Numerical solution of hydrodynamic problems on adaptive unstructured meshes

Yuri Vassilevski

Institute of Numerical Mathematics
Russian Academy of Sciences, Moscow

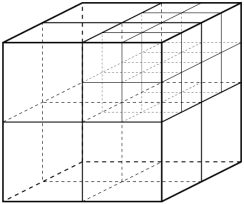
Summer Sumerpcomputing Academy
MSU June 25, 2015

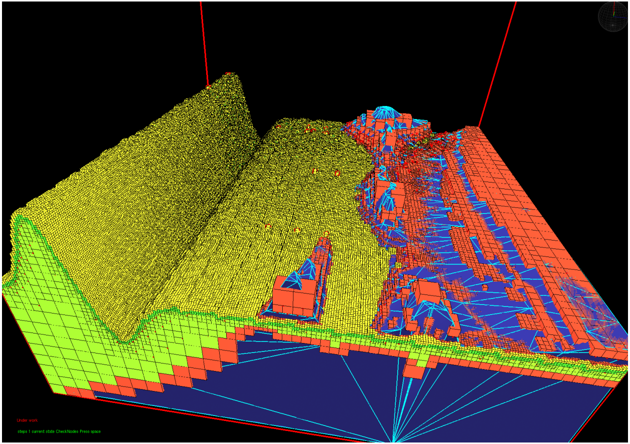
Variety of unstructured meshes

Tetrahedral meshes

Variety of unstructured meshes

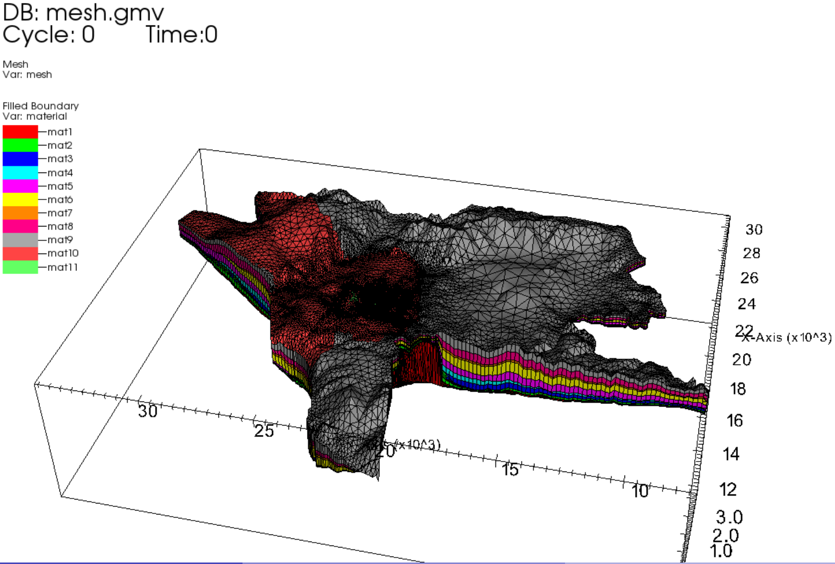
Octree meshes





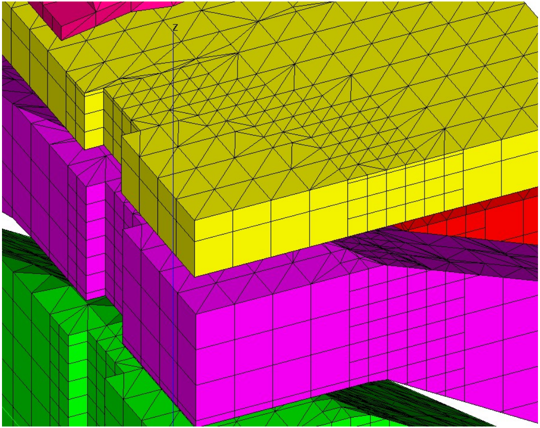
Variety of unstructured meshes

Triangular prismatic meshes



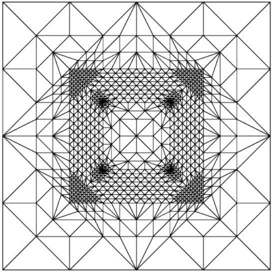
Variety of unstructured meshes

Meshes with cut-cells (polyhedral meshes)



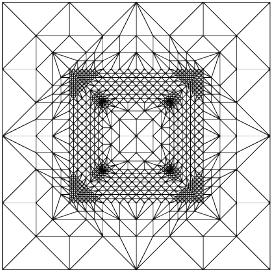
Types of mesh adaptation

Types of mesh adaptation

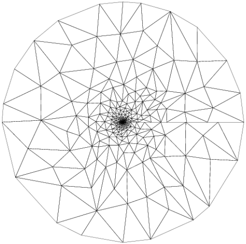


hierarchical

Types of mesh adaptation

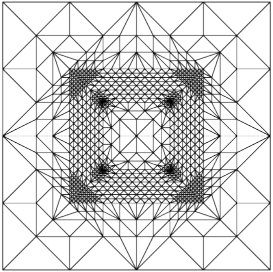


hierarchical

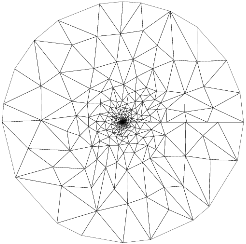


regular

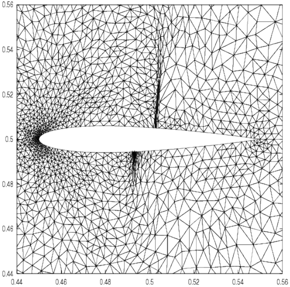
Types of mesh adaptation



hierarchical



regular



anisotropic

Metric-based control of simplicial meshes

$$\mathfrak{M} = \mathfrak{M}^T > 0 \quad \mathfrak{M} = \begin{bmatrix} M_{xx} & M_{xy} \\ M_{yx} & M_{yy} \end{bmatrix}, \quad \mathfrak{M} = \begin{bmatrix} M_{xx} & M_{xy} & M_{xz} \\ M_{yx} & M_{yy} & M_{yz} \\ M_{zx} & M_{zy} & M_{zz} \end{bmatrix}$$

- Area/volume of domain D:

$$|D|_{\mathfrak{M}} = \int_D \sqrt{\det(\mathfrak{M}(\mathbf{x}))}\,\mathrm{d}V \approx |D|\sqrt{\det(\mathfrak{M}(\mathbf{x}_*))}$$

- Length of parameterized curve ℓ:

$$|\ell|_{\mathfrak{M}} = \int_0^1 \sqrt{\gamma'(t)^T \mathfrak{M}(\gamma(t))\gamma'(t)}\,\mathrm{d}t$$

Length of parameterized edge $\mathbf{x} = \mathbf{x}_1 + t(\mathbf{x}_2 - \mathbf{x}_1)$:

$$|\mathbf{e}|_{\mathfrak{M}} = \int_0^1 \sqrt{(\mathbf{x}_2 - \mathbf{x}_1)^T \mathfrak{M}(\gamma(t))(\mathbf{x}_2 - \mathbf{x}_1)}\,\mathrm{d}t \approx \sqrt{(\mathbf{x}_1 - \mathbf{x}_2)^T \mathfrak{M}(\mathbf{x}_{12})(\mathbf{x}_1 - \mathbf{x}_2)}$$

- "Perimeter" of triangle/tetrahedron: $p_{\mathfrak{M}}(\Delta) = \sum_{k=1}^{n_{edges}} |\mathbf{e}_k|_{\mathfrak{M}}$

Metric-based control of simplicial meshes

$$\mathfrak{M} = \mathfrak{M}^T > 0 \quad \mathfrak{M} = \begin{bmatrix} M_{xx} & M_{xy} \\ M_{yx} & M_{yy} \end{bmatrix}, \quad \mathfrak{M} = \begin{bmatrix} M_{xx} & M_{xy} & M_{xz} \\ M_{yx} & M_{yy} & M_{yz} \\ M_{zx} & M_{zy} & M_{zz} \end{bmatrix}$$

- Area/volume of domain D:

$$|D|_{\mathfrak{M}} = \int_D \sqrt{\det(\mathfrak{M}(\mathbf{x}))}\, \mathrm{d}V \approx |D| \sqrt{\det(\mathfrak{M}(\mathbf{x}_*))}$$

- Length of parameterized curve ℓ:

$$|\ell|_{\mathfrak{M}} = \int_0^1 \sqrt{\gamma'(t)^T \mathfrak{M}(\gamma(t))\gamma'(t)}\, \mathrm{d}t$$

