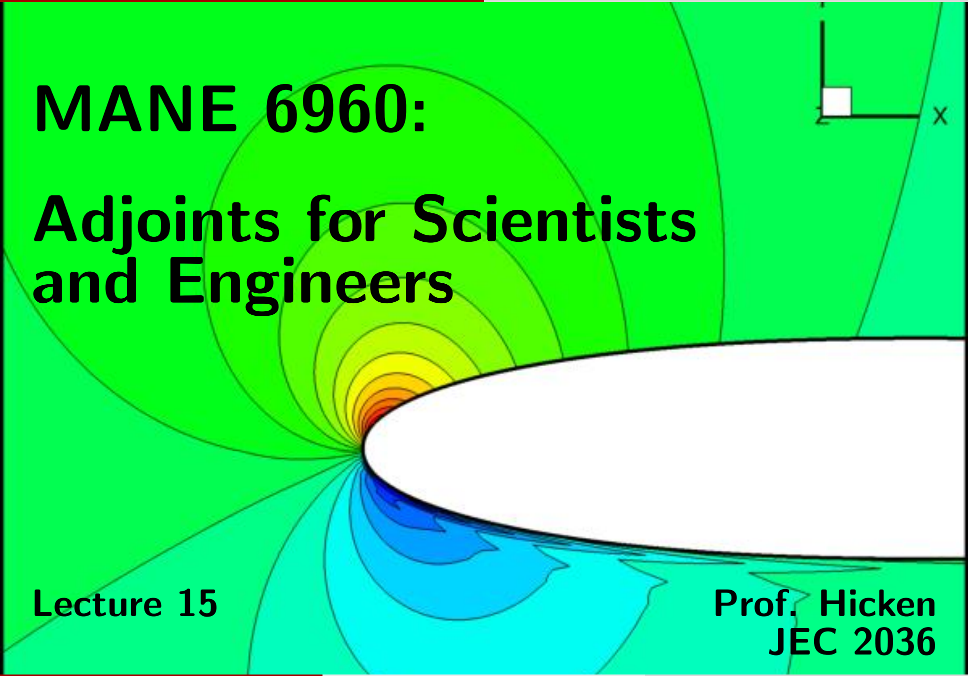


MANE 6960:

Adjoints for Scientists and Engineers

Lecture 15

Prof. Hicken
JEC 2036

Error Effectivity

Error Effectivity

Recall that the adjoint-weighted residual (AWR) method provides only an approximation to the true error:

$$\begin{aligned} J_H(u_H) - J(u) &\approx J_H(u_H) - J_h(u_h) \\ &= \bar{\psi}_h^T R_h(u_h^H) \\ &\approx \tilde{\psi}_h^T R_h(u_h^H). \end{aligned}$$

- Approximate $J(u)$ with $J_h(u_h)$
- Approximate the mean-value adjoint, $\bar{\psi}_h$
- Approximate the fine-space adjoint

Error Effectivity (cont.)

In light of the approximations made, the literature often uses the error effectivity to quantify the quality of the error estimate.

Definition: Error Effectivity [FD11]

The (output) error effectivity is

$$\eta_H^e \equiv \frac{J_H(u_H) - J_h(u_h)}{J_H(u_H) - J(u)}.$$

- Alternatively, $\eta_H^e = \tilde{\psi}_h^T R_h(u_h^H)/(J_H(u_h) - J(u))$
- Ideally, we would like $\eta_H^e = 1$, but this may not happen, even without the approximations in the AWR.

Error Effectivity (cont.)

If the we use simple element subdivision for the enriched space with, say, $h = H/2$, and an order q scheme then

$$J_H(u_H) - J(u) = CH^q$$

and $$J_H(u_H) - J_h(u_h) = CH^q - C(H/2)^q,$$

Consequently,

$$\eta_H^e = 1 - (1/2)^q,$$

which does not converge to 1 with refinement.

