We are IntechOpen, the world's leading publisher of Open Access books Built by scientists, for scientists

4,800
Open access books available

122,000
International authors and editors

135M
Downloads

Our authors are among the

154
Countries delivered to

TOP 1%
most cited scientists

12.2%
Contributors from top 500 universities

WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

Numbers displayed above are based on latest data collected.
For more information visit www.intechopen.com

8

The Elliptic Solvers in the Canadian Limited Area Forecasting Model GEM-LAM

Abdessamad Qaddouri and Vivian Lee
Atmospheric Science and Technology Directorate, Environment Canada
Canada

1. Introduction

The Global and Limited area versions of the Canadian global Environmental (GEM) model (Coté et al.,1998), currently used for operational numerical weather prediction (NWP) at Canadian Meteorological centre (CMC) employ an implicit time discretization on the spatial grids of tensors product. This gives rise to a separable 3D elliptic boundary value (EBV) problem that must be solved at each model time step. This EBV problem is also found in the implicit formulation of the high order diffusion equation which is used in GEM. Most models in operational NWP apply this selective diffusion in order to eliminate high wave-number noise due (for example) to numerical discretization. The solution of the EBV is in general, at the heart of most models used for NWP. It is currently solved in the limited area version (GEM-LAM) by applying either a direct or an iterative method.
As in the global GEM model (Qaddouri et al., 1999), the direct method in GEM-LAM model is implemented with either fast or slow Fourier transforms. Because of the nature of the boundary conditions in GEM-LAM, discrete cosine transforms (DCTs) are use instead of discrete Fourier transforms (DFTs). In the case of the slow transform, which involves a full matrix multiplication and this will be referred as MXMA, the cost per grid point increases linearly with the number of grid points along the transform direction and if this number divides properly, the fast Fourier transforms (FFT) can be used. It is important to find ways to optimize this slow transform as it dominates the cost of the solver and, it does represent a significant fraction of the total cost of the model time step. The iterative method is implemented in GEM-LAM to reduce the cost of the slow transform in the elliptic solver and it is based on a preconditioned Generalized Minimal REsidual (GMRES) algorithm (Saad, 1996). The GEM-LAM model is parallelized with a hybrid use of MPI and OpenMP. In the parallel version of the direct solver, four global communications are used which could be very time-consuming when the number of processors increases. In the iterative solver, each processor communicates with its neighbours and global communications are limited only to calculating a global dot product.
In the following sections, EBV problem to be solved is presented and the direct and iterative methods are described. The high order diffusion is discussed in section 3 while the parallelization of direct solution is shown in section 4. Results from numerical experiments are reported and analyzed in section 5, and the paper is concluded in section 6.

2. Positive Definite Helmholtz

The problem to be solved is a 3D separable EBV problem which after a vertical separation, it is reduced to a set of horizontal positive definite Helmholtz problems (Coté et al., 1998).
To illustrate the derivation of a Helmholtz problem from an implicit time discretization of the terms responsible for the gravity wave, and to show the nature of the horizontal boundary conditions for the GEM-LAM model , the following simple 2D linear Shallow water equations is integrated in a limited area of a non-rotating sphere with radius a :

$$\begin{aligned} &\frac{\partial U}{\partial t}+\frac{1}{a^2}\frac{\partial \phi}{\partial \lambda}=0, \\ &\frac{\partial V}{\partial t}+\frac{1}{a^2}cos(\theta)\frac{\partial \phi}{\partial \theta}=0, \\ &\frac{\partial \phi}{\partial t}+\phi^*\frac{1}{cos^2(\theta)}\left(\frac{\partial U}{\partial \lambda}+cos(\theta)\frac{\partial V}{\partial \theta}\right)=0, \end{aligned} \tag{1}$$

where the unknowns are the wind images *U, V* (wind times $cos(\theta)/a$), and ϕ is the perturbation geopotential from the reference geopotential ϕ^* . Using the staggered Arakawa C-grid (Arakawa et al., 1977), the wind components are placed in the middle of the lines joining the geopotential points. By employing the Crank-Nicholson time discretization , the resulting equations at the forecast time *t* are:

$$\frac{U}{\tau}+\frac{1}{a^2}\frac{\partial \phi}{\partial \lambda}=RU=\left[\frac{U}{\tau}+\frac{1}{a^2}\frac{\partial \phi}{\partial \lambda}\right]^-, \tag{2}$$

$$\frac{V}{\tau}+\frac{1}{a^2}cos(\theta)\frac{\partial \phi}{\partial \theta}=RV=\left[\frac{V}{\tau}+\frac{1}{a^2}cos(\theta)\frac{\partial \phi}{\partial \theta}\right]^-, \tag{3}$$

$$\frac{\phi}{\tau}+\phi^*\frac{1}{cos^2(\theta)}\left(\frac{\partial U}{\partial \lambda}+cos(\theta)\frac{\partial V}{\partial \theta}\right)=R\phi=\left[\frac{\phi}{\tau}+\phi^*\frac{1}{cos^2(\theta)}\left(\frac{\partial U}{\partial \lambda}+cos(\theta)\frac{\partial V}{\partial \theta}\right)\right]^-. \tag{4}$$

Where $\tau=\frac{\Delta t}{2}$ and the minus (-) superscript indicates the evaluation at the previous time step. To discretize in the space let, $\left(\lambda_i, i=1,NI, \theta_j, j=1,NJ\right)$ denote the set of ϕ -grid points, $\left(\tilde{\lambda}_i, i=0,NI, \theta_j, j=1,NJ\right)$ denote the set of *U*-grid points and $\left(\lambda_i, i=1,NI, \tilde{\theta}_j, j=0,NJ\right)$ denote the set of *V*-grid points with

$$\begin{aligned} &\tilde{\lambda}_i=\frac{\left(\lambda_i+\lambda_{i+1}\right)}{2}, \\ &\tilde{\theta}_i=\frac{\left(\theta_j+\theta_{j+1}\right)}{2}. \end{aligned} \tag{5}$$

