A Conflict Between Chip Architecture and a Direction of Ocean Model Development

Alistair Adcroft
Princeton University

Atmosphere, Oceans, and Computational Infrastructure
California Institute of Technology
May 16-18, 2018
When will my models become non-competitive?

Alistair Adcroft
Princeton University

Atmosphere, Oceans, and Computational Infrastructure
California Institute of Technology
May 16-18, 2018
• General vertical coordinate algorithms, and new [physical] parameterizations of sub-grid scale processes, look more and more like column-wise code and some are non-vectorizable
  – This is non-optimal on existing architectures but worse moving forward
• GPUs offer significant speedups only on parts of the code
  – Effective speedup of the whole model is moderate, if even positive
• The general coordinate algorithms allow comparably very large time-steps which [currently] far out pace the benefits of GPUs
  – Are there ways to get the new algorithms to work on GPUs or TPUs?
• Sound waves are filtered out
• Surface gravity waves
  – \( c_{bt} \approx \sqrt{gH} \approx 200 \text{ m/s} \)
• Internal waves
  – \( c_{iw} \approx NH \approx 2 \text{ m/s} \)
• Currents
  – \( U \approx 1-3 \text{ m/s} \)
  – \( W_{eddy} \approx 10 \text{ m/day} \)
  – \( W_{IG \ waves} \approx 100 \text{ m/day} \)
  – \( W_{overtorns} \approx U \approx 1 \text{ cm/s} \)

At coarse resolutions, numerics are dominated by considerations of rotational dynamics
At fine resolutions, codes/methods should look more towards CFD
External mode continues to be a critical barrier

<table>
<thead>
<tr>
<th></th>
<th>100 km</th>
<th>25 km</th>
<th>1 km</th>
<th>100 m</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \Delta x/c_{bt} )</td>
<td>8 minutes</td>
<td>2 minutes</td>
<td>5 seconds</td>
<td>( \frac{1}{2} ) second</td>
</tr>
<tr>
<td>( \Delta x/c_{iw} )</td>
<td>12 hours</td>
<td>3 hours</td>
<td>8 minutes</td>
<td>4 minutes</td>
</tr>
<tr>
<td>( \Delta x/U )</td>
<td>1 day</td>
<td>2 hours</td>
<td>5 minutes</td>
<td>2 minutes</td>
</tr>
<tr>
<td>( f^{-1} )</td>
<td>1.9 hours</td>
<td>1.9 hours</td>
<td>1.9 hours</td>
<td>1.9 hours</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>50 m</th>
<th>10 m</th>
<th>1 m</th>
<th>1 cm</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \Delta z/W )</td>
<td>1 days</td>
<td>2 hours</td>
<td>15 minutes</td>
<td>10 seconds</td>
</tr>
</tbody>
</table>
• Equations are not up for debate!
  – Need to be solved with particular methods for consistency
  – Discrepancies give tsunamis instead of tides

• Small amount of computation
  – Almost linear

• Frequent communication

• Latency sensitive

\[
\begin{align*}
\partial_t u_k &= -\frac{1}{\rho_0} \nabla p + \ldots \\
\partial_t h_k &= \nabla \cdot (h_k u_k) \\
\partial_t \eta &= \sum_k \partial_t h_k = \nabla \cdot \sum_k h_k u_k = \nabla \cdot (H U) \\
\partial_t U &= -g \nabla \eta + \ldots
\end{align*}
\]

Solve with baroclinic \( \Delta t_{bc} \)

SSH (constraint):

Reconcile with:

Solve with barotropic \( \Delta t_{bt} \)

\[ \Delta t_{bt} \ll \Delta t_{bc} \]
Adjustable strategies for barotropic solver

- **Low latency/low bandwidth**
  - Send small packets frequently

- **High bandwidth/high latency**
  - Send large packets infrequently

Signal propagates across many cores in a single baroclinic time step
Vertical coordinates and spurious mixing

- Long debate about appropriate vertical coordinate
  - Now fairly well understood

- Spurious mixing:
  - grid Reynolds number + ...
  - long time scale questions
A.L.E. allows you to work with any coordinate

Before the remap

Topography looks different

Very thin layers

White, Adcroft & Hallberg, JCP 2009
### Two basic algorithms

**Eulerian**

\[ \partial_z p = -g \rho (z, S^n, \theta^n) \]

\[ v_{h}^{n+1} = v_{h}^{n} + \Delta t \left( -\frac{1}{\rho_o} \nabla_z p + \ldots \right) \]

\[ \partial_z w = -\nabla \cdot v_{h}^{n+1} \]

\[ \theta^{n+1} = \theta^n - \Delta t \left[ \nabla \cdot (v_{h}^{n+1} \theta^n) + \ldots \right] \]

\[ \frac{\Delta t w}{\Delta Z} < 1 \]

---

**A.L.E. (flavor 1)**

\[ \partial_z p = -g \rho (z, S^n, \theta^n) \]

\[ v_{h}^{n+1} = v_{h}^{n} + \Delta t \left( -\frac{1}{\rho_o} \nabla_z p + \ldots \right) \]

\[ \delta_k (w^* + w_g) = -\nabla \cdot h^n v_{h}^{n+1} \]

\[ h_{n+1} = h^n + \Delta t \delta_k (w_g) \]

\[ h_{n+1}^{n+1} \theta^{n+1} = h^n \theta^n \]

\[ -\Delta t \left[ \nabla \cdot (h^n v_{h}^{n+1} \theta^n) \right] + \delta_k (w^* \theta^n) + \ldots \]

\[ \frac{\Delta t w^*}{\Delta Z} < 1 \]

\[ w^* = w - w_g \]