Length of parameterized edge $\mathbf{x} = \mathbf{x}_1 + t(\mathbf{x}_2 - \mathbf{x}_1)$:

$$|\mathbf{e}|_{\mathfrak{M}} = \int_0^1 \sqrt{(\mathbf{x}_2 - \mathbf{x}_1)^T \mathfrak{M}(\gamma(t))(\mathbf{x}_2 - \mathbf{x}_1)}\, \mathrm{d}t \approx \sqrt{(\mathbf{x}_1 - \mathbf{x}_2)^T \mathfrak{M}(\mathbf{x}_{12})(\mathbf{x}_1 - \mathbf{x}_2)}$$

- "Perimeter" of triangle/tetrahedron: $p_{\mathfrak{M}}(\Delta) = \sum_{k=1}^{n_{edges}} |\mathbf{e}_k|_{\mathfrak{M}}$

Metric-based control of simplicial meshes

$$\mathfrak{M} = \mathfrak{M}^T > 0 \quad \mathfrak{M} = \begin{bmatrix} M_{xx} & M_{xy} \\ M_{yx} & M_{yy} \end{bmatrix}, \quad \mathfrak{M} = \begin{bmatrix} M_{xx} & M_{xy} & M_{xz} \\ M_{yx} & M_{yy} & M_{yz} \\ M_{zx} & M_{zy} & M_{zz} \end{bmatrix}$$

- Area/volume of domain D:

$$|D|_{\mathfrak{M}} = \int_D \sqrt{\det(\mathfrak{M}(\mathbf{x}))}\,\mathrm{d}V \approx |D|\sqrt{\det(\mathfrak{M}(\mathbf{x}_*))}$$

- Length of parameterized curve ℓ:

$$|\ell|_{\mathfrak{M}} = \int_0^1 \sqrt{\gamma'(t)^T \mathfrak{M}(\gamma(t))\gamma'(t)}\,\mathrm{d}t$$

Length of parameterized edge $\mathbf{x} = \mathbf{x}_1 + t(\mathbf{x}_2 - \mathbf{x}_1)$:

$$|\mathbf{e}|_{\mathfrak{M}} = \int_0^1 \sqrt{(\mathbf{x}_2 - \mathbf{x}_1)^T \mathfrak{M}(\gamma(t))(\mathbf{x}_2 - \mathbf{x}_1)}\,\mathrm{d}t \approx \sqrt{(\mathbf{x}_1 - \mathbf{x}_2)^T \mathfrak{M}(\mathbf{x}_{12})(\mathbf{x}_1 - \mathbf{x}_2)}$$

- "Perimeter" of triangle/tetrahedron: $p_{\mathfrak{M}}(\Delta) = \sum_{k=1}^{n_{edges}} |\mathbf{e}_k|_{\mathfrak{M}}$

Metric-based control of simplicial meshes

$$\mathfrak{M} = \mathfrak{M}^T > 0 \quad \mathfrak{M} = \begin{bmatrix} M_{xx} & M_{xy} \\ M_{yx} & M_{yy} \end{bmatrix}, \quad \mathfrak{M} = \begin{bmatrix} M_{xx} & M_{xy} & M_{xz} \\ M_{yx} & M_{yy} & M_{yz} \\ M_{zx} & M_{zy} & M_{zz} \end{bmatrix}$$

- Area/volume of domain D:

$$|D|_{\mathfrak{M}} = \int_D \sqrt{\det(\mathfrak{M}(\mathbf{x}))}\,\mathrm{d}V \approx |D|\sqrt{\det(\mathfrak{M}(\mathbf{x}_*))}$$

- Length of parameterized curve ℓ:

$$|\ell|_{\mathfrak{M}} = \int_0^1 \sqrt{\gamma'(t)^T \mathfrak{M}(\gamma(t))\gamma'(t)}\,\mathrm{d}t$$

  Length of parameterized edge $\mathbf{x} = \mathbf{x}_1 + t(\mathbf{x}_2 - \mathbf{x}_1)$:

$$|\mathbf{e}|_{\mathfrak{M}} = \int_0^1 \sqrt{(\mathbf{x}_2 - \mathbf{x}_1)^T \mathfrak{M}(\gamma(t))(\mathbf{x}_2 - \mathbf{x}_1)}\,\mathrm{d}t \approx \sqrt{(\mathbf{x}_1 - \mathbf{x}_2)^T \mathfrak{M}(\mathbf{x}_{12})(\mathbf{x}_1 - \mathbf{x}_2)}$$

- "Perimeter" of triangle/tetrahedron: $p_{\mathfrak{M}}(\Delta) = \sum_{k=1}^{n_{edges}} |\mathbf{e}_k|_{\mathfrak{M}}$

Mesh shape quality

Quality of triangle Δ in metric $\mathfrak{M}$:

$$Q_{\mathfrak{M}}(\Delta) = 12\sqrt{3}\frac{|\Delta|_{\mathfrak{M}}}{p_{\mathfrak{M}}(\Delta)^2}$$

Mesh shape quality:

$$Q_{\mathfrak{M}}(\Omega_h) = \min_{\Delta \in \Omega_h} Q_{\mathfrak{M}}(\Delta)$$

Mesh shape quality

Quality of tetrahedron Δ in metric $\mathfrak{M}$:

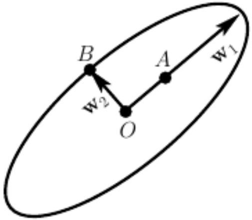
$$Q_{\mathfrak{M}}(\Delta) = 6^4\sqrt{2}\frac{|\Delta|_{\mathfrak{M}}}{p_{\mathfrak{M}}(\Delta)^3}$$

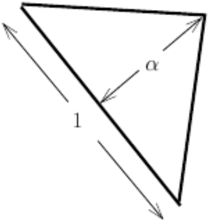
Mesh shape quality:

$$Q_{\mathfrak{M}}(\Omega_h) = \min_{\Delta \in \Omega_h} Q_{\mathfrak{M}}(\Delta)$$

Example

$$\mathfrak{M} = \begin{bmatrix} 5.89 & 2.51 \\ 2.51 & 5.01 \end{bmatrix}, \quad \lambda_1 = 8.0, \quad \lambda_2 = 2.9$$





$\mathfrak{M}$-equilateral triangle

height $\alpha = \sqrt{3\lambda_2/4\lambda_1}$

Control of mesh properties

Given tensor metric field $\mathfrak{M}(\mathbf{x})$ and desirable number of cells $N_\star$, we generate by a sequence of local modifications a $\mathfrak{M}$-quasiuniform mesh with $N_\star$ cells.

- $h_\star$ is a mesh size of $\mathfrak{M}$-quasiuniform mesh with $N_\star$ cells:

$$h_\star = \left(\frac{1}{N_\star\, V_d} \int_\Omega \sqrt{\det(\mathfrak{M}(\mathbf{x}))} dV \right)^{1/d}.$$

- $F(\cdot)$ is a smooth positive function with the only maximum $F(1) = 1$

Mesh quality:

$$Q(\Omega_h) = \min_{\Delta \in \Omega_h} Q(\Delta)$$

Monotone increase of $Q(\Omega_h)$ by a set of local modifications

Control of mesh properties

Quality of triangle Δ in metric $\mathfrak{M}$:

shape **size**

$$Q_{\mathfrak{M},N_\star}(\Delta) = 12\sqrt{3}\,\frac{|\Delta|_{\mathfrak{M}}}{p_{\mathfrak{M}}(\Delta)^2} F\left(\frac{p_{\mathfrak{M}}(\Delta)}{3h_\star}\right)$$

- $h_\star$ is a mesh size of $\mathfrak{M}$-quasiuniform mesh with $N_\star$ cells:

$$h_\star = \left(\frac{1}{N_\star\, V_d}\int_\Omega \sqrt{\det(\mathfrak{M}(\mathbf{x}))}dV\right)^{1/d}.$$

- $F(\cdot)$ is a smooth positive function with the only maximum $F(1) = 1$

Mesh quality:

$$Q(\Omega_h) = \min_{\Delta\in\Omega_h} Q(\Delta)$$

Monotone increase of $Q(\Omega_h)$ by a set of local modifications

Control of mesh properties

Quality of tetrahedron Δ in metric $\mathfrak{M}$:

$$Q_{\mathfrak{M},N_\star}(\Delta) = 6^4\sqrt{2}\,\underbrace{\frac{|\Delta|_{\mathfrak{M}}}{p_{\mathfrak{M}}(\Delta)^3}}_{\text{shape}}\,\underbrace{F\left(\frac{p_{\mathfrak{M}}(\Delta)}{6h_\star}\right)}_{\text{size}}$$