Note that the boundary conditions on *U* and *V* are applied at $\tilde{\lambda}_0, \tilde{\lambda}_N$ and $\tilde{\theta}_0 \tilde{\theta}_M$. Then apply the wall boundary conditions

$$\begin{aligned} U\left(\tilde{\lambda}_0,\theta,t\right)&=U_0(\theta)\\ U\left(\tilde{\lambda}_N,\theta,t\right)&=U_N(\theta)\\ V\left(\tilde{\lambda},\tilde{\theta}_0,t\right)&=V_0(\lambda)\\ V\left(\lambda,\tilde{\theta}_M,t\right)&=V_M(\lambda), \end{aligned} \tag{6}$$

which are provided by a global model (or a larger LAM model) for U_0, U_N, V_0 and V_M to the target limited area model. The operator Δ is defined as $\Delta\lambda_i=\lambda_{i+1}-\lambda_i$. Integrating the equations (2-4) over their specific control volume $\left(\Delta\tilde{\lambda}_{i-1}\times\Delta\theta_j\right)$, $\left(\Delta\lambda_i\times\Delta\tilde{\theta}_j\right)$ and $\left(\Delta\lambda_i\times\Delta\theta_j\right)$ will yield

$$\Delta\tilde{\lambda}_{i-1}\Delta\theta_j\frac{U_{i,j}}{\tau}+\frac{1}{a^2}\left(\phi_{i+1,j}-\phi_{i,j}\right)=\Delta\tilde{\lambda}_{i-1}\Delta\theta_j\,RU_{i,j}\,, \tag{7}$$

$$\Delta\lambda_i\times\Delta\tilde{\theta}_{j-1}\frac{V_{i,j}}{\tau}+\frac{1}{a^2}\left(cos\left(\tilde{\theta}_j\right)\phi_{i,j+1}-cos\left(\tilde{\theta}_{j-1}\right)\phi_{i,j}\right)=\Delta\lambda_i\times\Delta\tilde{\theta}_{j-1}\,RV_{i,j}\,, \tag{8}$$

$$\Delta\lambda_i\times\Delta\theta_j\frac{\phi_{i,j}}{\tau}+\phi^*\frac{1}{cos^2\left(\theta_i\right)}\left(\Delta\theta_j\left(U_{i+1,j}-U_{i,j}\right)+cos\left(\theta_j\right)\phi_{i,j+1}-cos\left(\theta_j\right)\left(V_{i+1,j}-V_{i,j}\right)\right)=\Delta\lambda_i\times\Delta\theta_j\,R\,\phi_{i,j}\,, \tag{9}$$

with

$$\begin{aligned} RU_{0,j}&=U_{0,}(j)/\tau\\ RU_{N,j}&=U_{N,}(j)/\tau\\ RV_{i,0}&=V_{0,}(i)/\tau\\ RV_{i,N}&=V_{N,}(i)/\tau. \end{aligned} \tag{10}$$

Two equations from (7) are re-written to express at two neighbouring points *(i+1,j)* and *(i,j)* in U-grid while two equations from (8) are re-written to express at two neighbouring points *(i,j+1)* and *(i,j)* in *V*-grid. These four resulting equations are added and combined with equation (9) to obtain the following discretized elliptic equation (11) for the geopotential only. This equation is called the (positive definite) Helmholtz equation by the NWP community.

$$\mathbf{A}\phi=\mathbf{r}, \tag{11}$$

where

$$\mathbf{A}=\left[\eta\,P(\theta)\otimes P(\lambda)-P'(\theta)\otimes P_{\lambda\lambda}-P_{\theta\theta}\otimes P(\lambda)\right], \tag{12}$$

where $\eta=\dfrac{a^2}{\phi^*\tau^2}$, $\otimes$ is the notation for tensor product and

$$\mathbf{r}=P(\theta)\otimes P(\lambda)R, \tag{13}$$

with

$$P(\theta)=\begin{bmatrix}\Delta\sin\tilde{\theta}_0 & & & \\ & \Delta\sin\tilde{\theta}_1 & & \\ & & \ddots & \\ & & & \Delta\sin\tilde{\theta}_{NJ\text{-}1}\end{bmatrix}, \tag{14}$$

$$P'(\theta)=\begin{bmatrix} \dfrac{\Delta\sin\tilde{\theta}_0}{\cos^2\theta_1} & & & \\ & \dfrac{\Delta\sin\tilde{\theta}_1}{\cos^2\theta_2} & & \\ & & \ddots & \\ & & & \dfrac{\Delta\sin\tilde{\theta}_{NJ-1}}{\cos^2\theta_{NJ}} \end{bmatrix}, \tag{15}$$

$$P_{\theta\theta}=\begin{bmatrix} -\dfrac{\cos^2\tilde{\theta}_1}{\Delta\sin\theta_1} & \dfrac{\cos^2\tilde{\theta}_1}{\Delta\sin\theta_1} & & & \\ \dfrac{\cos^2\tilde{\theta}_1}{\Delta\sin\theta_1} & -\dfrac{\cos^2\tilde{\theta}_1}{\Delta\sin\theta_1}-\dfrac{\cos^2\tilde{\theta}_2}{\Delta\sin\theta_2} & \dfrac{\cos^2\tilde{\theta}_2}{\Delta\sin\theta_2} & & \\ & \ddots & \ddots & \ddots & \\ & & -\dfrac{\cos^2\tilde{\theta}_{NJ-2}}{\Delta\sin\theta_{NJ-2}}-\dfrac{\cos^2\tilde{\theta}_{NJ-1}}{\Delta\sin\theta_{NJ-1}} & \dfrac{\cos^2\tilde{\theta}_{NJ-1}}{\Delta\sin\theta_{NJ-1}} \\ & & \dfrac{\cos^2\tilde{\theta}_{NJ-1}}{\Delta\sin\theta_{NJ-1}} & -\dfrac{\cos^2\tilde{\theta}_{NJ-1}}{\Delta\sin\theta_{NJ-1}} \end{bmatrix}, \tag{16}$$