---

**A.L.E. (flavor 2)**

\[ \partial_z p = -g \rho (z, S^n, \theta^n) \]

\[ v_{h}^{n+1} = v_{h}^{n} + \Delta t \left( -\frac{1}{\rho_o} \nabla_z p + \ldots \right) \]

\[ h^{+} = h^n - \Delta t \nabla \cdot (h^n v_{h}^{+}) \]

\[ h_{n+1}^{+} \leftarrow \delta_k Z \left( z^{+} \right) \]

\[ \theta^{n+1} = \theta^{+} \left( Z(z^{+}) \right) \]

---

Leclair & Madec, 2011, use this form

Bleck, 2002

---

Hirt et al., 1974
Remapping styles: flux form small CFL

- Flux-form $\rightarrow$ conservative
- CFL<1 local stencil
- Stencil is "known"

E.g. The kernel

$$F_{k+\frac{1}{2}} = \max\left(0, \Delta t w_{k-\frac{1}{2}}\right) \theta_{k+1} + \min\left(0, \Delta t w_{k-\frac{1}{2}}\right) \theta_k$$

is vectorizable in horizontal

$$F[:, K] = \max(0, w[:, K]) \theta[:, k + 1] + \min(0, w[:, K]) \theta[:, k]$$
Remapping styles: flux form for arbitrary CFL

- Flux-form → conservative
- CFL>1 often evaluates residual difference of large numbers
  - Potentially inaccurate
- Stencil can be full column:

\[
F_{k + \frac{1}{2}} = \begin{cases} 
\sum_{k} h_l \theta_l + \Delta t w_{k - \frac{1}{2}} \theta_m & w_{k - \frac{1}{2}} > 0 \\
\sum_{k-1}^{m} h_l \theta_l + \Delta t w_{k - \frac{1}{2}} \theta_m & w_{k - \frac{1}{2}} < 0 
\end{cases}
\]
  
  - Nested loops – eek!
• Accuracy independent of CFL
  – No residual of differences
• Stencil can still be full column

\[ h\theta_{k+1}^{n+1} = \int_{z_{k-1/2}^{n+1}}^{z_{k+1/2}^{n+1}} \theta_{k+1}^\dagger dz \]

• Conservation is less accurate
  – No equal/opposite flux terms
How we implement remapping in MOM6

Evaluate integrals or reconstructions for sub-layers

N+M-1 sub-layers

Sum up sub-layers on to target layers

\[
\begin{align*}
    z^{n+1} = \sum_{k=1}^{N} \int_{z_k}^{z_{k+1}} \theta_k^{n+1} dz
\end{align*}
\]

- No nested loops but there are now extra branches

Hallberg and Adcroft, in prep.
Subcycling to minimize compute per step

\[
\delta_k p = -\rho(z, S^n, \theta^n) \delta_k \Phi
\]

\[
v_{h}^{m+1} = v_{h}^{m} + \frac{1}{M\rho_o}(-\nabla_r p - \rho \nabla_r \Phi + \ldots)
\]

\[
h^{m+1} = h^{m} - \frac{1}{M} \Delta t \nabla_r (h^{m} v_{h}^{m+1})
\]

\[
U^{l+1} = U^{l} + \frac{1}{L} \Delta t (-\nabla \eta^{l} + \ldots)
\]

\[
\eta^{l+1} = \eta^{m} - \frac{1}{L} \Delta t \nabla_r (H U^{l+1})
\]

\[
h^{n} \theta^{n} = \delta_k Z(Z^{*}) ; \quad \delta_k Z(Z^{*})
\]

\[
\theta^{n+1} = \theta^{*}(Z(Z^{*})) ; \ldots
\]
Non SIMD aspect of GC + column physics

- Column-to-column variations in source/target grid
- Some parameterizations have built-in non-linear non-local solvers
What about TPUs (ML)?

• All discretizations ultimately look like a [sparse] matrix multiply
  – Challenge for our algorithm is construction of the matrix in each column is very “branchy”
    • Different pattern of addressing in each column
    • Changes every step

• Could we delegate to ML?
  – I have not tried this but willing to entertain
  – However, ML is inexact
• Sea-level calculation:

\[ \eta = -D + \sum_k h_k \]
\[ \text{err}(\eta) \sim \varepsilon \ D \]

• \( \varepsilon_{16\text{-bit}} = 2^{-11}, \text{err}(\eta)_{16\text{-bit}} \sim 3.4 \text{ m} \)

• \( \varepsilon_{32\text{-bit}} = 2^{-24}, \text{err}(\eta)_{32\text{-bit}} \sim 4 \times 10^{-4} \text{ m} \ (0.4 \text{ mm}) \)

• \( \varepsilon_{64\text{-bit}} = 2^{-53}, \text{err}(\eta)_{64\text{-bit}} \sim 8 \times 10^{-13} \text{ m} \ (0.008 \text{ Å}) \)

• Context of evolving sea-level:
  – Sea-level rise \( \sim 0.3 \text{ m/Cy} \) or \( 1 \times 10^{-5} \text{ m/day} \)
  – Tides \( \sim 10 \text{ m/day} \)
• Algorithm developments have yielded great gains
  – Challenging to implement on existing SIMD tech
• New hardware still doesn’t address old problems
  – Barotropic solve likely to always be a performance barrier unless model fits on one chip
• Inexact math of ML is a reality
  – Potential advantages where stochasticity is appropriate
  – Major problem for algorithms that rely on consistency