- $h_\star$ is a mesh size of $\mathfrak{M}$-quasiuniform mesh with $N_\star$ cells:

$$h_\star = \left(\frac{1}{N_\star V_d}\int_\Omega \sqrt{\det(\mathfrak{M}(\mathbf{x}))}dV\right)^{1/d}.$$

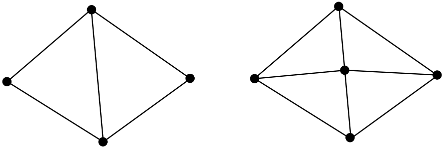
- $F(\cdot)$ is a smooth positive function with the only maximum $F(1) = 1$

Mesh quality:

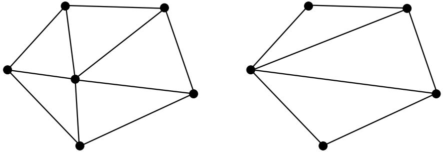
$$Q(\Omega_h) = \min_{\Delta\in\Omega_h} Q(\Delta)$$

Monotone increase of $Q(\Omega_h)$ by a set of local modifications

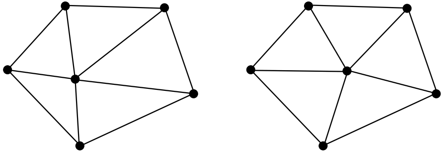
Local topological operations in 2D



Op 1: point insertion

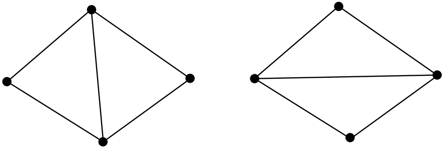


Op 2: point deletion

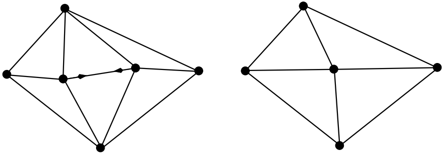


Op 3: point relocation

Local topological operations in 2D



Op 4: edge swap

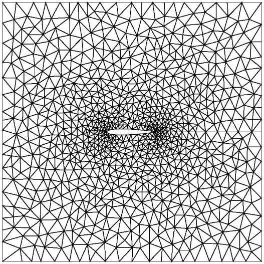


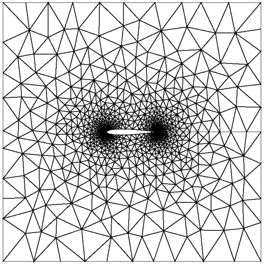
Op 5: edge collapsing

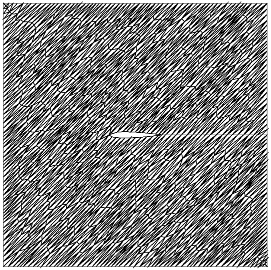
Locality is the key to robustness,
rich set of operations provides faster convergence

Examples of mesh control

Choice of $\mathfrak{M}$







$$h(\mathbf{x})^{-2}\mathbb{I}_2$$

$$\max\{|(x-0.4)^2+(y-0.5)^2+10^{-4}|^{a/2},\\ |(x-0.6)^2+(y-0.5)^2+10^{-4}|^{a/2}\}$$

$$\mathbb{R}_{\pi/4}\begin{bmatrix} 100 & 0 \\ 0 & 1 \end{bmatrix}\mathbb{R}^T_{\pi/4}$$

Example of metric-based adaptation loop

Initialization Step. Generate an initial triangulation Ω^h. Choose the final mesh quality Q_0, $Q_0 < 1$, and the final number $N_\star$ of mesh elements.

Iterative Step.

1. Compute the discrete solution $\mathcal{P}_{\Omega^h} u$ for triangulation Ω^h.
2. Recover the tensor metric field $\mathfrak{M}$ from $\mathcal{P}_{\Omega^h} u$. Stop iterations if $Q_{\mathfrak{M},N_\star}(\Omega^h) \geq Q_0$.
3. Generate the next mesh $\widetilde{\Omega}^h$ such that $Q_{\mathfrak{M},N_\star}(\widetilde{\Omega}^h) \geq Q_0$.
4. Set $\Omega^h = \widetilde{\Omega}^h$ and go to 1.

Approaches to recovery of $\mathfrak{M}$

- Recover discrete Hessian
  - + black-box
  - — lack of analysis, error control
- Use a posteriori error estimates
  - — problem dependent
  - + theory and error estimates exist

Approaches to recovery of $\mathfrak{M}$

- Recover discrete Hessian
  + black-box
  — lack of analysis, error control
- Use a posteriori error estimates
  — problem dependent
  + theory and error estimates exist

Approaches to recovery of $\mathfrak{M}$

- Recover discrete Hessian
  - $+$ black-box
  - $-$ lack of analysis, error control
- Use a posteriori error estimates
  - $-$ problem dependent
  - $+$ theory and error estimates exist

Metric and Hessian

If the adaptation goal is to minimize P_1-interpolation error

$$\|u - \mathcal{P}_{\Omega^h} u\|_{L_p(\Omega)}, \qquad 0 < p \leq \infty,$$

take

$$\mathfrak{M}(x) = (det|H(x)|)^{-1/(2p+2)}|H(x)|$$

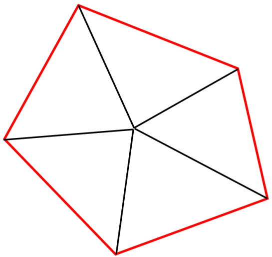
where H is the Hessian of u.

- $|H| = W^T|\Lambda|W$ from local spectral decomposition $H = W^T\Lambda W$.
- u is unknown, hence H is replaced with H^h. Then $\mathfrak{M} \leftarrow |H^h|$.

Variational recovery of Hessian

Weak definition of the Hessian

$$\int_\sigma H_{ij}^h(a)\,\phi_a^h\,\mathrm{d}x = -\int_\sigma \frac{\partial u^h}{\partial x_i}\frac{\partial \phi_a^h}{\partial x_j}\,\mathrm{d}x$$



a interior point

$\phi_a^h(a)$ is P_1-basis function for a.

Cell-based metric recovery from edge data α_k

Geometric control of edge-based errors

$$\alpha_k \quad \Rightarrow \quad \mathfrak{M} = \begin{bmatrix} \mathfrak{m}_{11} & \mathfrak{m}_{12} \\ \mathfrak{m}_{12} & \mathfrak{m}_{22} \end{bmatrix} ?$$

Cell-based metric recovery from edge data α_k

Geometric control of edge-based errors

$$\alpha_k \quad \Rightarrow \quad \mathfrak{M} = \begin{bmatrix} \mathfrak{m}_{11} & \mathfrak{m}_{12} \\ \mathfrak{m}_{12} & \mathfrak{m}_{22} \end{bmatrix} ?$$

Define quadratic (bubble) function $b_k(x) = \lambda_i(x)\lambda_j(x)$

Consider $v_2 = -\sum_{k=1}^{\#edges} \alpha_k b_k$ and set $\mathfrak{M} = |H(v_2)|$, if $\det H(v_2) \neq 0$.

Otherwise, consider $\hat{v}_2 = -\sum_{k=1}^{\#edges} \hat{\alpha}_k b_k$, where $\hat{\alpha}_{\max} = (1+\delta)\alpha_{\max}$

Theorem. Let α_k be the errors prescribed to edges of a triangle Δ such that

$$\alpha_k \geq 0 \qquad \text{and} \qquad \sum_{k=1}^{\#edges} \alpha_k > 0.$$

Then, there exists a constant tensor metric $\mathfrak{M}$ such that

$$0.4\,|\Delta|_{\mathfrak{M}} \leq \sum_{k=1}^{\#edges} \alpha_k \leq p_{\mathfrak{M}}(\Delta)^2.$$

similar result in 3D

A. Agouzal, Yu. Vassilevski. *Minimization of gradient errors of piecewise linear interpolation on simplicial meshes.* Comp.Meth. Appl.Mech.Engnr., 2010, V.199, p.2195–2203.

Fundamental principle of mesh adaptation

Minimal error is provided by meshes Ω^h where elemental errors $\|u - u_h\|_\Delta$ are equidistributed

Fundamental principle of mesh adaptation

But error $u - u_h$ is unknown, since u is unknown.