- However, if the convergence rate is know a priori, this can be taken into account in the error estimate.

Error Effectivity (cont.)

On the other hand, if we use p enrichment for the refined space with, say, an order $r > q$ method, then

$$J_H(u_H) - J_h(u_h) = C_q H^q - C_r H^r = C_q H^q (1 - \alpha H^{r-q})$$

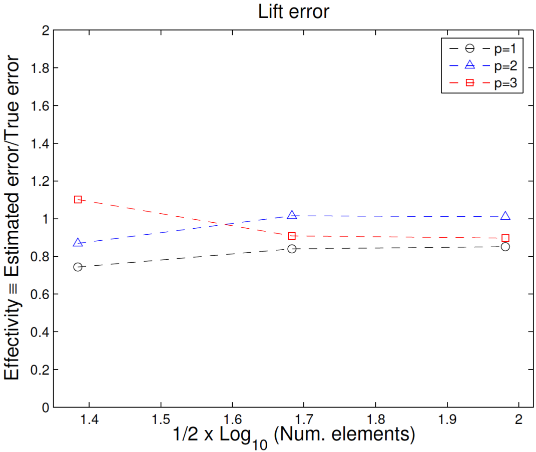
where $\alpha = C_r / C_q$. Thus, in this case

$$\eta_H^e = 1 - \alpha H^{(r-q)},$$

which does tend to ~~zero~~ one with refinement.

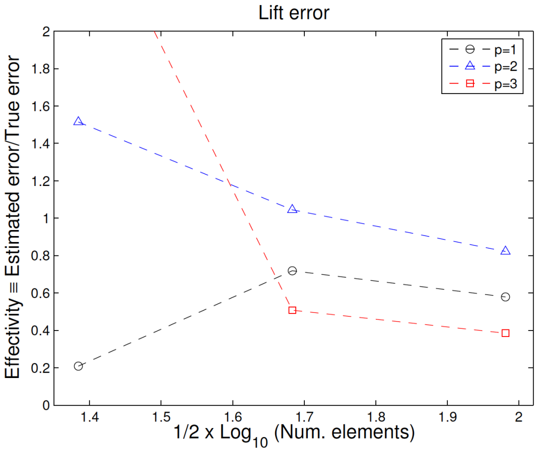
Error Effectivity (cont.)

η_H^e for lift functional (adjoint consistent discretization) [Lu05]



Error Effectivity (cont.)

η_H^e for lift functional (adjoint inconsistent discretization) [Lu05]



Output-based Mesh Adaptation: Error Localization

Overview

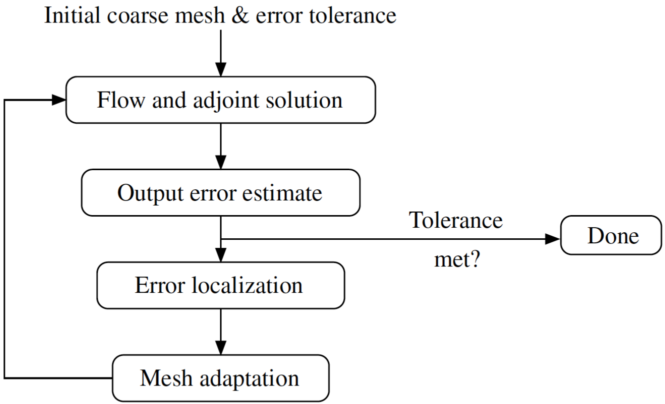
Once we have an estimate of the output error, and this error is larger than a desired tolerance, the next logical step is to consider adapting the mesh to make the error estimate fall below the tolerance.

In general, the process of estimating the error and adapting the mesh must be performed iteratively.

- The major components of this process are illustrated on the flowchart on the next slide.
- The rest of this chapter will describe the last two components: error localization and mesh adaptation.

Overview (cont.)

