Lecture 18: What Are Good Meshes?

Xiangmin Jiao

Stony Brook University
Types of Meshes

- Structured and unstructured meshes
  - Variants of structured meshes: curvilinear meshes, block structured meshes, overset meshes

- Simplicial meshes: edges in 1-D, triangles in 2-D, tetrahedra in 3-D
- Tensor-product-element meshes: quadrilaterals and hexahedra
- Mixed-element meshes: mixed triangles and quadrilaterals in 2-D; tets, prisms, pyramids, hexes in 3-D
- High-order meshes: quadratic/cubic/quartic/spectral elements
Accuracy & Stability Considerations: Element Shapes

- **What are good meshes?**
  - This is most basic and yet still open problem in meshing

- **Element shapes**
  - For simplices: *angle conditions* (or in simplified sense, *aspect ratios*)
    - *Maximum angle* should be minimized for derivative errors
    - *Minimum angle* should be maximized for stability (rarely a factor)
  - For tensor-product elements
    - *Aspect ratio* is important for derivative errors (analogous to *angle condition* of simplicial meshes)
    - *Skewness* should be bounded for completeness of polynomial basis
    - *Convexity* must be guaranteed to avoid zero or negative Jacobian
Element sizes

- Quasiuniformity in terms of element sizes, or more tightly, 1-ring neighborhood sizes, is necessary condition for convergence
- Longest edge lengths determine $h$ in error estimation
- Mesh grading is needed for anisotropic problems and for solutions with large gradients to equalize error distribution

Longest edge lengths and $d$th min-angle determine error in $q$th derivative with degree-$p$ polynomial: $\|D^q u\|_\infty \mathcal{O}(\cot^q \theta_d h^{p-q+1})$, so cannot be considered separately (unpublished result)
Accuracy & Stability Considerations: Symmetry

- For quadratic elements: symmetry (centroidality) is important for error cancellation and superconvergence

A centroidal quadratic mesh.  
An irregular quadratic mesh.

- Symmetry/centroidality is local property of 1-ring neighborhoods, but it determines global structure (at least away from boundaries)
Accuracy & Stability Considerations: Curved Boundaries

For curved boundaries the matter is even more complicated

- Accuracy of positions and normals at quadrature points of surface elements are essential to preserve convergence for Neumann boundaries, especially on nonelliptic (nonconvex) boundaries
- In addition, curved boundaries in general degrade superconvergence
  - Quadratic meshes lose symmetry near boundaries (even flat)
  - Spectral elements lose some superconvergence due to loss of degree of quadrature, which is due to loss of orthogonality of polynomials
- Bottom line:
  - High-order accurate boundaries are necessary but insufficient to ensure uniform high-order convergence
  - Higher degree basis are required along/near boundaries to achieve uniform high-order convergence, especially for superconvergent methods
  - Exact geometry is neither necessary nor sufficient
Mesh Quality Considerations: Efficiency

- Number of elements vs. number of nodes
  - *Degree of freedom* is proportional to *number of nodes*, so it is always a good idea to minimize it (assuming same accuracy can be achieved)
  - *Cost of assembly* is proportional to *number of elements* for unstructured meshes, so it is a good idea to minimize number of elements (assuming constant cost per element)

- However, this is more subtle with structured/semi-structured meshes, for which cost of assembly is independent of number of elements

- Therefore, minimizing number of elements may be good for given type of mesh, but it is misleading when comparing different types of meshes

- Most importantly, impact on linear solvers: should enable geometric multigrid methods for elliptic problems, which are more than one order of magnitude faster than AMG and Krylov-subspace methods
Other Considerations

- **Compatibility with solvers**
  - Many FEM solvers only support particular types of elements
  - Conformal vs. non-conformal meshes
    - Definition of conformal mesh: two elements intersect at subelement (node, edge, face, etc.)
    - Non-conformal meshes are in general undesirable, especially for superconvergent methods, because interpolation will degrade superconvergence

- **Parallelization of mesh generation to support large-scale simulations**
  - Shared-memory implementation is typically sufficient for efficiency
  - Distributed-memory is needed mostly only for compatibility with solvers