Assume that we have reliable and efficient a posteriori error estimates η_Δ

$$c_1 \eta_\Delta \leq \|u - u_h\|_\Delta \leq c_2 \eta_\Delta$$

Fundamental principle of mesh adaptation

Approximate error minimization is provided by meshes Ω^h
where elemental error estimates η_Δ are equidistributed

Fundamental principle of mesh adaptation

Approximate error minimization is provided by meshes Ω^h
where elemental error estimates η_Δ are equidistributed

Important issues:

- **Find η_Δ s.t. $c_2 \approx c_1 \approx 1$ in** $c_1\eta_\Delta \leq \|u - u_h\|_\Delta \leq c_2\eta_\Delta$
- **How to equidistribute η_Δ?**

Fundamental principle of mesh adaptation

Approximate error minimization is provided by meshes Ω^h
where elemental error estimates η_Δ are equidistributed

Important issues:

- **Find η_Δ s.t.** $c_2 \approx c_1 \approx 1$ **in** $c_1\eta_\Delta \le \|u - u_h\|_\Delta \le c_2\eta_\Delta$
- **How to equidistribute η_Δ?**

Fundamental principle of mesh adaptation

Approximate error minimization is provided by meshes Ω^h
where elemental error estimates η_Δ are equidistributed

Important issues:

- **Find η_Δ s.t.** $c_2 \approx c_1 \approx 1$ **in** $c_1\eta_\Delta \leq \|u - u_h\|_\Delta \leq c_2\eta_\Delta$
- **How to equidistribute η_Δ?**

Adaptive algorithm with metric control and aposteriori error estimates

1: Generate an initial mesh Ω^h, solve the problem, estimate the error, and compute a piecewise constant metric $\{\mathfrak{M}_\Delta\}_{\Delta \in \Omega^h}$.

2: **loop**

3: Generate a $\mathfrak{M}$-quasiuniform mesh Ω^h.

4: Solve the problem, estimate the error, and compute a new metric $\{\mathfrak{M}_\Delta\}_{\Delta \in \Omega^h}$.

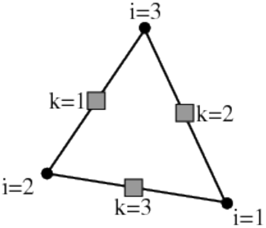
5: If Ω^h is $\mathfrak{M}$-quasiuniform, then exit.

6: **end loop**

Error equidistribution by metric control

Error of piecewise linear interpolation

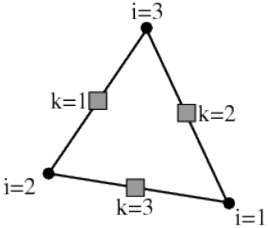
$$
\begin{aligned}
e_2 &= u_2 - \mathcal{I} u_2 \\
&= \sum_{k=1}^{\#edges} (u_2 - \mathcal{I} u_2)(c_k)\, b_k \\
&\equiv \sum_{k=1}^{\#edges} \gamma_k\, b_k,
\end{aligned}
$$



Error equidistribution by metric control

Error of piecewise linear interpolation

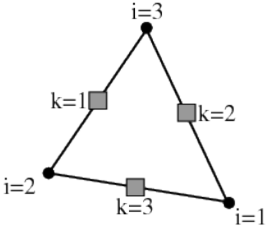
$$\begin{aligned} e_2 &= u_2 - \mathcal{I} u_2 \\ &= \sum_{k=1}^{\#edges} (u_2 - \mathcal{I} u_2)(c_k)\, b_k \\ &\equiv \sum_{k=1}^{\#edges} {\color{blue}\gamma_k}\, b_k, \end{aligned}$$



$$\nabla e_2 = \sum_{k=1}^{\#edges} {\color{blue}\gamma_k}\, \nabla b_k$$

Error equidistribution by metric control

Error of piecewise linear interpolation



$$
\begin{aligned}
e_2 &= u_2 - \mathcal{I} u_2 \\
&= \sum_{k=1}^{\#edges} (u_2 - \mathcal{I} u_2)(c_k)\, b_k \\
&\equiv \sum_{k=1}^{\#edges} \gamma_k\, b_k,
\end{aligned}
$$

$$\nabla e_2 = \sum_{k=1}^{\#edges} \gamma_k \nabla b_k$$

$$\|\nabla e_2\|_2^2 = |\Delta|\,(\mathbb{B}\gamma,\ \gamma)$$

$$\mathbb{B}_{k,l} = \frac{1}{|\Delta|} \int_\Delta \nabla b_k \cdot \nabla b_l \,\mathrm{d}x$$

Bounds on gradient of error

Error of piecewise linear interpolation

We split $\|\nabla e_2\|_2$ into $\#edges$ edge-based error estimates $\alpha_k \geq 0$

$$\|\nabla e_2\|_2 = |\Delta|^{\frac{1}{2}} \sum_{k=1}^{\#edges} \alpha_k \quad \text{and} \quad \sum_{k=1}^{\#edges} \alpha_k = (\mathbb{B}\gamma,\, \gamma)^{\frac{1}{2}}$$

Bounds on gradient of error

Error of piecewise linear interpolation

We split $\|\nabla e_2\|_2$ into $\#edges$ edge-based error estimates $\alpha_k \geq 0$

$$\|\nabla e_2\|_2 = |\Delta|^{\frac{1}{2}} \sum_{k=1}^{\#edges} \alpha_k \quad \text{and} \quad \sum_{k=1}^{\#edges} \alpha_k = (\mathbb{B}\gamma,\, \gamma)^{\frac{1}{2}}$$

We choose

$$\alpha_k = |\gamma_k| (\mathbb{B}\gamma,\, \gamma)^{\frac{1}{2}} \left(\sum_{k=1}^{\#edges} |\gamma_k| \right)^{-1}$$

since $\alpha_k = C|\gamma_k|$ equidistributes $\|e_2\|_{L^\infty}$ over all edges of Δ.

Bounds on gradient of error

Error of piecewise linear interpolation

We split $\|\nabla e_2\|_2$ into $\#edges$ edge-based error estimates $\alpha_k \geq 0$

$$\|\nabla e_2\|_2 = |\Delta|^{\frac{1}{2}} \sum_{k=1}^{\#edges} \alpha_k \quad \text{and} \quad \sum_{k=1}^{\#edges} \alpha_k = (\mathbb{B}\gamma,\, \gamma)^{\frac{1}{2}}$$

We choose

$$\alpha_k = |\gamma_k| (\mathbb{B}\gamma,\, \gamma)^{\frac{1}{2}} \left(\sum_{k=1}^{\#edges} |\gamma_k| \right)^{-1}$$

since $\alpha_k = C|\gamma_k|$ equidistributes $\|e_2\|_{L^\infty}$ over all edges of Δ.

$$\widetilde{\mathfrak{M}} = \det(\mathfrak{M})^{-1/4}\, \mathfrak{M}$$

$$c_1 |\Delta|_{\widetilde{\mathfrak{M}}} |\Delta|_{\widetilde{\mathfrak{M}}} \leq \|\nabla e_2\|_2^2 \leq c_2 |\Delta|_{\widetilde{\mathfrak{M}}} |\partial\Delta|_{\widetilde{\mathfrak{M}}}^2$$

Bounds on gradient of error

Error of piecewise linear interpolation

We split $\|\nabla e_2\|_2$ into $\#edges$ edge-based error estimates $\alpha_k \geq 0$

$$\|\nabla e_2\|_2 = |\Delta|^{\frac{1}{2}} \sum_{k=1}^{\#edges} \alpha_k \quad \text{and} \quad \sum_{k=1}^{\#edges} \alpha_k = (\mathbb{B}\gamma,\, \gamma)^{\frac{1}{2}}$$

We choose

$$\alpha_k = |\gamma_k| (\mathbb{B}\gamma,\, \gamma)^{\frac{1}{2}} \left(\sum_{k=1}^{\#edges} |\gamma_k| \right)^{-1}$$

since $\alpha_k = C|\gamma_k|$ equidistributes $\|e_2\|_{L^\infty}$ over all edges of Δ.

$$\widetilde{\mathfrak{M}} = \det(\mathfrak{M})^{-1/4}\, \mathfrak{M}$$

$$c_1 |\Delta|_{\widetilde{\mathfrak{M}}} |\Delta|_{\widetilde{\mathfrak{M}}} \leq \|\nabla e_2\|_2^2 \leq c_2 |\Delta|_{\widetilde{\mathfrak{M}}} |\partial\Delta|_{\widetilde{\mathfrak{M}}}^2$$

$\widetilde{\mathfrak{M}}$-quasiuniform mesh equidistributes $\|\nabla e\|_2$ and is quasi-optimal mesh

A posteriori error equidistribution by metric control

Hierarchical error estimate for P_1 FEM solution

Let u_h^* be a P_1 finite element solution of a second-order PDE.