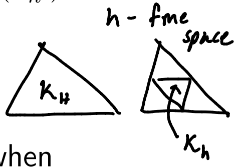
Generic output-based mesh adaptation process [FD11]



Error Localization: DG and FV

Suppose we are using either a finite-volume (FV) or Discontinuous-Galerkin (DG) finite-element method. With these methods, the AWR output-error estimate can be written as a sum of contributions over elements/cells [FD11]:

$$J_H(u_H) - J_h(u_h) \approx \sum_{\kappa_H \in T_H} \sum_{\kappa_h \in \kappa_H} \tilde{\psi}_h^T \Big|_{\kappa_h} R_h(u_h^H).$$



- T_H is the set of coarse elements
- κ_h denotes a refined element embedded in κ_H; when p-enrichment is used the actual coarse and fine elements may coincide, i.e. $\kappa_h = \kappa_H$.

Error Localization: DG and FV (cont.)

A popular and straightforward way to localize the AWR estimate is to take the absolute value of each element contribution. Thus, the error associated with element κ_H is given by

$$\epsilon_{\kappa_H} = \left| \sum_{\kappa_h \in \kappa_H} \tilde{\psi}_h^T \Big|_{\kappa_h} R_h(u_h^H) \right|.$$

- For systems of equations (e.g. Euler, NS), practicitioners usually compute a separate error for each equation, and then sum the result.

Error Localization: DG and FV (cont.)

Finally, we note that the local errors do not provide a bound on the error. That is, even though

$$\left|\tilde{\psi}_h^T R_h(u_h^H)\right| \leq \sum_{\kappa_H \in T_H} \epsilon_{\kappa_H},$$

we do not necessarily have

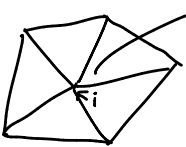
$$|J_H(u_H) - J_h(u_h)| \not\leq \sum_{\kappa_H \in T_H} \epsilon_{\kappa_H}.$$

- This is because of the various approximations made in the practical implementation of the AWR method.
- However, with refinement, the sum of error indicators does tend toward $|J_H(u_H) - J_h(u_h)|$.

Error Localization: CG

The DG and FV methods can use the above localization, because their weighted residuals vanish on each element independently.

The same is not true for continuous Galerkin (CG) finite-element methods, since the test function space provides contributions over several elements.



ψ_i is nonzero over several elements

Error Localization: CG (cont.)

To see this, consider the Galerkin FEM residual for a Poisson PDE, ignoring boundary conditions:

$$v_H^T R_H(u_H) = \sum_{K_H \in T_H} \int_{K_H} (\nabla v_H) \cdot (\nabla u_H) \, d\Omega = 0, \quad \forall v_H \in V_H$$

The terms being summed are not zero independently. (only the sum is zero)

Error Localization: CG (cont.)

This issue can be corrected by expressing the residual in strong form:

$$
\begin{aligned}
v_H^T\, R_H(u_H) &= \sum_{\kappa_H \in T_H} \int_{\kappa_H} \nabla \cdot (v_H \nabla u_H)\, d\Omega - \int_{\kappa_H} \nabla^2 u_H \, d\Omega \\
&= \sum_{\kappa_H} \frac{1}{2} \int_{\partial \kappa_H} v_H [\![n \cdot \nabla u_H]\!] \, d\Gamma - \int_{\kappa_H} v_H \nabla^2 u_H \, d\Omega,
\end{aligned}
$$

where $[\![n \cdot \nabla u_H]\!]$ denotes the jump in the normal derivative across the element interface.

- The factor of $1/2$ on the element boundary integral arises because the surface integral is partitioned between adjacent elements; this is done to isolate the term to the element.

Error Localization: CG (cont.)

