A Global Cloud Resolving Model
Goals

- Uniform global horizontal grid spacing of 4 km or better (“cloud permitting”)
- 100 or more layers up to at least the stratopause
- Parameterizations of microphysics, turbulence (including small clouds), and radiation
- Execution speed of at least several simulated days per wall-clock day on immediately available systems
- Annual cycle simulation by end of 2011.
Motivations

- Parameterizations are still problematic.
- There are no spectral gaps.
- The equations themselves change at high resolution.
- GCRMs will be used for NWP within 10 years.
- GCRMs will be used for climate time-slices shortly thereafter.
- It’s going to take some time to learn how to do GCRMs well.
Scaling Science

Length, Spatial extent, #Atoms, Weak scaling

Convergence, systematic errors due to cutoffs, etc.

Initial Conditions, e.g. molecule, boundaries, Ensembles

Time scale Optimizations, Strong scaling

Simulation method, e.g. DFT, QMC or HF/SCF; LES or DNS
Basic design

- Non-hydrostatic
- Vertically propagating sound waves filtered
- Vorticity equation (instead of momentum equation)
- Mass and energy conserving
- Geodesic grid
- Z-coordinates (for now...)
Geodesic Grid

Icosahedron

Bisect each edge and connect the dots

Pop out onto the unit sphere

And so on, until we reach our target resolution...
# Some grids of interest

<table>
<thead>
<tr>
<th>Level of recursion</th>
<th>Number of grid columns</th>
<th>Distance between grid columns, km</th>
</tr>
</thead>
<tbody>
<tr>
<td>9</td>
<td>2,621,442</td>
<td>15.64</td>
</tr>
<tr>
<td>10</td>
<td>10,485,762</td>
<td>7.819</td>
</tr>
<tr>
<td>11</td>
<td>41,943,042</td>
<td>3.909</td>
</tr>
<tr>
<td>12</td>
<td>167,772,162</td>
<td>1.955</td>
</tr>
<tr>
<td>13</td>
<td>671,088,642</td>
<td>0.977</td>
</tr>
</tbody>
</table>
Jablonowski Test Case

- 262,1442 cells (15.64km) on 640 cores of franklin
- 850 hPa relative vorticity
### Scaling test of 3D-multigrid on Franklin

<table>
<thead>
<tr>
<th>Grid resolution</th>
<th>Number of cores</th>
<th>2560</th>
<th>5120</th>
<th>10240</th>
<th>20480</th>
</tr>
</thead>
<tbody>
<tr>
<td>41,943,042 (11) (3.909km)</td>
<td>19.57</td>
<td>10.96</td>
<td>5.56</td>
<td>2.87</td>
<td></td>
</tr>
<tr>
<td>167,088,642 (12) (1.955km)</td>
<td>85.76</td>
<td>39.37</td>
<td>21.91</td>
<td>10.84</td>
<td></td>
</tr>
</tbody>
</table>
## Scaling test of 3D-multigrid on Jaguar

- The **NCCS Cray XT5** with 181,000 cores
- 20 V-cycles
- 80 layers

<table>
<thead>
<tr>
<th>Time (s)</th>
<th>Number of cores</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>2560</td>
</tr>
<tr>
<td></td>
<td>10240</td>
</tr>
<tr>
<td></td>
<td>40960</td>
</tr>
<tr>
<td>167,088,642 (12) (1.955km)</td>
<td>80.123</td>
</tr>
</tbody>
</table>
## Full dynamical core on Franklin

<table>
<thead>
<tr>
<th>Grid</th>
<th>PEs (Nodes)</th>
<th>GFlop /sec</th>
<th>Sec/day</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>40 (10)</td>
<td>5.4</td>
<td>26</td>
</tr>
<tr>
<td>6</td>
<td>160 (40)</td>
<td>17.70</td>
<td>66</td>
</tr>
<tr>
<td>7</td>
<td>640 (160)</td>
<td>57.5</td>
<td>130</td>
</tr>
<tr>
<td>8</td>
<td>2560 (640)</td>
<td>168.30</td>
<td>355</td>
</tr>
<tr>
<td>9</td>
<td>2560 (640)</td>
<td>339.7</td>
<td>1403</td>
</tr>
<tr>
<td>10</td>
<td>5120 (1280)</td>
<td>638.3</td>
<td>5495</td>
</tr>
<tr>
<td>11</td>
<td>10240 (2560)</td>
<td>1366.4</td>
<td>20139</td>
</tr>
</tbody>
</table>

We think this can speed up by about a factor of two.
Key (rough) numbers

- ~40 million grid columns
- ~100 layers
- ~10 3D prognostic fields
- ~10 3D diagnostic fields
- ~0.4 TB per full write
- Time step ~ 10 seconds -- not just a stability issue
- Can use at least 20 K processors on XT5 -- probably 40 K
- Will produce about 5 simulated days per wall-clock day on 20 K processors with a 4 km grid spacing
- ~50000 processor hours/simulated day on Grid 11
Computational challenges

- Efficient execution on a very large number of processors
- Parallel I/O (especially O)
- Management and distribution of the voluminous model output
- Analysis and visualization

These are “infrastructure” issues that will be faced by anyone using a GCRM.
The API can be configured to allocate n nodes to serve as IO Aggregators.

The API is designed to support multiple parallel (or serial) IO layers
  pnetcdf, netcdf4, netcdf3…

Grid and associated data linearized so that the sequence of grid cells follows a self-similar space-filling two-dimensional curve

• Blocks within panels can be written as contiguous blocks
• Order not dependent on number of processors
• For parallel analysis, achieves good locality without special handling
Summary

- Qualitatively different
- Just barely feasible now
- Weak scaling and new "simulation method"
- Output volume huge but controllable
- Analysis and visualization challenges