$$P(\lambda)=\begin{bmatrix} \Delta\tilde{\lambda}_0 & & & \\ & \Delta\tilde{\lambda}_1 & & \\ & & \ddots & \\ & & & \Delta\tilde{\lambda}_{NI-1} \end{bmatrix}, \tag{17}$$

$$P_{\lambda\lambda}=\begin{bmatrix} -\dfrac{1}{\Delta\lambda_1} & \dfrac{1}{\Delta\lambda_1} & & & \\ \dfrac{1}{\Delta\lambda_1} & -\dfrac{1}{\Delta\lambda_1}-\dfrac{1}{\Delta\lambda_2} & \dfrac{1}{\Delta\lambda_2} & & \\ & \dfrac{1}{\Delta\lambda_2} & \ddots & \ddots & \\ & & \ddots & \ddots & \dfrac{1}{\Delta\lambda_{NI-1}} \\ & & & \dfrac{1}{\Delta\lambda_{NI-1}} & -\dfrac{1}{\Delta\lambda_{NI-1}} \end{bmatrix}. \tag{18}$$

In GEM-LAM with NK vertical levels, we have NK different constants $\eta(k)$, k=1,NK which are eigenvalues of a generalized eigenvalue problem like in Eq. 21 but along the vertical direction. After expanding the variables in z-direction eigenvectors, a vertical separation is done in the direct solver, and NK Helmholtz problems like the one in Eq.11 are obtained. Since the 3D problem is vertically separable, only the horizontal aspect needs to be considered though the numerical results obtained are for the full 3D elliptic solver.

Because of the nature of the horizontal boundary conditions in GEM-LAM, a brief discussion on the discrete cosine Fourier transforms in the context of differential equations is presented in the following sub-section.

2.1 Discrete Cosine Fourier transforms

Like discrete Fourier transforms (DFTs), discrete cosine transforms (DCTs) express a function in terms of a sum of sinusoids with different frequencies and amplitudes. DCT uses only cosine functions, while DFT uses both cosine and sine in the form of complex exponentials. The DFT implies a periodic extension of the function true along the longitude axis in the GEM global atmospheric model. The expression of the matrix $P_{\lambda\lambda}$ gives a clear idea about the boundary conditions implicitly used for the Helmholtz problem in the longitudinal directions in GEM-LAM. It can be seen that an even extension of the functions, possible by the use of DCT, is needed (Denis et al., 2004). Consider the analysis of right hand side of the equation (11) for each latitude *j*, the discrete cosine transform is

$$\tilde{\mathbf{r}}_{k,j} = \sum_{i=0}^{NI-1} \mathbf{r}_{i,j} \cos\left[\frac{\pi}{NI}\left(i+\frac{1}{2}\right)k\right]. \tag{19}$$

This DCT used in GEM-LAM implies this following boundary condition: $\mathbf{r}_{i,j}$ is even around the grid point $i=-1/2$, even around $i=NI-1/2$ and $\tilde{\mathbf{r}}_{k,j}$ is even around the wave-number $k=0$ and around $k=NI$. The DCT matrix is made orthogonal by scaling with a factor $\sqrt{2/NI}$ and if *NI* divides properly, the Fast Fourier transform (FFT) is used.

2.2 Direct Solution of Helmholtz problem

The direct solution of (11) is obtained by exploiting its separability, and expanding ϕ in λ-direction eigenvectors that diagonalize $\mathbf{A}$, i.e.

$$\phi_{ij} = \sum_{i=1}^{NI} \psi_i^{[I]} \phi_j^{[I]}, \tag{20}$$

with

$$P_{\lambda\lambda}\,\psi^{[I]} = \varepsilon_I\, P(\lambda)\psi^{[I]}, I = 1, NI, \tag{21}$$

and the property of orthogonality in

$$\sum_{i=1}^{NI} \psi_i^{[I']}\, P(\lambda)_{ii}\, \psi_i^{[I]} = \delta_{\mathrm{II'}}, \tag{22}$$

which is used to project (11) on each mode in turn. The result can then be written in matrix form as

$$\mathbf{A}^{[I]}\phi^{[I]} + \eta\, I^{[I]}(\theta)\phi^{[I]} = \mathbf{r}^{[I]}, \tag{23}$$

where

$$\mathbf{A}^{[I]} = \left[-\varepsilon_I\, P'(\theta) - P_{\theta\theta}\right]; \quad I^{[I]}(\theta) = P(\theta). \tag{24}$$

The algorithm can then be summarized as:

1. Analysis of the right-hand side: $\mathbf{r}_j^{[l]} = \sum_{i=1}^{NI} \psi_i^{[l]} \mathbf{r}_{ij}$,

2. solution of the NI tridiagonal problems in equation (23),

3. synthesis of the solution, $\phi_{ij} = \sum_{i=1}^{NI} \psi_i^{[l]} \phi_j^{[l]}$.

It is well known that for a uniform λ-grid, the modes ψ^l are proportional to the usual Fourier modes and the analysis and synthesis steps can be implemented with a real Fourier transform, and if furthermore *NI* factorizes properly, the Fast Fourier Transform (FFT) algorithm can be used, otherwise these steps are implemented with a full matrix multiplication (MXMA). In GEM-LAM, ψ^l are the cosine functions as in equation (19).

2.3 Iterative solution of Helmholtz problem

The equation (11) can also be solved by a robust parallel preconditioned GMRES algorithm where block-Jacobi iteration is used as a preconditioner (single level additive Schwarz). The blocks correspond to a different direct solution of the local Helmholtz problem in sub-domains. The GMRES code used here is the same parallel code developed by Saad and Malevsky (Saad & Malevsky, 1995), where a reverse communication mechanism is implemented in order to avoid passing a large list of arguments to the routines and, the data structure is left suitable for all possible types of preconditioners. The code can be read as follows:

```
Icode =0
1 continue
Call gmres(_)
If (icode.eq.1) then
 Call preconditioner()
Goto 1
Else if (icode.eq.2)
Call Matrix-vector()
Goto 1
Endif.
```

For each subdomain (processor), the preconditioner is called locally as a direct solution like the paragraph above but it only uses the slow transform where the matrix multiplication (MXMA) uses local data. It is known here that the preconditioning operation involves no inter-processors communication. In the matrix-vector operation, each processor will need to communicate with its neighbours.

