Preliminary Results of GRAPES’ Helmholtz solver using GCR and PETSc tools

Xiangjun Wu(1), Lilun Zhang(2), Junqiang Song(2) and Dehui Chen(1)
(1) Center for Numerical Weather Prediction, CMA
(2) School of Computer Science, National University of Defense Technology
wuxj@cma.gov.cn
Outline:

- Some features of GRAPES-global model
- Using GCR-CL scheme for solving Helmholtz equation
- Using PETSc tools for solving Helmholtz equation
- Some tests
- Summary and future work
Main features of the GRAPES Global model

- Latitude-longitude grid point, spherical coordinate
- Non-hydrostatic/Hydrostatic optionally
- Height terrain-following coordinate
- Charney-Phillips variable staggering in vertical
- 2 time-level SISL scheme
- 3D preconditioned GCR solver for Helmholtz equation
- Arakawa-C grid horizontally: v at poles
- Polar filtering
Arakawa-C grid for horizontal arrangement of variables for GRAPES model:
Charney-Phillips vertical arrangement of variables:

<table>
<thead>
<tr>
<th>Top level</th>
<th>Charney-Phillips</th>
<th>Vertical level</th>
<th>Lorenz</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$z, \hat{z}, \theta, w, \hat{w}$</td>
<td>$k = N$</td>
<td>$z, \hat{z}, w, \hat{w}$</td>
</tr>
<tr>
<td></td>
<td>$z, \hat{z}, \rho, P, u, v$</td>
<td>$k = N - 0.5$</td>
<td>$z, \hat{z}, \theta, \rho, P, u, v$</td>
</tr>
<tr>
<td></td>
<td>$z, \hat{z}, \theta, w, \hat{w}$</td>
<td>$k = N - 1.0$</td>
<td>$z, \hat{z}, w, \hat{w}$</td>
</tr>
<tr>
<td></td>
<td>$z, \hat{z}, \rho, P, u, v$</td>
<td>$k = N - 1.5$</td>
<td>$z, \hat{z}, \theta, \rho, P, u, v$</td>
</tr>
<tr>
<td>Earth's surface</td>
<td>$z, \theta, w, \hat{w}$</td>
<td>$k = 2.0$</td>
<td>$z, \hat{z}, w, \hat{w}$</td>
</tr>
<tr>
<td></td>
<td>$z, \hat{z}, \rho, P, u, v$</td>
<td>$k = 1.5$</td>
<td>$z, \hat{z}, \theta, \rho, P, u, v$</td>
</tr>
<tr>
<td></td>
<td>$z, \hat{z}, \theta, w, \hat{w}$</td>
<td>$k = 1.0$</td>
<td>$z, \hat{z}, w, \hat{w}$</td>
</tr>
<tr>
<td></td>
<td>$z, \hat{z}, \rho, P, u, v$</td>
<td>$k = 0.5$</td>
<td>$z, \hat{z}, \theta, \rho, P, u, v$</td>
</tr>
<tr>
<td></td>
<td>$z, \theta, w, \hat{w}$</td>
<td>$k = 0$</td>
<td>$z, \hat{z}, w, \hat{w}$</td>
</tr>
</tbody>
</table>
Prediction equation for GRAPES model:

\[ u^{n+1} = \left[ \xi_1 \frac{1}{a \cos \phi} \frac{\partial}{\partial \lambda} + \xi_2 \frac{1}{a} \frac{\partial}{\partial \varphi} + \xi_3 \frac{\partial}{\partial z} \right] (\Pi)^{n+1} + \xi_0 \]  

\[ v^{n+1} = \left[ \xi_1 \frac{1}{a \cos \phi} \frac{\partial}{\partial \lambda} + \xi_2 \frac{1}{a} \frac{\partial}{\partial \varphi} + \xi_3 \frac{\partial}{\partial z} \right] (\Pi)^{n+1} + \xi_0 \]  

\[ w^{n+1} = \left[ \xi_1 \frac{1}{a \cos \phi} \frac{\partial}{\partial \lambda} + \xi_2 \frac{1}{a} \frac{\partial}{\partial \varphi} + \xi_3 \frac{\partial}{\partial z} \right] (\Pi)^{n+1} + \xi_0 \]  

\[ (\theta)^{n+1} = \xi_4 \frac{\partial}{\partial z} (\Pi)^{n+1} + \xi_5 \]  

\[ (\Pi)^{n+1} = \xi_\Pi u^{n+1} + \xi_\Pi v^{n+1} + \xi_\Pi w^{n+1} + \xi_\Pi (D_3) + A_\Pi \]
Equation (5) is Helmholtz equation. Solution of Helmholtz equation is key for GRAPES mode computing. Due to variables $u, v, w, \theta$ represented by function of $\pi$, it’s possible to solve other variables only after its solution.
Let the Helmholtz equation is:

$$A\vec{x} = \vec{b}_0$$

The approximate solution is:

$$\vec{x}_0$$

The residual is:

$$\vec{r} = \vec{b}_0 - A\vec{x}_0$$

The preconditioner is:

$$M$$

So the Helmholtz equation with precondition is:

$$(M^{-1}A)\vec{x} = M^{-1}\vec{b}$$
Arrangement of 19 coefficients for Helmholtz equation of GRAPES:

- The coefficient matrix of Helmholtz equation is a larger sparse matrix.
- The values of $B_1, B_{10}, B_{15}, B_2, B_3$ are more larger than others.
**Precondition:**

- A good preconditioner requires fewer iterations to converge for a given sparse linear system.

