k(i,q)u^k(q) \qquad i \in \{1,2,...,(N+1)^d\}, \quad k \in \{1,2,...,K\} \tag{4.5}$$

$$s^k(i) = B^k(i)u^k(i) \qquad i \in \{1,2,...,(N+1)^d\}, \quad k \in \{1,2,...,K\} \tag{4.6}$$

The array layout described at the beginning of this section defines a one-to-one correspondence between the spectral elements and the virtual processors. All data related to a given element are stored in the memory of a single processor. Therefore, there is no communication among neighboring processors during these elemental level computations, and they are performed concurrently across all virtual processors.

After applying tensor-product factorization, the computational complexities to evaluate (4.5) and (4.6) would be $C_1KN^{d+1}/Q$ and $C_2KN^d/Q$, respectively, where $Q$ is the number of physical processors involved. On the Connection Machine systems, parallel data structure allows (4.6) to be performed in an array operation, which means that thousands of simultaneous multiplications are made across all the array elements. Hence, $C_2$ is a small number. Consequently, diagonal preconditioning (4.2) is especially efficient in the data parallel environment: it does not require direct stiffness summation, and only local computation is involved, which is very fast. Iteration counts can be reduced by twenty to thirty percent with about a one percent increase in cost.

The processing nodes on the latest CM-5 model are equipped with powerful vector-processing units that can further reduce the cost of elemental level computation. These vector-processing units are most efficient when the order of the spectral elements is high.

The second step is to carry out direct stiffness summation, $\sum_{k=1}^{K}{}'$, in which contributions from local nodes that share the same physical coordinates are first accumulated, and then assigned back to those local nodes. In a serial spectral element model, this procedure can be accomplished by using global and local index systems, and is automatically done as the matrix computation is made for each spectral element. In the parallel spectral element model, however, it is more efficient to use a separate step for the direct stiffness summation. Since each spectral element has at least one edge (two-dimensional case) or one surface (three-dimensional case) that is shared by a neighboring element, the direct stiffness summation can be carried out simultaneously along these edges or surfaces enabling structured message exchange, i.e., edge-based message exchange for two-dimensional problems, and surface-based message exchange for three-dimensional ones. Since this kind of information exchange takes place along the linkages of the "macro-element-skeleton", it can be easily synchronized for all elements in the entire domain. The work per processor that is required in this procedure is $C_3dKN^{d-1}/Q$. The structured message exchange mostly avoids explicit short messages, and it considerably improves the parallel efficiency of the spectral element model [1].

With parallel prefix of the CM Fortran, MATMUL and SUM, the inner product (4.4) is executed completely in parallel. Its computational complexity is $C_4 K N^d/Q$. Due to the high level of concurrency afforded by the parallel prefixes, $C_4$ is a small number.

Eq.(3.8) only requires a direct method to solve. The computation kernel here is the evaluation of the advection term where concurrency can be achieved at different levels of the computation. We first evaluate the shears of velocities at all nodal points

$$\frac{\partial u_m^{n,e}(l)}{\partial x_j} = \sum_{s=1}^{(N+1)^d} u_m^{n,e}(s) \frac{\partial \psi_s^e}{\partial x_j}$$
$$l \in \{1, 2, ..., (N+1)^d\}, \quad m, j \in \{1, 2, ..., d\}^2, \quad e \in \{1, 2, ..., K\} \quad (4.7)$$

This operation is executed concurrently across all virtual processors. With the partial summation method, the computational complexity for (4.7) is $C_5 K(N+1)^{d+1}/Q$. The advection term in (3.8) also can be written as

$$C_{m,j}^{n,e}(p,q) = \sum_{s=1}^{(N+1)^d} u_m^{n,e}(s) \frac{\partial \psi_s^e(q)}{\partial x_j} \int_{\hat{\Omega}^e} \psi_p^e \psi_q^e |J^e| d\hat{\Omega}^e$$
$$p, q \in \{1, 2, ..., (N+1)^d\}^2, \quad j, m \in \{1, ..., d\}^2 \quad (4.8)$$

Therefore, once the shears of velocities are obtained, the remaining operation to evaluate the advection terms is the same as that of (4.6). Hence, the total computational complexity of (4.8) is $C_2 K(N+1)^d/Q + C_5 K(N+1)^{d+1}/Q$.

We can undertake similar parallelization analysis for solving the discretized shallow water equations (3.13) and (3.14). In this case, the computational complexity for matrix-vector product and that for the convection term are the same as those derived for the Navier-Stokes equations, except $d = 2$. Since no iterative procedures are involved, the model for the shallow water equations is relatively inexpensive compared to a "true" three-dimensional model.

As spectral elements are of high-order, most of the costly operations are at the elemental level, and they are executed concurrently. The spectral granularity at the elemental level can take full advantage of the computing power that the latest processing units provide. The structured message exchange, combined with parallel prefix, makes inter-element communication a lower-order rather than a highe-order cost, compared to that of elemental level computation. This communication cost should be much smaller than that of the h-type finite element model, partially because many fewer redundant nodel values, shared by more than one element, have to be stored.