Written in strong form, with the face-integrals partitioned between elements, the AWR error estimate can now be localized as follows:

$$\epsilon_{\kappa_H} \equiv \left| \sum_{\kappa_h \in \kappa_H} \tilde{\psi}_h^T \Big|_{\kappa_h} R_h(u_h^H) + \frac{1}{2} \, \tilde{\psi}_h^T \Big|_{\partial \kappa_h} R_{h,\partial\kappa}(u_h^H) \right|,$$

where $R_{h,\partial\kappa}$ denotes the interface jump terms on the fine space/mesh.

- Similar care must be taken with advection-stablization methods for CG such as SUPG and GLS.

h-Adaptation Methods

Overview of Mesh Adaptation Methods

Once we have localized the error over the elements, we can consider refining or coarsening the mesh, i.e. we can adapt the mesh. There are three general classes of mesh refinement:

h **refinement:** change the elements sizes, for example, by subdivision.

p **refinement:** maintain the element sizes, but change the order of the polynomial basis in each element.

r **refinement:** maintain the mesh topology by repositioning the vertices.

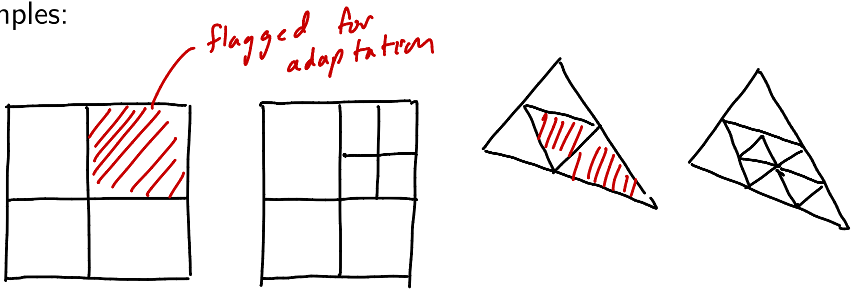
These three approaches are not mutually exclusive. For example, hp refinement is an increasingly popular method.

In this lecture, we will limit the discussion to h refinement, which is the most common approach.

Element subdivision

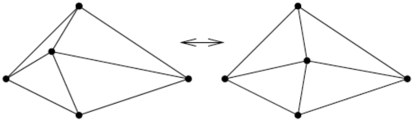
If the discretization and mesh support non-conforming elements, also know as hanging nodes, then we can use simple element-based subdivision to refine the mesh.

Examples:

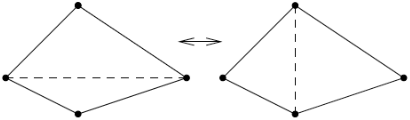


Simplex-mesh Operations

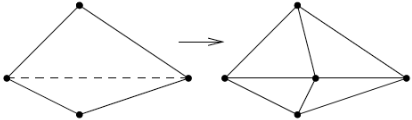
Simplex meshes, e.g. triangular and tetrahedral meshes, can be adapted using a set of simple local operations. These same operations are frequently used during mesh generation.



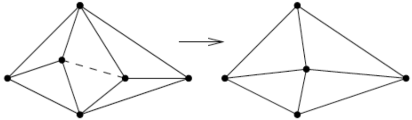
node movement



triangle swap



triangle split



triangle collapse

Global Remeshing

Another possibility, which we will discuss further next class, is to globally regenerate the mesh.

- The idea is to provide the mesh generator with target element sizes and orientations
- The target sizes are determined by the localized errors, which the target directions are often determined based on the Hessian of the solution (for $p = 1$ discretizations.)

Output-based Mesh Adaptation Strategies

What is the Goal?

The goal of mesh adaptation is to obtain the mesh that uses the fewest degrees of freedom to achieve a desired error tolerance, whether the error is in the solution or some output of interest.