Let U_L^* be the vector of degrees of freedom corresponding to nodal basis functions λ_i^h:

$$u_h^* = \sum_{vertices} U_{L,i}^* \, \lambda_i^h$$

The FE method results in the algebraic problem

$$A_{LL} \, U_L^* = F_L.$$

A posteriori error equidistribution by metric control

Hierarchical error estimate for P_1 FEM solution

$$A_{LL}\, U_L^* = F_L.$$

To estimate the discretization error associated with the solution u_h^*, we enrich the FE basis by quadratic (bubble) functions on mesh edges:

$$\begin{aligned} u_h = \sum_{vertices} U_{L,i}\, \lambda_i^h + \sum_{edges} U_{Q,k}\, b_k^h &= u_h^* + d_h \\ d_h &= \sum_{vertices} D_{L,i}\, \lambda_i^h + \sum_{edges} D_{Q,k}\, b_k^h. \end{aligned}$$

The FE method gives a larger algebraic problem for a vector $(U_L,\, U_Q)$.
It is convenient to write it down for the correction vector $(D_L,\, D_Q)$:

$$\begin{bmatrix} A_{LL} & A_{LQ} \\ A_{QL} & A_{QQ} \end{bmatrix} \begin{bmatrix} D_L \\ D_Q \end{bmatrix} = \begin{bmatrix} R_L \\ R_Q \end{bmatrix}, \qquad \begin{aligned} R_L &= F_L - A_{LL}\, U_L^* \\ R_Q &= F_Q - A_{QL}\, U_L^* \end{aligned}$$

A posteriori error equidistribution by metric control

Hierarchical error estimate for P_1 FEM solution

The theory estimates the error as

$$\eta_\Delta := \|\nabla d_{h,Q}\|_{L^2(\Delta)} = \| \sum_{k=1}^{\#edges} (D_{Q,k} \nabla b_k)\|_{L^2(\Delta)}.$$

This estimate gives one number for triangle Δ. However, the coefficients in front of the bubble functions can be used to extract the directional information. We define α_k to be proportional to $D_{Q,k}$:

$$\alpha_k = \frac{|D_{Q,k}|}{\sum\limits_{k=1}^{\#edges} |D_{Q,k}|} \eta_\Delta.$$

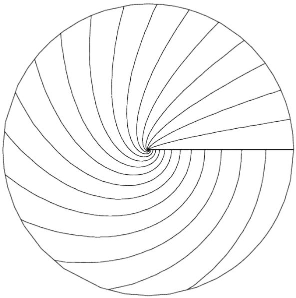
Using these α_k, we constuct a constant metric $\widetilde{\mathfrak{M}}_\Delta$ in Δ.

Diffusion problem with isotropic solution

Let Ω be a unit disk with a radial cut. We consider the classical crack problem with the exact solution

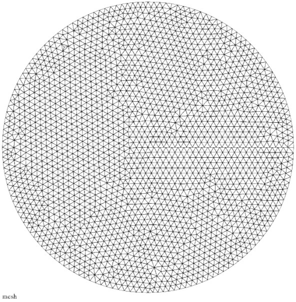
$$u(r,\theta) = r^{1/4}\sin(\theta/4), \qquad \theta \in [0,\, 2\pi).$$

We consider the following boundary value problem:



$$\begin{aligned} \Delta\, u &= 0 && \text{in} \quad \Omega \setminus S \\ u &= \sin(\theta/4) && \text{on} \quad \partial\Omega \setminus S \\ u &= 0 && \text{on} \quad S^+ \\ \frac{\partial u}{\partial n} &= 0 && \text{on} \quad S^- \end{aligned}$$

Diffusion problem with isotropic solution



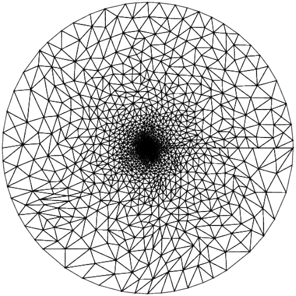
N_T	$\|\nabla d_{h,Q}\|_{L^2(\Omega)}$	$\|\nabla e\|_{L^2(\Omega)}$
1000	1.08e-1	1.09e-1
4000	5.42e-2	5.38e-2
16000	2.72e-2	2.73e-2
64000	1.36e-2	1.52e-2
rate	0.50	0.47

The adaptive mesh is isotropic. The error reduction is proportional to

$$N_T^{-1/2}$$

which is the theoretically predicted optimal estimate.

Diffusion problem with isotropic solution



N_T	$\|\nabla d_{h,Q}\|_{L^2(\Omega)}$	$\|\nabla e\|_{L^2(\Omega)}$
1000	1.08e-1	1.09e-1
4000	5.42e-2	5.38e-2
16000	2.72e-2	2.73e-2
64000	1.36e-2	1.52e-2
rate	0.50	0.47

The adaptive mesh is isotropic. The error reduction is proportional to

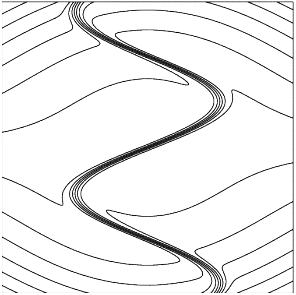
$$N_T^{-1/2}$$

which is the theoretically predicted optimal estimate.

Diffusion problem with anisotropic solution

Let Ω be a square $(-1,\ 1)^2$. We consider the boundary value problem with the exact solution

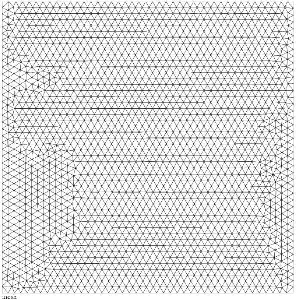
$$u(x,\ y) = yx^2 + y^3 + \tanh(6(\sin(5y) - 2x)).$$



$$\begin{array}{rcll} -\mathrm{div}(K\,\nabla u) & = & f & \text{in} \quad \Omega \\ u & = & u_0 & \text{on} \quad \partial\Omega \end{array}$$

where $K = \mathrm{diag}\{1, 0.1\}$.

Diffusion problem with anisotropic solution



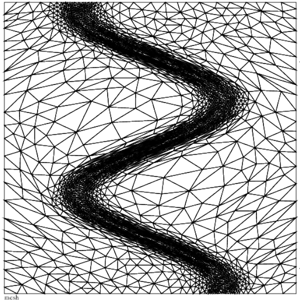
N_T	$\|K^{\frac{1}{2}} \nabla d_{h,Q}\|_{L^2(\Omega)}$	$\|K^{\frac{1}{2}} \nabla e\|_{L^2(\Omega)}$
1000	8.21e-1	8.03e-1
4000	4.16e-1	3.77e-1
16000	2.07e-1	1.87e-1
64000	1.29e-1	9.44e-2
rate	0.45	0.51

The adaptive mesh is anisotropic, $\max_{\Delta} R_{\Delta}/r_{\Delta} = 7600$.
The error reduction is proportional to

$$N_T^{-1/2}$$

which is the optimal estimate.

Diffusion problem with anisotropic solution



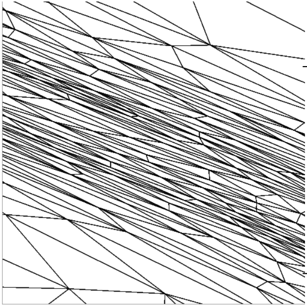
N_T	$\lVert K^{\frac{1}{2}} \nabla d_{h,Q} \rVert_{L^2(\Omega)}$	$\lVert K^{\frac{1}{2}} \nabla e \rVert_{L^2(\Omega)}$
1000	8.21e-1	8.03e-1
4000	4.16e-1	3.77e-1
16000	2.07e-1	1.87e-1
64000	1.29e-1	9.44e-2
rate	0.45	0.51

The adaptive mesh is anisotropic, $\max_{\Delta} R_\Delta / r_\Delta = 7600$.
The error reduction is proportional to

$$N_T^{-1/2}$$

which is the optimal estimate.

Diffusion problem with anisotropic solution



N_T	$\|K^{\frac{1}{2}}\nabla d_{h,Q}\|_{L^2(\Omega)}$	$\|K^{\frac{1}{2}}\nabla e\|_{L^2(\Omega)}$
1000	8.21e-1	8.03e-1
4000	4.16e-1	3.77e-1
16000	2.07e-1	1.87e-1
64000	1.29e-1	9.44e-2
rate	0.45	0.51

The adaptive mesh is anisotropic, $\max_{\Delta} R_\Delta / r_\Delta = 7600$.
The error reduction is proportional to

$$N_T^{-1/2}$$

which is the optimal estimate.