- Roughly speaking, a preconditioner is any form of implicit or explicit modification of the original coefficient matrix.

- For GRAPES model, choosing the main elements to build the preconditioner maybe a good idea.
For 19 coefficients, choose $B_1, B_{10}, B_{15}, B_2, B_3$ to build the precondition matrix.
The escaped time ratio for main subroutines of 0.5° GRAPES Global model (1 day forecast on 64)

The higher the model resolution, the higher the time ratio.
The Solver for Helmholtz equation of GRAPES model:

1. Using GCR (Generalised Conjugate Residual) with preconditioner algorithm (called it as gcr-cl now used in GRAPES);

2. Using PETSc ksp solver + hypre preconditioner.
What’s PETSc?  (http://www.mcs.anl.gov/petsc/petsc-as/)
the Portable, Extensible Toolkit for Scientific Computation

PETSc is a suite of data structures and routines for the scalable (parallel) solution of scientific applications modeled by partial differential equations. It consists of a variety of libraries which manipulates a particular family of objects (such as vectors, matrix, distributed arrays, Krylov subspace method, preconditioners, et).
Organization of the PETSc Libraries

- Level of abstraction
  - Application codes
    - PDE Solvers
    - SNES (Nonlinear Equations Solvers)
    - KSP (Krylov Subspace Methods)
    - Matrices
    - BLAS
    - LAPACK
    - MPI
  - TS (Time stepping)
    - SLES (Linear Equations Solvers)
    - PC (Preconditioners)
    - Vectors
    - Index Sets
    - Draw
### Parallel Numerical Components of PETSC

<table>
<thead>
<tr>
<th>Nonlinear Solvers</th>
<th>Time Steppers</th>
</tr>
</thead>
<tbody>
<tr>
<td>Newton-based Methods</td>
<td>Euler</td>
</tr>
<tr>
<td>Line Search</td>
<td>Backward Euler</td>
</tr>
<tr>
<td>Trust Region</td>
<td>Pseudo-Time Stepping</td>
</tr>
<tr>
<td>other</td>
<td>other</td>
</tr>
</tbody>
</table>

#### Krylov Subspace Method

<table>
<thead>
<tr>
<th>GMRES</th>
<th>CG</th>
<th>CGS</th>
<th>Bi-CG-Stab</th>
<th>TFQMR</th>
<th>Richardson</th>
<th>Chebyshev</th>
<th>other</th>
</tr>
</thead>
</table>

#### Precondition

<table>
<thead>
<tr>
<th>Additive Schwarz</th>
<th>Block Jacobi</th>
<th>Jacobi</th>
<th>ILU</th>
<th>ICC</th>
<th>LU (sequential only)</th>
<th>other</th>
</tr>
</thead>
</table>

#### Matrices

<table>
<thead>
<tr>
<th>Compressed Sparse Row (AIJ)</th>
<th>Block Compressed Sparse Row (BAIJ)</th>
<th>Block Diagonal (BDiag)</th>
<th>Dense</th>
<th>other</th>
</tr>
</thead>
</table>

#### Vectors

<table>
<thead>
<tr>
<th>Indices</th>
<th>Block Indices</th>
<th>Stride</th>
<th>other</th>
</tr>
</thead>
</table>
What’s Hypre? (http://acts.nersc.gov/hypre/)

high performance preconditions

Hypre is a library for solving large, sparse linear systems of equations on massively parallel computers. Hypre contains several families of preconditioner algorithms focused on the scalable solution of very large sparse linear systems. These algorithms include structured multigrid and element-based algebraic multigrid.
Global and Regional Assimilation and PrEdiction System

A example subroutine using PETSc functions:

...  
  If(nstep==1)  
    Call PetscInitialize()  
    ...  
    Call MatCreateMPIAIJ()  
    ...  
    Call set_A-petsc()  
    ...  
    Call KspCreate()  
    ...  
    Endif  
    ...  
  Call set_b_petsc()  
  ...  
  Call kspSolver()  
  ...  
  Call PetscFinalize()  
  stop
Test for Rossby-Haurwitz wave

- 1.0° resolution for global model (grid size: 360x180x36)
- The residual $l_2$-norm $< 10^{-9}$ for convergence
- Main variables are double precision
- Run on IBM-cluster
- Use 64 MPI tasks
- 80days forecast
- For PETSc scheme, use GMRES+ BoomerAMG(hypre)
Global and Regional Assimilation and PrEdiction System
Global and Regional Assimilation and PrEdiction System
### Table1 : Escaped Time for 80day forecast of Rossby-Haurwitz wave with 19 coefficients on IBM-cluster1600