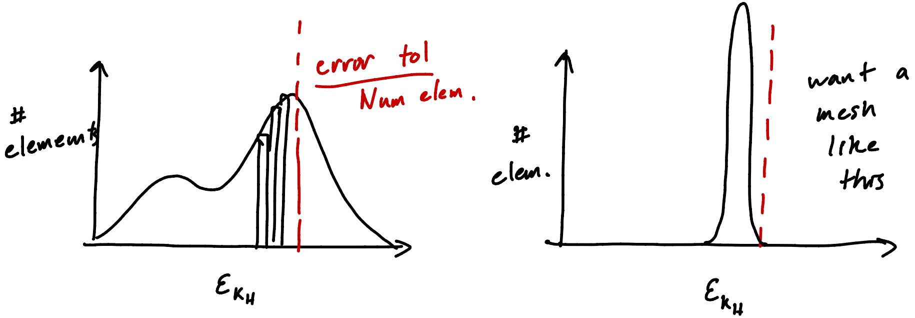
Furthermore, the final mesh should be obtained using as little computational effort as possible.

- If the cost of finding the "best" mesh is larger than the cost of running on a uniformly refined mesh that produces the same error, then the adaptation process is inefficient.

Equidistribution of Error

A ubiquitous assumption in the mesh adaptation literature is that all elements should carry approximately the same contribution to the total error.

- That is, the error should be equidistributed.
- For the AWR method, this means ϵ_{κ_H} is equal for all κ_H.



Constant Threshold

The simplest strategy for output-based mesh adaptation is to set a constant threshold ϵ_{targ}, and refine all elements that have $\epsilon_{\kappa_H} > \epsilon_{\text{targ}}$.

- While easy to implement, this can lead to over refinement, and therefore costly intermediate iterations.

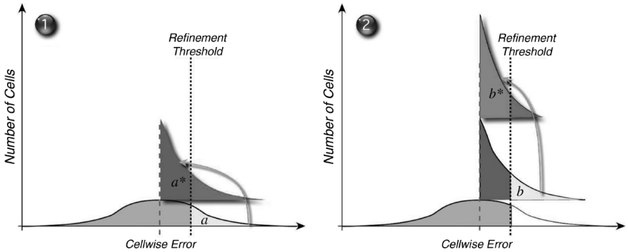


Figure from [NAW08]

Decreasing Threshold

A popular alternative to using a constant threshold is to gradually decrease the threshold.

- This ensures elements with the largest errors are refined first.
- Consequently, the mesh grows more slowly, leading to fewer expensive solves on the finest meshes.

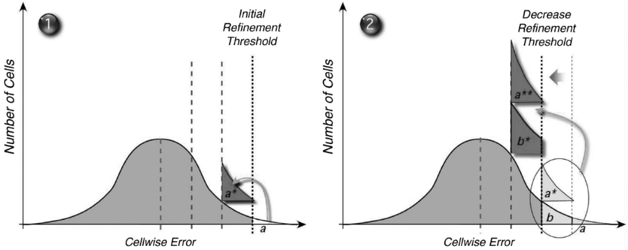


Figure from [NAW08]

Note on Global Remeshing

The above thresholding methods are important when element subdivision is used for refinement, e.g. Cartesian grids with hanging nodes.

However, when the adaptation mechanics can vary from element to element, e.g. in global remeshing or simplex-based operations, other strategies are possible.

- In particular, with global remeshing obviously all the elements are obviously changed, in general.

References

[FD11] Krzysztof J. Fidkowski and David L. Darmofal, *Review of output-based error estimation and mesh adaptation in computational fluid dynamics*, AIAA Journal **49** (2011), no. 4, 673–694.

[Lu05] James C. Lu, *An a posteriori error control framework for adaptive precision optimization using discontinuous Galerkin finite element method*, Ph.D. thesis, Massachusetts Institute of Technology, Cambridge, Massachusetts, 2005.

[NAW08] Marian Nemec, Michael J. Aftosmis, and Mathias Wintzer, *Adjoint-based adaptive mesh refinement for complex geometries*, The 46th AIAA Aerospace Sciences Meeting and Exhibit (Reno, Nevada, United States), no. AIAA–2008–725, January 2008.