MPAS-Atmosphere Mesh Generation

Michael G. Duda NSF NCAR/MMM





This material is based upon work supported by the NSF National Center for Atmospheric Research, a major facility sponsored by the U.S. National Science Foundation and managed by the University Corporation for Atmospheric Research. Any opinions, findings and conclusions or recommendations expressed in this material do not necessarily reflect the views of NSF.



Motivation

How do we generate the meshes that are a central feature of MPAS?





Both meshes in the preceding slide are (spherical) centroidal Voronoi tessellations:

- **Voronoi** = each grid volume (cell) V_i is uniquely associated with a *generating point* x_i such that all points within V_i are closer to x_i than to any other x_i
 - Lines joining generating points of adjacent cells are

MPAS

- 1. bisected by the shared cell face; and
- 2. intersect the shared cell face at a right angle.
- <u>Centroidal</u> = the generating point for each Voronoi cell is also the mass centroid of that cell (**w.r.t. some density function**)



Three Voronoi cells, V_i , along with their associated generating points, x_i . Each Voronoi cell corner is uniquely associated with a Delaunay triangle (dashed line).



Du et al.¹ show that the minimizer of a particular problem is necessarily a CVT

- It is reasonable, then, to approach the generation of CVTs as an optimization problem
- Various algorithms could be brought to bear, but we employ *Lloyd's Method*

For constant density in the plane, hexagons provide the minimum energy configuration of Voronoi regions (Newman², also Gersho³).



Originally developed by Stuart Lloyd at Bell Labs for optimal signal quantization

Given an initial set of generating points and a density function $\rho(\mathbf{x})$, Lloyd's method may be used to arrive at a CVT:



An example CVT (right) produced using Lloyd iteration, beginning from random points in the domain. From Du et al.¹



For a density function $\rho(\mathbf{x}) > 0$, it is conjectured (Ju et al.⁴) that, as the number of Voronoi cells increases, the diameters, *h*, of Voronoi cells are related by

$$\frac{h_i}{h_j} \approx \left(\frac{\rho(\mathbf{x}_j)}{\rho(\mathbf{x}_i)}\right)^{1/(d'+2)}$$

where d' is 2.

By defining an appropriate density function, therefore, we can distribute cells essentially however we choose!





A few notes on density functions

Functions with continuous first derivatives seem to work well; most of our atmosphere meshes use a density function based on Ringler et al.⁵ :

$$\rho(\mathbf{x}) = \frac{1-\gamma}{2} \left[\tanh\left(\frac{\beta - \|\mathbf{x}_c - \mathbf{x}\|}{\alpha}\right) + 1 \right] + \gamma$$

Here, x_c is the center of the refinement region, β is the half-width of the high-resolution region, α is the width of the "transition zone", and γ is the minimum density. x_c and x are constrained to lie on the surface of the sphere.



How small can our "transition" zone be if we want the average cell diameter to vary by at most a factor of r (with r > 1) from any cell to its neighbors? At least

$$\frac{\Delta x_{coarse}}{r-1} - \frac{\Delta x_{fine}}{r-1}$$



Density function controls the relative resolution in mesh, while the total number of generating points set the absolute mesh resolutions

• We use a simple Monte-Carlo method to estimate the required number of generating points





Resulting mean cell diameters from a 30-6 km mesh with the number of generating points estimated using the Monte-Carlo method



The obvious approach suggests simply defining the density function, generating the required number of generating points randomly, and iterating to convergence





With uniformly-distributed random initial generating points, there is apparently quite a lot of time spent in allowing points to migrate around the sphere

 This suggests we try to get cells into nearly their final position as "cheaply" as possible

Rather than choosing uniformly-distributed points, one might instead choose points based on the density function

 As a second attempt, draw initial points from a probability density function defined as ρ(x)^{0.5}



Choosing initial generating points according to the *p.d.f.* given by $\rho(\mathbf{x})^{0.5}$ does significantly reduce the generating point movement overall





The preceding approaches may be fine for coarse meshes, but convergence can still take quite some time for high-resolution meshes.