3. The 2D high order implicit diffusion

The problem to solve is a high order diffusion equation which takes the form of the time dependent equation

$$\frac{\partial \phi}{\partial \lambda} = \mu * (-1)^{\frac{m}{2}+1} * \nabla^m \phi, \tag{25}$$

where ϕ is any prognostic variable, the integer m can be any multiple of 2 and denotes the order. The diffusion coefficient μ is considered constant for simplicity. The diffusion is periodically applied to specific meteorological fields and its effect is equivalent to applying diffusion in the prognostic equation for a period of time. In the operational GEM-LAM, only the direct solution (sec.3.2) is implemented for the 2D high order diffusion which considers either the slow (MXMA) or fast (FFT) Fourier transforms. The work for the iterative solution has not been completed.

3.1 Temporel Discretization

Implicit discretization of the equation (25) is

$$\frac{\phi^{n+1}-\phi^{n}}{\Delta t}=\mu*(-1)^{\frac{m}{2}+1}*\nabla^{m}\phi^{n+1}\,. \tag{26}$$

Integration of the equation (26) gives the following equation

$$\nabla^{m}\phi^{n+1}+\frac{1}{\mu*(-1)^{\frac{m}{2}+1}\Delta t}\phi^{n+1}=\frac{1}{\mu*(-1)^{\frac{m}{2}+1}\Delta t}\phi^{n}\,, \tag{27}$$

i.e.,

$$\nabla^{m}\phi+\eta\phi=R\,, \tag{28}$$

where R is the forcing term, η is a prescribed parameter. Numerical solution of (28) on the sphere requires a conservative discretization of ∇^{m}, with a consistent treatment of the boundary condition. In this paper, with m = 4 for example, the equation (28) would become the following two second order equations to be solved:

$$\nabla^{2}Z+\eta\phi=f;\ \nabla^{2}\phi-Z=0\,, \tag{29}$$

where ∇^{2} operator on the sphere is

$$\nabla^{2}=\frac{1}{a^{2}\cos^{2}\theta}\frac{\partial^{2}}{\partial\lambda^{2}}+\frac{1}{a^{2}\cos\theta}\frac{\partial}{\partial\theta}\left(\cos\theta\frac{\partial}{\partial\theta}\right), \tag{30}$$

and λ is the longitude, θ is the latitude and a is the earth's radius.

3.2 Spatial discretization

The spatial discretization is on the Arakawa C-grid as described in section 2. This is a tensorial grid. The discrete form of (29) is

$$\mathbf{A}Z+\eta I(\lambda,\theta)\phi=\mathbf{r};\ \ \mathbf{A}\phi-I(\lambda,\theta)Z=0, \tag{31}$$

where

$$\mathbf{A}=\left[-P'(\theta)\otimes P_{\lambda\lambda}-P_{\theta\theta}\otimes P(\lambda)\right], \tag{32}$$

where $\otimes$ is the notation for tensor product and

$$\mathbf{r} = P(\theta) \otimes P(\lambda) R; \quad I(\lambda, \theta) = P(\theta) \otimes P(\lambda), \tag{33}$$

3.3 Direct Solution

The direct solution of (15) is obtained by exploiting the separability as in the Helmholtz problem, and expanding ϕ and Z in λ -direction eigenvectors that diagonalize $\mathbf{A}$, i.e.

$$\phi_{ij} = \sum_{i=1}^{NI} \psi_i^{[I]} \phi_j^{[I]} \quad Z_{ij} = \sum_{i=1}^{NI} \psi_i^{[I]} Z_j^{[I]}, \tag{34}$$

with

$$P_{\lambda\lambda} \psi^{[I]} = \varepsilon_I P(\lambda) \psi^{[I]}, I = 1, NI, \tag{35}$$

and the property of orthogonality

$$\sum_{i=1}^{NI} \psi_i^{[I']} P(\lambda)_{ii} \psi_i^{[I]} = \delta_{\mathbf{II'}}, \tag{36}$$

is used to project (15) on each mode in turn. The result written in matrix form is obtained as

$$\mathbf{A}^{[I]} Z^{[I]} + \eta I^{[I]}(\theta) \phi^{[I]} = \mathbf{r}^{[I]}; \quad \mathbf{A}^{[I]} \phi^{[I]} - I^{[I]}(\theta) Z^{[I]} = 0, \quad I = 1,..NI, \tag{37}$$

where

$$\mathbf{A}^{[I]} = \left[-\varepsilon_I P'(\theta) - P_{\theta\theta} \right]; \quad I^{[I]}(\theta) = P(\theta). \tag{38}$$

The algorithm can then be summarized as:

1. analysis of the right-hand side: $\mathbf{r}_j^{[I]} = \sum_{i=1}^{NI} \psi_i^{[I]} \mathbf{r}_{ij}$,
2. solution of the problems in equation (37),
3. synthesis of the solution, $\phi_{ij} = \sum_{i=1}^{NI} \psi_i^{[I]} \phi_j^{[I]}$,

As seen in the direct solution of the Helmholtz problem, a uniform λ -grid the modes ψ^I are proportional to the usual Fourier modes. The analysis and synthesis steps can then be simply implemented with a real Fourier transform, and if *NI* also factorizes properly, the Fast Fourier Transform (FFT) algorithm can be applied. If the NI cannot be divided properly, these steps would become the "slow" transform case using the full matrix multiplication (MXMA). The *NI* problems in (37) can be then transformed to *NI* block tridiagonal problems to solve. With contrast to (Yong et al., 1994) where the recursive method in (Lindzen & Kuo, 1969) is used, a block LU factorization is used in this paper.