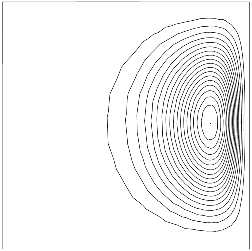
Navier-Stokes equations

Let Ω be a square $(0,1)^2$. We consider the Navier-Stokes equations with the exact solution $(\mathbf{u}, p)$, $\mathbf{u} = (v, w)$ (Berrone, 2001)

$$
\begin{aligned}
v(x,y) &= \left(1-\cos\left(\frac{2\pi(e^{R_1 x}-1)}{e^{R_1}-1}\right)\right)\sin\left(\frac{2\pi(e^{R_2 y}-1)}{e^{R_2}-1}\right)\frac{R_2}{2\pi}\frac{e^{R_2 y}}{(e^{R_2}-1)} \\
w(x,y) &= -\sin\left(\frac{2\pi(e^{R_1 x}-1)}{e^{R_1}-1}\right)\left(1-\cos\left(\frac{2\pi(e^{R_2 y}-1)}{e^{R_2}-1}\right)\right)\frac{R_1}{2\pi}\frac{e^{R_1 x}}{(e^{R_1}-1)} \\
p(x,y) &= \sin\left(\frac{2\pi(e^{R_1 x}-1)}{e^{R_1}-1}\right)\sin\left(\frac{2\pi(e^{R_2 y}-1)}{e^{R_2}-1}\right)R_1 R_2\frac{e^{R_1 x}e^{R_2 y}}{(e^{R_1 x}-1)(e^{R_2 y}-1)}
\end{aligned}
$$

where $R_1 = 4.2985$, $R_2 = 0.1$

$\mathbf{u}$ represents a counterclockwise vortex



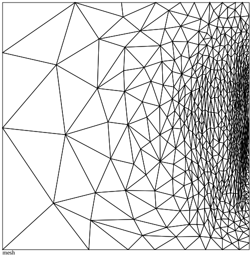
$$
\begin{aligned}
-0.1\Delta\,\mathbf{u} + (\mathbf{u}\cdot\nabla)\mathbf{u} + \nabla p &= \mathbf{f} && \text{in } \Omega, \\
\operatorname{div}\mathbf{u} &= 0 && \text{in } \Omega, \\
\mathbf{u} &= 0 && \text{on } \partial\Omega
\end{aligned}
$$

Navier-Stokes equations

Hood-Taylor $P_2 - P_1$ FEM

$$\gamma_k^{(1)} = (v_h - \mathcal{I}_1 v_h)(c_k), \quad \gamma_k^{(2)} = (w_h - \mathcal{I}_1 w_h)(c_k)$$

$$\alpha_k = \left(|\gamma_k^{(1)}| + |\gamma_k^{(2)}|\right) \left((\mathbb{B}\gamma^{(1)}, \gamma^{(1)}) + (\mathbb{B}\gamma^{(2)}, \gamma^{(2)})\right) \left(\sum_{k=1}^{3} |\gamma_k^{(1)}| + |\gamma_k^{(2)}|\right)^{-1}$$



N_T	$\tilde{E}(\mathbf{u})$	$E(\mathbf{u})$
1000	3.3e-1	7.2e-2
4000	1.6e-1	2.1e-2
16000	8.1e-2	6.1e-3
64000	4.0e-2	1.7e-3
rate	0.5	0.9

$$\tilde{E}(\mathbf{u}) = \left(\|\nabla(\mathcal{I}_1 v_h - v)\|^2_{L^2(\Omega)} + \|\nabla(\mathcal{I}_1 w_h - w)\|^2_{L^2(\Omega)}\right)^{1/2}$$

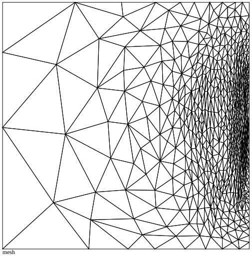
Reduction of $\tilde{E}(\mathbf{u})$ is proportional to $N_T^{-1/2}$

Navier-Stokes equations

Hood-Taylor $P_2 - P_1$ FEM

$$\gamma_k^{(1)} = (v_h - \mathcal{I}_1 v_h)(c_k), \quad \gamma_k^{(2)} = (w_h - \mathcal{I}_1 w_h)(c_k)$$

$$\alpha_k = \left(|\gamma_k^{(1)}| + |\gamma_k^{(2)}|\right) \left((\mathbb{B}\gamma^{(1)}, \gamma^{(1)}) + (\mathbb{B}\gamma^{(2)}, \gamma^{(2)})\right) \left(\sum_{k=1}^{3} |\gamma_k^{(1)}| + |\gamma_k^{(2)}|\right)^{-1}$$



N_T	$\tilde{E}(\mathbf{u})$	$E(\mathbf{u})$
1000	3.3e-1	7.2e-2
4000	1.6e-1	2.1e-2
16000	8.1e-2	6.1e-3
64000	4.0e-2	1.7e-3
rate	0.5	0.9

$$E(\mathbf{u}) = \left(\|\nabla(v_h - v)\|^2_{L^2(\Omega)} + \|\nabla(w_h - w)\|^2_{L^2(\Omega)}\right)^{1/2}$$

(!) Reduction of $E(\mathbf{u})$ is proportional to N_T^{-1}

Advanced numerical instruments Ani#D

Ani2D	Ani3D
www.sf.net/projects/ani2d	www.sf.net/projects/ani3d
6300 downloads	4000 downloads

Alternatives: FreeFEM (F.Hecht)

Advanced Numerical Instruments, Ani2D

Open source software for FEM solution of BVPs on triangulations

Ani2D is a set of independent libraries for

- mesh generation
- mesh adaptation (hierarchical or metric-based, isotropic or anisotropic)
- FEM discretization of 2nd order PDEs
- solution of algebraic systems (linear and nonlinear)
- visualization of mesh and FEM solution

Advanced Numerical Instruments, Ani2D

Open source software for FEM solution of BVPs on triangulations

Ani2D is a set of independent libraries for

- mesh generation
- mesh adaptation (hierarchical or metric-based, isotropic or anisotropic)
- FEM discretization of 2nd order PDEs
- solution of algebraic systems (linear and nonlinear)
- visualization of mesh and FEM solution

The libraries may be combined to solve complex problems.
The libraries can be included easily in other packages.

Ani2D is released under the GNU GPL Licence, tested under Linux, Unix, Windows.

Ani2D developers

Principal developers:

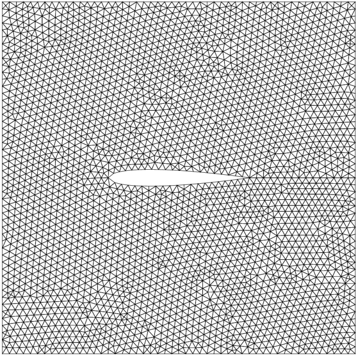
- Konstantin Lipnikov (LANL)
- Yuri Vassilevski (INM RAS)

Developers:

- Alexander Danilov (INM RAS)
- Vadim Chugunov (INM RAS)
- Sergei Goreinov (INM RAS)
- Alexey Chernyshenko (INM RAS)

Meshing package aniAFT (Advancing Front Technique)

Analytical representation of boundary

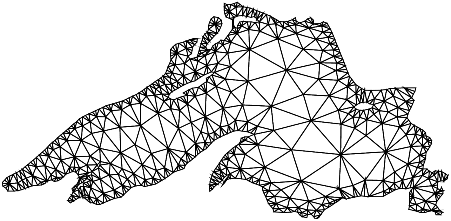


```
! complement of a wing NACA0012 to the unit square
double precision bv(2,7), bltail(2,8)
integer Nbv, Nbl, bl(7,8)
! numbers of boundary nodes and boundary edges
data Nbv/7/, Nbl/8/
! boundary nodes
data bv/0,0, 0,1, 1,1, 1,0, .4,.5, .6,.5, 1,.5/
! outer boundary edges
data bl/1,2,0,-1,-1,1,0, 4,1,0,-1,-1,1,0,
& 2,3,0,-1,1,1,0, 7,4,0,-1,1,1,0,
& 3,7,0,-1,1,1,0, 6,7,2,0,11,1,1,
& 6,5,1,-1,2,1,0, 5,6,1,-1,2,1,0/
! curved data for each outer boundary edge
data bltail/0,0, 0,0, 0,0, 0,0, 0,0, 0,1, 0,.5, .5,1/
```

```
external userboundary
call registeruserfn(userboundary)
...
ierr = aft2dboundary(Nbv, bv, Nbl, bl, bltail, h,

nv, vrt, nt, tri, labelT, nb, bnd, labelB, nc, crv, iFNC)
```

Meshing package aniAFT (Advancing Front Technique)

Grid representation of boundary



518 518 number of vertices and edges
0. 0.064933 coordinates of vertices
0.002293 0.059187
0.007467 0.055733
0.01092 0.050573
....
0.648853 0.1954
3 4 connectivity list for edges
4 5
5 6
....
235 236

ierr = aft2dfront(Nbr, brd, Nvr, vbr,
nv, vrt, nt, tri, labelT, nb, bnd, labelB)

Mesh size control

- generate quasi-uniform mesh with meshstep h:

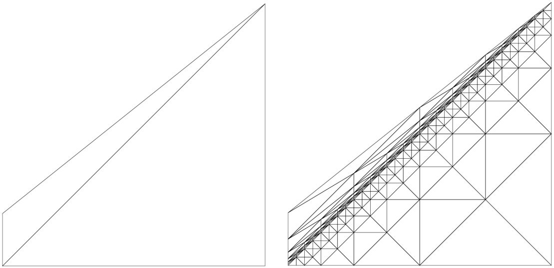
  h = 0.02
  ierr = aft2dboundary(Nbv, bv, Nbl, bl, bltail, h, ...