Todd Ringler (then at LANL) had suggested an approach that leads to meshes in significantly less time, and is what we now employ almost exclusively:

- 1. Begin with a random set $X_0 = \{x_i\}$, with $|X_0| = O(100)$, of initial generating points suitable for a mesh that is 2^n times coarser in resolution than the target mesh
- 2. For *k* = 0 ... *n*
 - 1. Perform Lloyd iterations on X_k to convergence to produce a mesh M_k
 - 2. If k < n, let X_{k+1} be the set of points resulting from the bisection of M_k
- 3. The mesh M_n is the a CVT with the desired resolution



Experience indicates that the incremental approach to grid generation does lead to a higher-quality mesh in significantly less wall-clock time





The incremental generation approach appears to be robust enough, but generating a mesh *is still very slow*

- Can take several months on a single multi-core desktop for highresolution meshes
- Lloyd iteration has two main computational phases, one of which is embarrassingly parallel





Unless we're parallelizing a part of the Lloyd iteration that takes considerable time, we'll not gain very much

• How much time is spent in the two main phases of *our implementation of* Lloyd's method?

Number of generating points	Centroid computation (seconds)	Voronoi tessellation (seconds)	Ratio of centroid time to Voronoi time		
100	0.02688	0.000138	194.8		"Toy" meshes with random initial
1000	0.24740	0.001441	171.7		
10000	2.462	0.01958	125.7		generating points
101378	3 25.37	0.1478	171.7		
405506	5 101.9	0.8024	127.0	F	Refinement stages from a 15- 3 km mesh
1622018	406.9	3.753	108.4		
6488066	5 1629	23.86	68.29		



With the addition of a single OpenMP parallel loop, we're able to gain a significant improvement in performance



From Amdahl's Law, we achieve an overall speedup of 11.3 on a 2x6-core Xeon desktop!



Parallelization of centroid computation with OpenMP is nearly trivial, but limits scaling to the number of cores in a shared-memory node

 Even with 24 threads on the desktop in my office, generating a 15-3 km mesh with ~6.5 M cells still took months!

Jacobsen et al.⁶ describe a more scalable approach with both phases of Lloyd's method parallelized with MPI

- Parallelize Voronoi tessellation using stereographic projections
- This approach seems better for larger meshes in particular



In contrast to variable-resolution meshes, quasi-uniform meshes are comparatively simple!

- Grid distance in original icosahedron = 7530 km (for earth radius 6371 km)
- Divide triangles by any integral factor, and let vertices of triangles be generating points after projection to the sphere



Bisecting an icosahedral mesh and projecting vertices to the sphere. Lipscomb and Ringler (2005)



For quasi-uniform MPAS meshes, we may divide by any positive integer factor.



Final remarks

Grid generation at present is a slow process that often requires human judgment in, e.g., knowing when a stage of mesh generation is converged "enough", working around stable but undesirable mesh features, etc.

- Before this is supportable to the community, we need to make mesh generation more user-friendly
 - Integrated grid metrics are needed to tell us:
 - When to stop iterating
 - Whether we can expect good numerical properties from resulting mesh
- Improvements to performance are still needed
 - Can techniques like over-relaxation be applied to improve convergence rate?
 - Implement code optimizations, exploit GPUs for embarrassingly parallel work, etc. to improve iteration rate





References

¹ Du, Q., V. Faber, and M. Gunzburger, 1999. "Centroidal Voronoi Tessellations: Applications and Algorithms". *SIAM Review*, **41**, 637 – 676.

² Newman, D. J., 1982. "The Hexagon Theorem". *IEEE Transactions on Information Theory*, **28**, 137 – 139.

³ Gersho, A., 1979. "Asymptotically Optimal Block Quantization". *IEEE Transactions on Information Theory*, **25**, 373 – 380.

⁴ "Voronoi Diagrams and their Application in Climate and Global Modeling", Chapter in *Numerical Techniques for Global Atmospheric Models*. Springer-Verlag Berlin Heidelberg, 2011.

⁵ Ringler, T. D., D. Jacobsen, M. Gunzburger, L. Ju, M. Duda, and W. Skamarock, 2011. "Exploring a Multiresolution Modeling Approach within the Shallow-Water Equations". *Mon. Wea. Rev.*, **139**, 3348 – 3368.

⁶ Jacobsen, D.W., M. Gunzburger, T. Ringler, J. Burkardt, and J. Peterson, 2013. "Parallel algorithms for planar and spherical Delaunay construction with an application to centroidal Voronoi tessellations". *Geosci. Model Dev.*, **6**, 1353 – 1365.