Let $X_j = \begin{bmatrix} Z_j^{[I]} \\ \phi_j^{[I]} \end{bmatrix}$, with $X_0 = \begin{bmatrix} 0 \\ 0 \end{bmatrix} = X_{NJ+1}$, the system in (37) can be written as

$$\begin{bmatrix}\begin{bmatrix} A_{j,j-1}^{[I]} & 0 \\ 0 & A_{j,j-1}^{[I]}\end{bmatrix}\begin{bmatrix} A_{j,j}^{[I]} & \left(\eta * P_{jj}(\theta)\right) \\ -P_{jj}(\theta) & A_{j,j}^{[I]}\end{bmatrix}\begin{bmatrix} A_{j,j+1}^{[I]} & 0 \\ 0 & A_{j,j+1}^{[I]}\end{bmatrix}\end{bmatrix}\begin{bmatrix} X_{j-1} \\ X_j \\ X_{j+1}\end{bmatrix} = \begin{bmatrix} r_j^{[I]} \\ 0\end{bmatrix} \quad j=1,..NJ, \tag{39}$$

i.e.,

$$M \quad X = B\,, \tag{40}$$

where M is a matrix of dimension $2 * NJ$, or a block tridiagonal matrix of dimension NJ where each element is a 2 by 2 matrix. We can then represent M in the form

$$M = \begin{bmatrix} D_1 & E_1 & & & \\ F_2 & D_2 & E_3 & & \\ & F_3 & \ddots & \ddots & \\ & & \ddots & D_{NJ-1} & E_{NJ-1} \\ & & & F_{NJ} & D_{NJ}\end{bmatrix}, \tag{41}$$

where blocks E_i and F_i are 2 by 2 diagonal matrices, and D_i are 2 by 2 full matrices. Let D be the block-diagonal matrix consisting of the diagonal blocks D_i, L the block strictly-lower triangular matrix consisting of the sub-diagonal blocks F_i, and U the block strictly-upper triangular matrix consisting of the super-diagonal blocks E_i. Then the above matrix M has the form

$$M = L + D + U\,. \tag{42}$$

A block LU factorization of M is defined by

$$M = (L + \Delta)\Delta^{-1}(\Delta + U)\,, \tag{43}$$

where L and U are the same as above, and Δ_i are of dimension 2 by 2 and are defined by the recurrence :

$$\Delta_1 = D_1\,; \quad \Delta_i = D_i - F_i\,\Delta_{i-1}^{-1}\,E_i\,; \quad i = 2,..NJ\,, \tag{44}$$

where the 2 by 2 Δ_i^{-1} matrices are explicitly and exactly calculated. It is obvious that this set up is done once and its initial cost would be amortized over many solutions. The solution consists of standard forward elimination and backward substitution for the following triangular systems:

$$(L + \Delta)Y = B; \quad (U + \Delta)X = \Delta * Y\,. \tag{45}$$

The algorithm can now then be summarized as:

1. analysis of the right-hand side: $\mathbf{r}_j^{[I]} = \sum_{i=1}^{NI} \psi_i^{[I]} \mathbf{r}_{ij}$,
2. solution of the NI block tridiagonal problems $M \quad X = B$,
3. synthesis of the solution, $\phi_{ij} = \sum_{i=1}^{NI} \psi_i^{[I]} \phi_j^{[I]}$.

The above algorithm has been extended to the diffusion equation of any order $m = 2,4,6,8..$ Step 1 which consists of the analysis of the right-hand side and Step 3, which is the synthesis of the solution are almost identical. The elements of block matrices L, U and D become dimensions $m/2$ by $m/2$.

4. Parallel Direct Algorithms

The domain decomposition (mapping of grid points to processors) in GEM as in most large scale weather prediction models, is purely horizontal. This means that each processor owns all the NK vertical grid points in a given horizontal sub-domain. A parallel Direct solver for Helmholtz problem has been developed in (Qaddouri et al., 1999) and the same idea is applied for the high order diffusion equation. The direct solution algorithm presented in the previous sections has a great potential for vectorization and parallelism since at each step, there is recursive work along only one direction at a time, leaving the others available for vectorization. The remapping (or transposition) method is used for interprocessors communication.

The algorithm necessitates 4 global communication steps and reads for a problem with $NI \times NJ \times NK$ grid points and $P \times Q$ processors grid:

1. transposition of the right-hand sides to bring all the λ -direction grid points in each processor, the distribution becomes in θ -z subdomains:

$$\left(\frac{NI}{P}, \frac{NJ}{Q}, NK\right) \rightarrow \left(\frac{NJ}{P}, \frac{NK}{Q}, NI\right), \tag{46}$$

2. analysis of the right-hand sides,
3. transposition of the data to bring all the θ -direction grid points in each processor, the distribution becomes in z- λ subdomains:

$$\left(\frac{NJ}{Q}, \frac{NK}{P}, NI\right) \rightarrow \left(\frac{NK}{P}, \frac{NI}{Q}, NJ\right), \tag{47}$$

4. solutions of the block tri-diagonal problems,
5. inversion of the second transpose with the solutions of step 4:

$$\left(\frac{NK}{P}, \frac{NI}{Q}, NJ\right) \rightarrow \left(\frac{NJ}{Q}, \frac{NK}{P}, NI\right), \tag{48}$$

6. synthesis of the solutions,
7. inversion of the first transposition, the solutions are again distributed in λ - θ subdomains:

$$\left(\frac{NJ}{Q}, \frac{NK}{P}, NI\right) \rightarrow \left(\frac{NI}{P}, \frac{NJ}{Q}, NK\right). \tag{49}$$

5. Numerical Tests

All numerical tests presented in this chapter are done using the current forecast model GEM-LAM. In any given meteorological site, there are always time constraints when implementing operational runs. At CMC, both MPI and OpenMP are used to parallelize the GEM model in order to gain the best performance possible. The MPI is used for the communication between the sub-domains which has a global effect on the model. The OpenMP is applied to each sub-domain on selected time consuming functions or loops where their algorithms have no data dependencies and no communication. All tests were done on the IBM p575+ cluster with 121 compute nodes where each node contains 16 processors. Within each node, the 16 processors share a memory of 52G. The timings presented in Table 1 measure all the major components of a GEM-LAM model run optimized as much as possible using the combined effect of MPI and OpenMP. The model configuration for this table is from one of CMC's operational LAM (2.5km horizontal uniform-resolution) where *NI*=565, *NJ*=494 and *NK*=58 and the MPI topology was set at *P*=6, (processors along the longitude), *Q*=12, (processors along the latitude) and the OpenMP configuration was set to OpenMP=4. Note that timings of the solver and of the diffusion are not negligible compared to the other model components.