- user's meshsize function:

  external usermeshsize
  call registersizefn(usermeshsize)

- no control (mesh size geometric coarsening)

Hierarchical local refinement



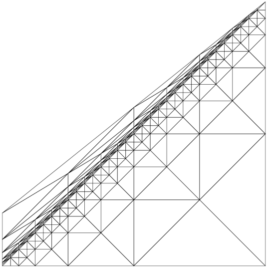
Subroutine RefineRule (nt, tri, vrt, verf, ilevel)
! refine towards the diagonal y=x
Do i = 1, nt
....
! at least one vertex belongs to y=x
If (xy .eq. 0) then
 verf(i) = 2 ! two levels of bisection
Else
 verf(i) = 0 ! no need to refine
End if
End do

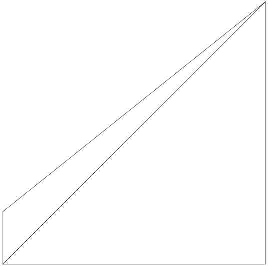
external RefineRule
...
Do ilevel = 1, 5
 Call LocalRefine (nv, nvmax, nb, nbmax, nt, ntmax,
 vrt, tri, bnd, labelB, labelT,
 RefineRule, ilevel, maxlevel, history, MaxWi, iW, iERR)

End do

Refining/Coarsening by Bisection, aniRCB

Hierarchical local coarsening



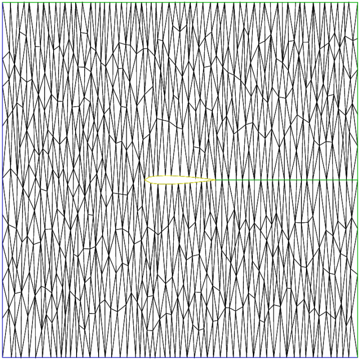


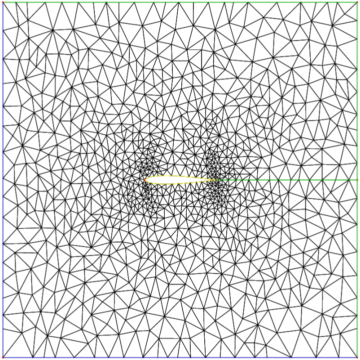
```
Subroutine CoarseRule (nE, IPE,
XYP, verf, ilevel)
...
! coarse towards the diagonal y=x
Do i = 1, nt
If (xy .eq. 0) then
 verf(i) = 2 ! two levels of merging
Else
 verf(i) = 0 ! no need to coarse
End if
End do
```

```
external CoarseRule
...
Do ilevel = 5, 1, -1
 Call LocalCoarse ( nv, nvmax, nb, nbmax, nt, ntmax,
 vrt, tri, bnd, labelB, labelT,
 CoarseRule, ilevel, maxlevel, history, MaxWi, iW, iERR)
End do
```

Metric-based adaptation, aniMBA

generation of a mesh using an analytic metric

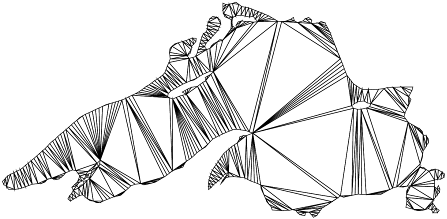


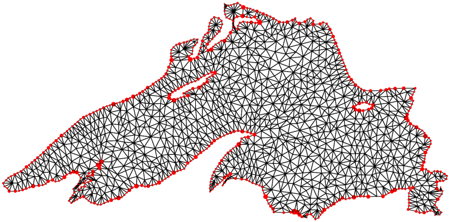


Call mbaAnalytic(
nv, nvfix, nvmax, vrt, labelv, fixedV,
nb, nbfix, nbmax, bnd, labelB, fixedB,
nc, Crv, iFnc, CrvFunction,
nt, ntfix, ntmax, tri, labelT, fixedT,
nEStar, Quality, control, MetricFunction,
MaxWr, MaxWi, rW, iW, iERR)

Metric-based adaptation, aniMBA

mesh cosmetics and untangling fixes the elements with bad shape and even tangled





```
Call mbaFixShape(
 nv, nvfix, nvmax, vrt, labelv, fixedV,
 nb, nbfix, nbmax, bnd, labelB, fixedB,
 nc, Crv, iFnc, ANI_CrvFunction,
 nt, ntfix, ntmax, tri, labelT, fixedT,
 nEStar, Quality, control, ANI_MetricFunction,
 MaxWr, MaxWi, rW, iW, iERR)
```

Metric-based adaptation, aniMBA

generation of a mesh using a user-defined metric at input mesh nodes

```
Call mbaNodal(
 nv, nvfix, nvmax, vrt, labelv, fixedV,
 nb, nbfix, nbmax, bnd, labelB, fixedB,
 nc, Crv, iFnc, CrvFunction,
 nt, ntfix, ntmax, tri, labelT, fixedT,
 nEStar, Quality, control, Metric,
 MaxWr, MaxWi, rW, iW, iERR)
```

Real*8 Metric(3,nvmax)

$$\begin{pmatrix} M_{11} & M_{12} \\ M_{12} & M_{22} \end{pmatrix}$$

Metric-based adaptation, aniMBA

generation of a mesh using a user-defined metric at input mesh nodes

```
Call mbaNodal(
 nv, nvfix, nvmax, vrt, labelv, fixedV,
 nb, nbfix, nbmax, bnd, labelB, fixedB,
 nc, Crv, iFnc, CrvFunction,
 nt, ntfix, ntmax, tri, labelT, fixedT,
 nEStar, Quality, control, Metric,
 MaxWr, MaxWi, rW, iW, iERR)
```

Real*8 Metric(3,nvmax)

$$\begin{pmatrix} M_{11} & M_{12} \\ M_{12} & M_{22} \end{pmatrix}$$

Local Metric Recovery, aniLMR

Local metric recovery from discrete function (P_1 FEM)

```
Call Nodal2MetricVAR(U,
vrt, nv, tri, nt, nnd, nb, Metric,
MaxWr, rW, MaxWi, iW)
```

Local Metric Recovery, aniLMR

Local metric recovery from edge-based error estimator

```
Call EdgeEst2MetricMAX(Error, nv, nt, vrt, tri,
 Metric, MaxWr, rW)

Call EdgeEst2GradMetricMAX(Error, nv, nt, vrt, tri,
 Metric, MaxWr, rW)
```

Metric can be modified for error minimization in L^p:

```
Call Lp_norm(nP, Lp, Metric)
```

Finite Element Method Discretization, aniFEM

Computing of elemental matrix on a triangle $< D\, Op_A(u),\, Op_B(v) >$

- D is a tensor,
- Op_A and Op_B are linear first-order or zero-order differential operators,
- u and v are finite element basis functions.