<table>
<thead>
<tr>
<th>scheme</th>
<th>gcr-cl</th>
<th>Petsc</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Time helm/total</td>
<td>Number of iterations</td>
</tr>
<tr>
<td><strong>Step 1</strong></td>
<td>5.11/7.71</td>
<td>275</td>
</tr>
<tr>
<td><strong>Step 520</strong></td>
<td>3.70/4.66</td>
<td>196</td>
</tr>
<tr>
<td><strong>Step 5319</strong></td>
<td>2.27/3.23</td>
<td>147</td>
</tr>
<tr>
<td><strong>Wall clock</strong></td>
<td><strong>5h37m11.40s</strong></td>
<td></td>
</tr>
</tbody>
</table>
Global and Regional Assimilation and PrEdiction System

Test for real data

- 0.5° resolution for global model (grid size: 720x361x36)
- Main variables are double precision
- On IBM-cluster 1600
- Use 64 processors
- Initial data: 31 May 2007
- 60 min for radiation
- 1 days forecast, 144 steps, DT=600s
- For petsc scheme, use GMRES+ BoomerAMG(hypre)
- Test1 for convergence using residual $l_2$-norm < $10^{-7}$
- Test2 for convergence using residual $l_2$-norm < $10^{-12}$
Global and Regional Assimilation and PrEdiction System
Global and Regional Assimilation and PrEdiction System
### Table2(test1) : Escaped Time for test of real data on IBM-cluster 1600(residual $l^2$-norm $<10^{-7}$)

<table>
<thead>
<tr>
<th>convergence</th>
<th>E07-d-org</th>
<th>E07-petsc</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Time helm/total</td>
<td>Number of iterations</td>
</tr>
<tr>
<td><strong>Step 1</strong></td>
<td>19.01/80.56</td>
<td>184</td>
</tr>
<tr>
<td><strong>Step 42</strong></td>
<td>15.32/46.58</td>
<td>147</td>
</tr>
<tr>
<td><strong>Step 137</strong></td>
<td>13.45/23.37</td>
<td>147</td>
</tr>
<tr>
<td><strong>Wall clock</strong></td>
<td>1h13m53.93s</td>
<td></td>
</tr>
</tbody>
</table>
Table 3 (test2) : Escaped Time for test of real data on IBM-cluster1600 (residual $l^2$-norm <10^{-12})

<table>
<thead>
<tr>
<th>convergence</th>
<th>E12-d-org</th>
<th>E12-petsc</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Time helm/total</td>
<td>Number of iterations</td>
</tr>
<tr>
<td>Step 1</td>
<td>45.61/107.76</td>
<td>623</td>
</tr>
<tr>
<td>Step 42</td>
<td>38.16/69.33</td>
<td>519</td>
</tr>
<tr>
<td>Step 137</td>
<td>37.29/47.22</td>
<td>511</td>
</tr>
<tr>
<td>Wall clock</td>
<td>2h12m22.18s</td>
<td></td>
</tr>
</tbody>
</table>
Fig: One step time of helmholtz solver for different number of processor (residual $l^2$-norm <10$^{-12}$)
Fig: wall clock of 1 day forecast and the efficiency for different number of processor (residual $l^2$-norm $<10^{-12}$)
Table 4: Escaped Time for real data test on Intel based machine (residual $l_2$-norm $<10^{-12}$)

<table>
<thead>
<tr>
<th>Number processor</th>
<th>Original scheme</th>
<th>PETSc tools</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Time helm for one step</td>
<td>Time helm for one step</td>
</tr>
<tr>
<td>1</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>4</td>
<td>171.885</td>
<td>-</td>
</tr>
<tr>
<td>8</td>
<td>155.88</td>
<td>53.18</td>
</tr>
<tr>
<td>16</td>
<td>83.04</td>
<td>25.16</td>
</tr>
<tr>
<td>32</td>
<td>42.64</td>
<td>13.93</td>
</tr>
<tr>
<td>64</td>
<td>25.47</td>
<td>8.12</td>
</tr>
<tr>
<td>128</td>
<td>21.14</td>
<td>3.97</td>
</tr>
<tr>
<td>256</td>
<td>11.24</td>
<td>2.16</td>
</tr>
</tbody>
</table>
summary:

- *Good Solution for Helmholtz equation using PETSc tools*
- *Same convergence precision, fewer iterations using PETSc for Solution of Helmholtz equation*
- *The solution for Helmholtz equation using PETSc has more well scalability*
- *The running time of two scheme is comparable, and at some platform using PETSc tools were more better.*
Future work:

- Testing more cases and give the profile.
- Testing different combination of hypre preconditioner plus GRMES.
- Testing with large number processors (more than 256 etc.) under high-resolution model.
- Optimize PETSc solution for Helmholtz equation.
- Comparable research on parallel algorithm library of preconditioner except hypre, such as Trillions and SuperLU-dist, etc.
Thank you!