Components	Time (sec)	Percentage of Total runtime
Right-Hand Side	26.29	.90
Advection	626.66	21.39
Preparation for Non-Linear	47.49	1.62
Non-Linear	125.59	4.29
Solver (fft)	**187.03**	**6.38**
Back Substitution	94.33	3.22
Physics	985.61	33.65
Horizontal Diffusion	**205.32**	**7.01**
Vertical Sponge	67.12	2.29
Vertical Diffusion	335.0	11.44
Nesting and blending	127.37	4.34
Initialization and Setup	32.5	1.11
Output dynamic fields	54.42	1.86
Output physics fields	14.56	0.50
Total	2929.29	100.0

Table 1. Breakdown of model timings in the major components of a Canadian 2.5km Limited Area Model (LAM) over the eastern part of Canada for an integration of 24 hours on 18 nodes (6 x 12 x 4 PEs) .

5.1 Numerical Tests using MPI only

The experiments are designed to test the effects of MPI and OpenMP separately on the performance for the different solvers (direct MXMA-based , direct FFT-based and Iterative) and for the FFT and MXMA in implicit horizontal diffusion. The model configuration used here for these tests is a LAM (limited area modelling) at a 2.5km horizontal resolution with a setup similar to an operational run. In the experiments to test the MPI only, the number of processors were increased using MPI and at the same time, the number of computational

points were maintained the same for each tile. In other words, each tile always worked on approximately 96 by 96 points by 80 levels in order to analyze the behaviour of the communication costs as the number of processors increases. Figure 1 displays the regions of interest in the series of MPI test runs. The number of grid points along the latitude was chosen in such a way that the FFT solver could also be included in the study. Table 2 displays the timings of some of the major components of the model for the MPI test runs where the MPI topology is denoted as *P*x*Q* and the OpenMP=1. It can be seen up to 25 nodes that, the increase in timings is mostly due to communication in the solver and in the diffusion (FFT-based) whereas, in the physics and the advection modules, there were no communications at all. In the experiments using 100 nodes, almost the entire cluster was utilized. There seemed to be a dramatic increase in the timings of almost all the model components for these huge runs and it is suspected that either the memory or the communication was limited. Further investigation is needed for these large integrations as it is anticipated that this could be the size of domain needed for the future of weather forecasting.

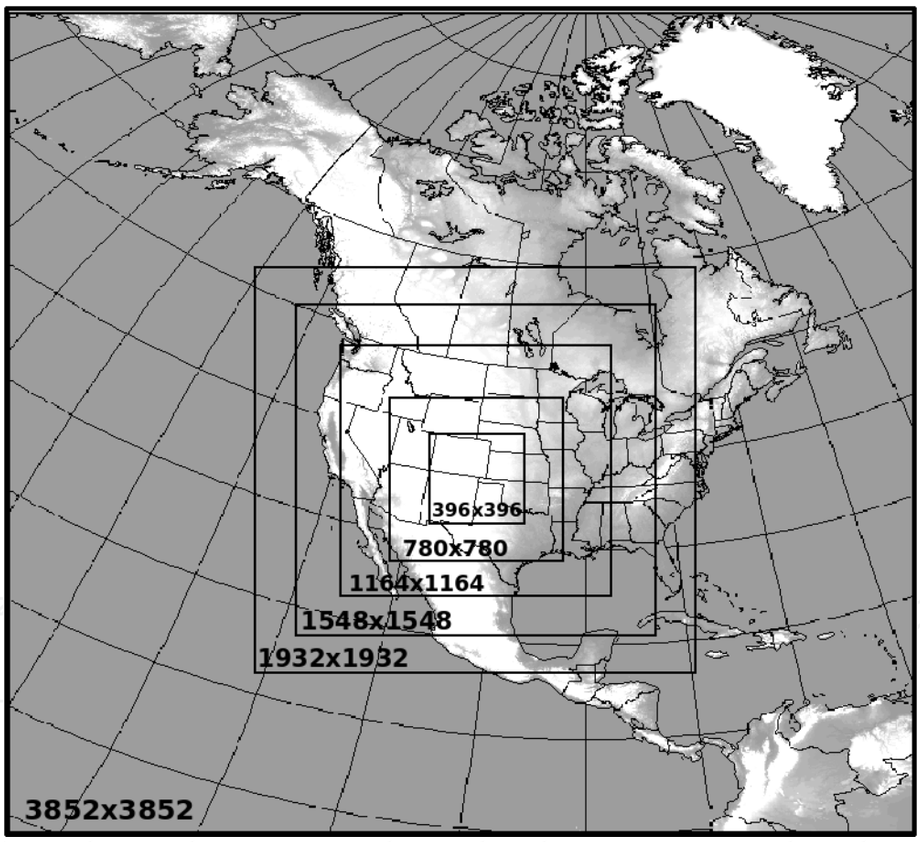


Fig. 1. LAM 2.5km resolution topography used in the MPI test runs where the results are recorded in Table 2. The number of vertical levels is 80.