Tables 1 and 2 show the CM-5 timing results per pressure iteration in solving the two-dimensional and three-dimensional Navier-Stokes equations, respectively. We find that for a fixed order of the basis functions ($N$), expanding the size of the problem by increasing $K$ (the number of the spectral elements) and, at the same time, increasing the number of processing nodes by the same proportion, hardly changes the computational cost in terms of CPU time per pressure iteration. Due to the high parallel efficiency, the spectral element models can fully take advantage of the highly scalable performance of the massively parallel architectures.

Table 1: CM-5 Timing Results per Pressure Iteration (2-D)

| N | K | Num. Grid Pts. | Num. Nodes* | CPU (s) |
|---|---|---|---|---|
| 7 | 1024 | 50625 | 32 | 0.062 |
| 7 | 4096 | 201601 | 128 | 0.063 |
| 15 | 1024 | 231361 | 32 | 0.317 |
| 15 | 4096 | 923521 | 128 | 0.318 |

* Each CM-5 node has 4 vector-processing units and a peak performance rating of 128 MFLOPS.

Table 2: CM-5 Timing Results per Pressure Iteration (3-D)

| N | K | Num. Grid Pts. | Num. Nodes | CPU (s) |
|---|---|---|---|---|
| 4 | 4096 | 274625 | 64 | 0.106 |
| 4 | 32768 | 2146689 | 512 | 0.108 |
| 7 | 4096 | 1442897 | 64 | 0.473 |
| 7 | 32768 | 11390625 | 512 | 0.475 |

Table 3 shows the parallel performance for the spectral element model for the shallow water equations. To compare the performance of the spectral element model on the CM-5 with that on a sequential computer, say, RISC/6000, we made Table 4 for "equivalent performance", $\eta_{EP}$, which is defined as $\eta_{EP} = 28 MFLOPS \times K \times \frac{T_{RISC}}{T_{CM5}}$. $K$ is the number of spectral elements, $T_{RISC}$ is the double precision timing on the RISC/6000 (model 560) which has a LINPACK performance rating of 28 MFLOPS ($T_{RISC}$=0.012 s per spectral element per time step), and $T_{CM5}$ is the double-precision timing on the CM-5.

Table 3: CM-5 Timing Results (in seconds) per Time Step for the Spectral Element Shallow Water Equation Model (N=16)

| K | 128 nodes | 256 nodes | 512 nodes |
|---|---|---|---|
| 100 | 0.31 | 0.31 | 0.31 |
| 256 | 0.32 | 0.31 | 0.31 |
| 4900 | 0.66 | 0.48 | 0.32 |
| 10000 | 0.83 | 0.48 | 0.32 |
| 22500 | 1.71 | 0.97 | 0.66 |
| 40000 | 2.33 | 1.29 | 0.80 |
| 90000 | - | - | 1.6 |

Table 4: Equivalent Performance for CM-5 (in MFLOPS) for the Spectral Element Shallow-Water-Equation Model (N=16)

| K | 128 nodes | 256 nodes | 512 nodes |
|---|---|---|---|
| 100 | 115 | 115 | 115 |
| 256 | 2789 | 2789 | 2879 |
| 4900 | 2650 | 3644 | 5466 |
| 10000 | 4301 | 7437 | 11156 |
| 22500 | 4725 | 8280 | 12170 |
| 40000 | 6208 | 10984 | 17850 |
| 90000 | - | - | 20081 |

One advantage of the high-order domain decomposition numerical model is that it can take better advantage of the vector-processing units than can low-order models. Table

5 shows that the difference between the performance of the spectral element model with vectorization and that without vectorization increases as the order of the basis functions, $N$, increases.

Table 5: CM-5 Timing Results per Time Step for the 2-D N-S Model (32 nodes) With an Iteration Error of $O(10^{-6})$

| | | CPU (s) | |
|---|---|---|---|
| N | K | With Vectorization | Without Vectorization |
| 3 | 576 | 2.2 | 3.2 |
| 4 | 100 | 2.1 | 4.8 |
| 5 | 16 | 1.9 | 7.9 |
| 6 | 4 | 2.6 | 11.4 |

# 5 CONCLUSIONS

In present work, we have shown that high-order domain decomposition methods can be efficiently applied in a data parallel programming environment. The optimized computational efficiency of the parallel spectral element model comes not only from the exponential convergence of its numerical solutions, but also from its efficient usage of the powerful vector-processing units of the latest parallel architectures which have a highly scalable performance. The nonstaggered grid formulation is convenient for, and shows no disadvantage in, these parallel spectral element models.

## REFERENCES


1. Fischer, P. F. and A. T. Patera, J. Comput. Phys., 92 (1991) 380.
2. Korczak, K. Z. and A. T. Patera, J. Comput. Phys., 62 (1986) 361.
3. Ma, H., J. Comput. Phys., 109 (1993) 133.
4. Maday, Y. and A. T. Patera, in *State-Of-The-Art Surveys on Computational Mechanics,* A.D. Noor and J. T. Oden (eds.), New York, New York, 1989.
5. Patera, A. T., J. Comput. Phys., 54 (1984) 468.
6. Ronquist, E. M., Optimal Spectral Element Methods for the Unsteady Three Dimensional Navier-Stokes Equations, Ph.D. Thesis, The Massachusetts Institute of Technology (1988).