Finite Element Method Discretization, aniFEM

Computing of elemental matrix on a triangle $< D\, Op_A(u), Op_B(v) >$

```
Call FEM2Dtri(XY1, XY2, XY3,
              OpA, FemA, OpB, FemB,
              label, Dcoef, dDATA, iDATA, iSYS, order,
              LDA, A, nRow, nCol)

Call FEM2Dext(XY1, XY2, XY3,
              lbE, lbF, lbP, dDATA, iDATA, iSYS,
              LDA, A, F, nRow, nCol,
              template, templateC)
```

Finite Element Method Discretization, aniFEM

Computing of elemental matrix on a triangle $< D\, Op_A(u),\, Op_B(v) >$

Finite elements

FEM_P0	The piecewise constant (T_1).
FEM_P1	The continuous piecewise linear (V_1, V_2, V_3).
FEM_P2	The continuous piecewise quadratic ($V_1, V_2, V_3,\ E_1, E_2, E_3$).
FEM_P3	The continuous piecewise cubic ($V_1, V_2, V_3,\ E_1, E_2, E_3,\ E_1, E_2, E_3,\ T_1$).
FEM_P4	The continuous piecewise quartic ($V_1, V_2, V_3,\ E_1, E_2, E_3,\ E_1, E_2, E_3,\ E_1, E_2, E_3,\ T_1, T_1, T_1$).
FEM_P1vector	The continuous vector piecewise linear ($V_1, V_2, V_3,\ V_1, V_2, V_3$).
FEM_P2vector	The continuous vector piecewise quadratic. The unknowns are ordered first as in the quadratic element and then by the space dimension.
FEM_P2reduced	The Bernardi-Fortin-Raugel finite element, the continuous vector piecewise linear functions enriched by edge bubbles ($V_1, V_2, V_3,\ E_1, E_2, E_3\ V_1, V_2, V_3$).
FEM_MINI	The continuous vector piecewise linear functions enriched by a central bubble ($V_1, V_2, V_3,\ E_1,\ V_1, V_2, V_3,\ E_1$)
FEM_RT0	The lowest order Raviart-Thomas finite elements
FEM_BDM1	The lowest order Brezzi-Douglas-Marini finite elements
FEM_CR1	The Crouzeix-Raviart finite element.
FEM_CR1vector	The vector Crouzeix-Raviart finite element. The unknowns are ordered first by vertices and then by the space directions (x and y).

Finite Element Method Discretization, aniFEM

Computing of elemental matrix on a triangle $< D\, Op_A(u), Op_B(v) >$

Discrete operators Op_A and Op_B.

IDEN	identity operator
GRAD	gradient operator
DIV	divergence operator
CURL	rotor operator
DUDX	partial derivative d/dx
DUDY	partial derivative d/dy
DUDN	partial derivative in direction of an exterior normal

Finite Element Method Discretization, aniFEM

Computing of elemental matrix on a triangle $< D\, Op_A(u), Op_B(v) >$

Tensors D

TENSOR_NULL	identity tensor
TENSOR_SCALAR	scalar tensor
TENSOR_SYMMETRIC	symmetric tensor
TENSOR_GENERAL	general (rectangular or non-symmetric) tensor

Finite Element Method Discretization, aniFEM

Computing of elemental matrix on a triangle $< D\, Op_A(u),\ Op_B(v) >$

Quadrature formulae:

order $= 1$	quadrature formula with one central point
order $= 2$	quadrature formula with 3 points on triangle edges
order $= 5$	quadrature formula with 7 points inside triangle
order $= 6$	quadrature formula with 12 points inside triangle
order $= 9$	quadrature formula with 19 points inside triangle
order $= 13$	quadrature formula with 37 points inside triangle

Finite Element Method Discretization, aniFEM

Computing of elemental matrix on a triangle $< D\, Op_A(u),\, Op_B(v) >$

Assembling routine

```
Subroutine BilinearFormTemplate(
        nP, nF, nE, XYP, lbP, IPF, lbF, IPE, lbE,
        FEM2Dext, dDATA, iDATA, control,
        MaxF, MaxA, IA, JA, A, F, nRow, nCol,
        MaxWi, MaxWr, iW, rW)
```

Sparse matrix is output in CSR/CSC/AMG format

Finite Element Method Discretization, aniFEM

Computing of elemental matrix on a triangle $< D\, Op_A(u),\, Op_B(v) >$

Error calculation:

$$\|u - u_h\|_*^p = \int_\Delta |D\,(Op_A(u_h) - u) \cdot (Op_A(u_h) - u)|^{p/2}\,\mathrm{d}x,$$

```
Call fem2Derr(XY1, XY2, XY3, Lp,
        operatorA, FEMtypeA, Uh, Fu, dDATAFU, iDATAFU,
        label, D, dDATA, iDATA, iSYS, order, ERR)
```

Solvers of algebraic systems

LU sparse factorization, aniLU

$$Ax = b$$

$$P_1 A P_2 = LU$$

- A is given in CSC format
- UMFPACK v5.1 library

Solvers of algebraic systems

Iterative solvers with ILU preconditioners, aniILU

$$Ax = b$$

- A is given in CSR format
- BiCGstab, GMRES(k), PCG
- ILU0, ILU2 (second order)

```
Call slpbcgs(prevec, IPREVEC, iW,rW,
             matvec, IMATVEC, ia,ja,a,
             WORK, MW, NW,
             N, RHS, SOL,

             ITER, RESID, INFO, NUNIT)
```

Solvers of algebraic systems

Inexact Newton-Krylov Jacobian-Free with ILU preconditioners, aniINB

$$F(u) = 0$$

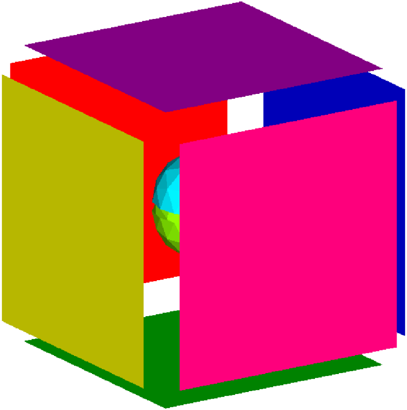
- F is function given by user
- BiCGstab, Jacobian-Free: $J(u)v \approx \delta^{-1}(F(u+\delta v) - F(u))$
- ILU0, ILU2 (second order) preconditioners

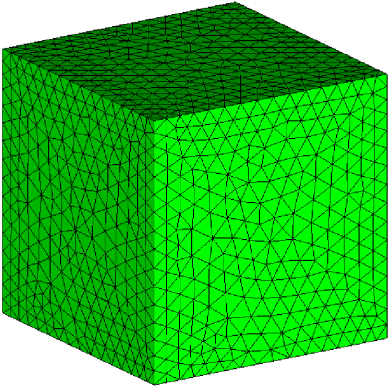
```
external prevec, funvec
....
Call slInexactNewton(prevec, IPREVEC, iWprevec, rWprevec,
         funvec, rpar, ipar,
         N, SOL,

         RESID, STPTOL, rWORK, LenrWORK, INFO)
```

Options for domain definition in Ani3D

http://sourceforge.net/projects/ani3d

Boundary as a union of smooth parameterized patches

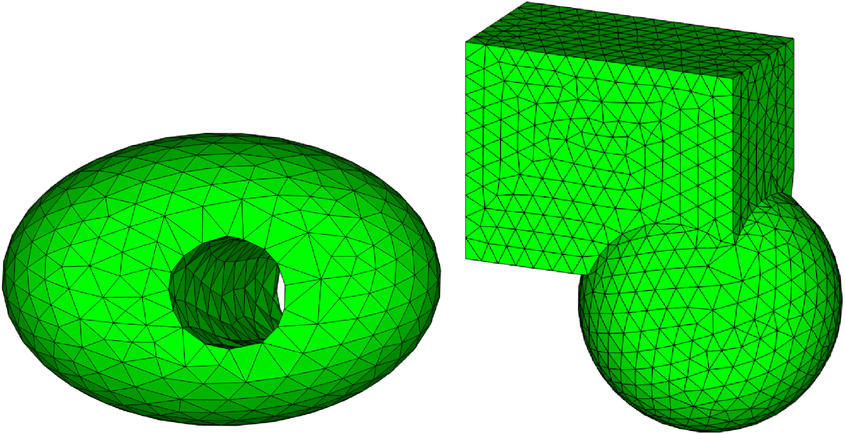




Options for domain definition in Ani3D

http://sourceforge.net/projects/ani3d

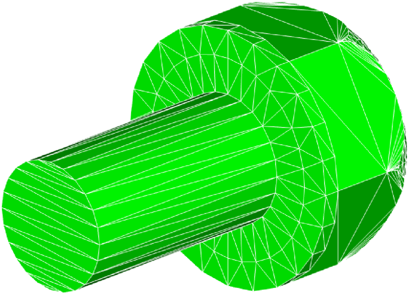
Boundary as a combination of given front primitives

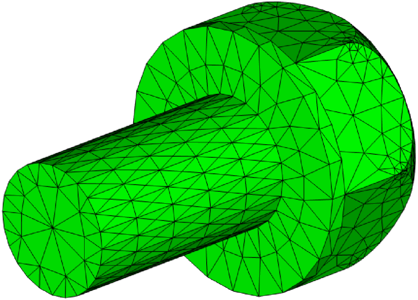


Options for domain definition in Ani3D

http://sourceforge.net/projects/ani3d

Boundary as a CAD mesh with triangular facets





Options for domain definition in Ani3D

http://sourceforge.net/projects/ani3d

Boundary representation through a CAD system