Nodes	*P* x *Q* (*NI* x *NJ* pts)	# of PEs	FFT solver (secs) (count=100)	Advection (secs)	Physic (secs)	FFT Hzd (secs) (count=100)
1	4x4 (396x396)	16	23.32	149	241	47
4	8x8 (780x780)	64	30.41	153	241	52
9	12x12 (1164x1164)	144	36.7	148	238	53
16	16x16 (1548x1548)	256	44.1	152	238	61
25	20x20 (1932x1932)	400	57.0	151	240	77
100	40x40 (3852x3852)	1600	109	190	239	133

Table 2. Breakdown of model timings in the major components of the MPI test runs using a LAM grid at 2.5km resolution with 80 levels. The grid is increased to give almost the same number of points per tile for each test run (There is no OpenMP).

Applying the same MPI tests (identical configurations) but just changing the solvers, Table 3 displays the timing comparisons between using the FFT-based direct solver, the MXMA-based direct solver and the newly implemented PGMRES-based iterative solver. Note that the optimization for the iterative solver has not yet been completely refined but, it looks promising as the number of iterations required remains relatively steady.

Nodes	*P*x*Q* (*NI* x *NJ*) points	# of PEs	FFT solver (secs)	MXMA solver (secs)	Iterative solver (secs)	# of iterations
1	4x4 (396x396)	16	23.31	56.5	76.0	4
4	8x8 (780x780)	64	30.41	104.9	89.6	4
9	12x12 (1164x1164)	144	36.7	149.7	95.5	5
16	16x16 (1548x1548)	256	44.1	201.9	121.31	5
25	20x20 (1932x1932)	400	57.0	282.6	132.9	5
100	40x40 (3852x3852)	1600	109	N/A	195.85	5

Table 3. Comparison of timings between using the FFT, MXMA and the iterative Jacobi solver in the MPI test runs (The number of solver calls is 100).

A plot of the timings versus the number of nodes used per run is shown in Figure 2 and it can be seen that the timings become longer due to increased communication as more nodes were added. In the runs of the measured times in the FFT solver, there is an increase of 86 seconds from a run using one node to 100 nodes giving a factor of increase of 3.7 times over the one node run. By doing the same comparison for the runs using the iterative solver, there is an increase of 120 seconds which gives a factor increase of 1.6 times over its one

node run. This shows that the communications cost more in proportion to the cost of computation in the FFT solver than in the iterative solver. It would be worth the effort to improve the performance of the iterative solver more within one node by using an approximate local solver which will diminish the computing time spent in the preconditioning operation.

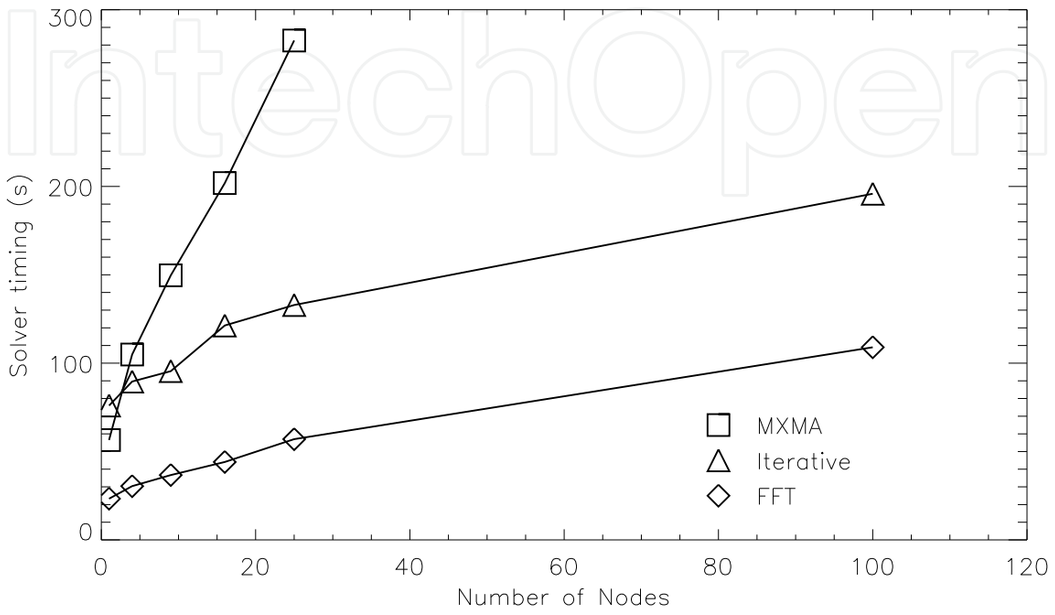


Fig. 2. Timings for FFT, MXMA and Iterative solvers in MPI test runs (computation on approximately 96 pts x 96 pts x 80 levels per processor)

Similar results are seen in Table 4 when comparing the two types of horizontal diffusion. All numerical tests presented here for the horizontal diffusion were using the sixth order implicit horizontal diffusion (m=6 in Eq.25).

Nodes	P x Q (NI x NJ points)	# of PEs	FFT diffusion (secs)	MXMA diffusion (secs)
1	4x4 (396x396)	16	47	73
4	8x8 (780x780)	64	52	129
9	12x12 (1164x1164)	144	53	179
16	16x16 (1548x1548)	256	61	252
25	20x20 (1932x1932)	400	77	360
100	40x40 (3852x3852)	1600	133	N/A

Table 4. Comparison of timings between using the FFT and MXMA in the horizontal diffusion solver within the MPI test runs (The number of diffusion calls is 100).

5.2 Numerical Tests on OpenMp

Tests for OpenMP were made using two different MPI configurations but using the same LAM grid configuration. The region of interest of the grid is shown in Figure 1 using 1548 by 1548 grid points. The FFT results are not included in this study as the OpenMP section for it was not correctly implemented. One of the OpenMP tests made is to vary the OpenMP on a MPI topology of 16 by 16 and these results are shown in Table 5. The other test is to vary the OpenMP on a MPI topology of 8 by 8 and these results are shown in Table 6. The MPI topology and OpenMP configuration is denoted by (*P*x*Q*xOpenMP). Note that with the same size LAM grid, the configuration with more MPI is more efficient than with the one with more OpenMP. Given the same number of PEs (using 256 or using 512), the MPI topology of 16 by 16 out-performs the one with more weight given to the OpenMP. This is an expected result as the model was parallelized with MPI entirely, and then with OpenMP in sub-sections of the code.

OpenMP	# of PEs	MXMA solver (secs)	MXMA Relative Speedup	Jacobi solver (secs)	Jacobi Relative Speedup
1	256	201.9	1	121.31	1
2	512	111.17	1.8	69.82	1.74
4	1024	68.13	2.96	45.53	2.6

Table 5. Comparison of timings between using the Jacobi Iterative and the MXMA solver within the OpenMP test runs where the grid is (1548x1548x80Levels) using the MPI topology of (16x16xOpenMP). The number of solver calls is 100.

OpenMP	# of PEs	MXMA solver (secs)	MXMA Relative Speedup	Jacobi solver (secs)	Jacobi Relative Speedup
1	64	1135.08	1	1443.6	1
2	128	631.04	1.8	824.89	1.72
4	256	231.25	4.9	216.71	6.66
8	512	136.21	8.33	137.73	10.46

Table 6. Comparison of timings between using the Jacobi Iterative and the MXMA solver within the OpenMP test runs where the grid is (1548x1548x80Levels) using the MPI topology of (8x8xOpenMP). The number of solver calls is 100.

6. Conclusion

In this paper, we have examined the performance of the elliptic solvers in the context of atmospheric modelling with a limited area grid. By using only the MPI paradigm (see Table.3) we can see that by increasing the grid size relative to the increase of the number of processors (in order that each processor keeps same amount of work), the number of iterations needed for the iterative solver to converge remains constant. We can say that the block-Jacobi preconditioner used in GEM-LAM is robust even though it involves no inter-processor communications and it is based on a single level additive Schwarz. We can also

say that the non-scalability in real time is due to local inter-processor communications involved in matrix-vector operations and also due to the global communications (all-reduce) in the global dot product operations. It was also shown that by using the hybrid of MPI and OpenMP correctly, the parallel direct and iterative solvers present good scalability and even super-scalability in the aspect of OpenMP (Table.6). By implementing all these tests, we have also discovered that OpenMP was not well applied in the Fast Direct solver which is why the poor results are omitted from this paper. As it was noted earlier, the iterative solver needs more work in its optimization. In the future, we will introduce the second level in the additive Schwartz preconditioner (coarse grid) and we will employ a local approximate solver instead of an exact local direct solver which should reduce the computing cost.

7. References

Arakawa A. and V. R. Lamb, 1977: Computational design of the basic dynamical processes of the UCLA general circulation model. *Methods in Computational Physics*, Vol. 17, 174-267, J.Chang, Ed., Academic Press, 1977,

Coté, J., S. Gravel, A. Methot, A. Patoine, M.Roch and A. Staniford, 1998: The operational CMC-MRB Global Environmental Multiscale (GEM) model, Part I: design considerations and formulation. *Mon. Wea.* Rev. 126, 1373-1395

Denis, B., Coté, J. and R. Laprise, 2002: Spectral decomposition of two-dimensional atmospheric fields on limited-area domains using the discrete cosine transform (DCT). *Mon. Wea.* Rev. 130, 1812-1829.

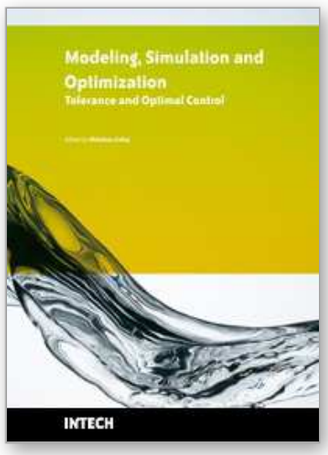
Lindzen, R. S. and Kuo, H.L. 1969: *A reliable method for the numerical integration of a large class of ordinary and partial differential equations*. Mon. Wea. Rev., 97, 732-734

Qaddouri, A., J. Coté and M. Valin, 1999: A parallel direct 3D elliptic solver. *Proceedings of the 13th Annual International Symposium on High Performance Computing Systems and Applications*, Kingston, Canada, June 13-16, 1999, Kluwer Academic Publishers, 2000, 429-442

Saad, Y. and A. Malevsky, 1995: Data Structures, Computational, and Communication Kernels for Distributed Memory Sparse Iterative Solvers, *Proceedings of Parallel Computing Technologies (PaCT-95), 3rd International conference*, In: Lecture Notes in Computer Science series, Vol. 964, 252-257, ISSN 0302-9743, St. Petersburg, Russia, September 12-15, 1995, Victor Malyshkin (Ed.), Springer Verlag,

Saad, Y., 1996: Iterative Methods for Sparse Linear Systems, PWS Publishing Co International Thomson Publishing Inc, ISBN 0-534-94776-X, 1996.

Yong Li, S. Moorthi, J.R. Bates, 1994: Direct solution of the implicit formulation of fourth order horizontal diffusion for grid point models on the sphere, *NASA GLA Technical report series in atmospheric modelling and data assimilation*, Vol.2



Modeling Simulation and Optimization - Tolerance and Optimal Control

Edited by Shkelzen Cakaj

ISBN 978-953-307-056-8
Hard cover, 304 pages
Publisher InTech
Published online 01, April, 2010
Published in print edition April, 2010

Parametric representation of shapes, mechanical components modeling with 3D visualization techniques using object oriented programming, the well known golden ratio application on vertical and horizontal displacement investigations of the ground surface, spatial modeling and simulating of dynamic continuous fluid flow process, simulation model for waste-water treatment, an interaction of tilt and illumination conditions at flight simulation and errors in taxiing performance, plant layout optimal plot plan, atmospheric modeling for weather prediction, a stochastic search method that explores the solutions for hill climbing process, cellular automata simulations, thyristor switching characteristics simulation, and simulation framework toward bandwidth quantization and measurement, are all topics with appropriate results from different research backgrounds focused on tolerance analysis and optimal control provided in this book.

INTECH
open science | open minds

InTech Europe
University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China
Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821

IntechOpen