r-refinement in isogeometric analysis

Gang Xu, Bernard Mourrain, Régis Duvigneau, André Galligo

GALAAD and OPALE Team, INRIA Sophia-Antipolis

University of Nice Sophia-Antipolis

Gang.Xu@inria.fr

21st September, 2010

Outline

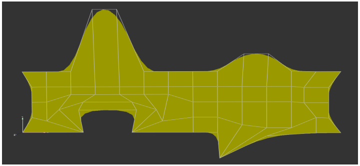
1. *Introduction of isogeometric analysis (IGA)*
2. *Model quality in isogeometric analysis*
3. *Introduction of r-refinement in isogeometric analysis*
4. *r-refinement for problems with unknown exact solution*
   - Residual-based a posteriori error estimation
   - r-refinement and error assessment
   - Examples and comparison
5. *Conclusion and future work*

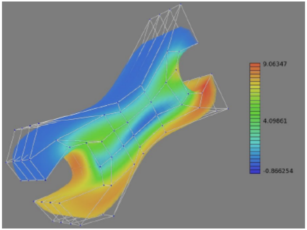
Isogeometric analysis (IGA)

- IGA is an isoparametric, exact geometry approach, which is recently providing very promising results as an alternative to finite element analysis (FEA).
- proposed by Prof. T. Hughes et al. from University of Texas at Austin in 2005
- motivation:
  - seamless integration of CAD and CAE.
  - avoid geometry approximations of mesh generation in FEA
  - high regularity and refinement of B-spline functions.
- basic idea: use the same standard mathematical representation as in CAD systems (such as NURBS) for both the geometry and the solution field (such as thermal conduction).

Representation in IGA

- computational domain:
  - 2D: planar B-spline surface
  - 3D: B-spline volume
- solution field :
  - 2D: B-spline surface with 3D control points
  - 3D: B-spline volume with 4D control points



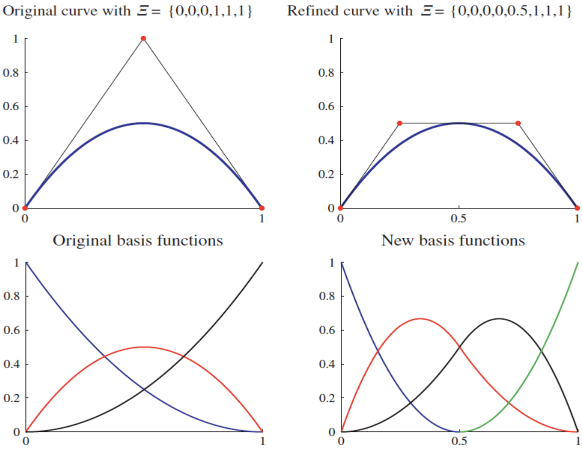


Related work

- application in various simulation problems : [Y. Bazilevs et al.2006],[J.A. Cottrell et al.2006], [Y. Bazilevs et al.2008]...
- application of various geometric modeling tools in IGA
  - NURBS : [T. Hughes et al., 2005]
  - T-spline: [M. Dörfel et al., 2010],[Y. Bazilevs et al.2010]
  - Subdivision surface: [F. Cirak et al., 2000]
  - PHT-spline: [Tian et al., 2010]
  - Catmull-Clark subdivision solids: [Burkhart et al., 2010]
- accuracy and efficiency improvement of IGA framework by reparameterization and refinement operations
  - h-refinement: knot insertion [J.A. Cottrell et al.2007]
  - p-refinement: order elevation [T. Hughes et al., 2005]
  - k-refinement: order elevation + knot insertion [T. Hughes et al. 2008]

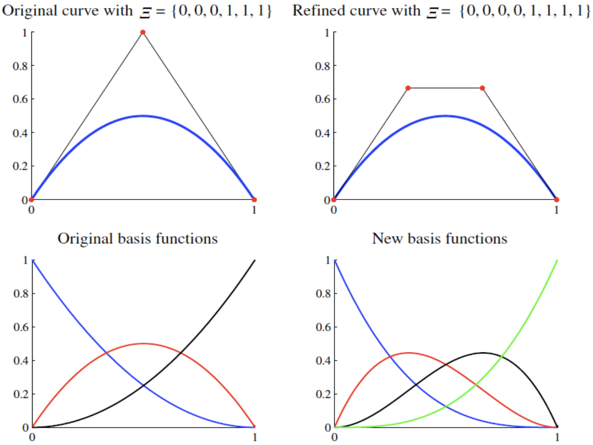
h-refinement

- knot insertion: adding a new knot into the existing knot vector without changing the shape of curve/surface/volume



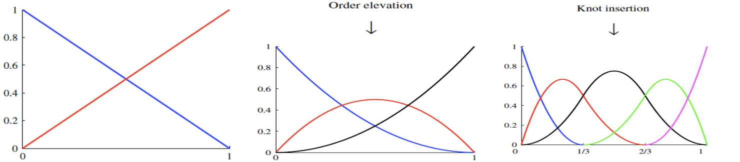
p-refinement

- order elevation: increase the order of B-spline basis function without changing the shape of the curve/surface/volume



k-refinement

- Order elevation + knot insertion : p-h-refinement



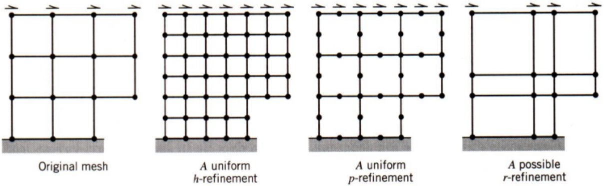
$\Xi = \{0,0,1,1\},\ p = 1$ $\Xi = \{0,0,0,1,1,1\},\ p = 2$ $\Xi = \{0,0,0,\frac{1}{3},\frac{2}{3},1,1,1\},\ p = 2$

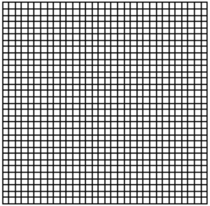
Remark

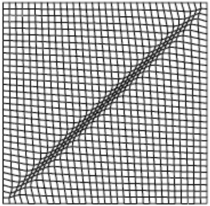
- p-h-k refinement: increase number of freedom (control points) to achieve better analysis results without changing the geometry
- Keep number of freedom as constant to obtain better results ?
- One option: optimize the placement of inner control points
- Parameterization of computational domain is changed while keeping the boundary geometry

r-refinement in FEA

- r-refinement: remesh operation on computational domain to minimize the cost function while keeping the number of elements as a constant.
- moving nodes, moving mesh,....
- r-refinement in IGA: similarity and difference?







Outline

1. *Introduction of isogeometric analysis (IGA)*
2. ***Model quality in isogeometric analysis***
3. *Introduction of r-refinement in isogeometric analysis*
4. *r-refinement for problems with unknown exact solution*
   - Residual-based a posteriori error estimation
   - r-refinement and error assessment
   - Examples and comparison
5. *Conclusion and future work*

Test model — heat conduction problem

Given a domain Ω with $\Gamma = \partial\Omega_D \cup \partial\Omega_N$,

$$
\begin{aligned}
\nabla(\kappa(x)\nabla U(x)) &= f(x) \quad \text{in } \Omega \\
U(x) &= U_0(x) \text{ on } \partial\Omega_D \\
\kappa(x)\frac{\partial U}{\partial n}(x) &= \Phi_0(x) \text{ on } \partial\Omega_N,
\end{aligned}
\tag{1}
$$

where x are the Cartesian coordinates, U represents the temperature field and κ the thermal conductivity. Dirichlet and Neumann boundary conditions are applied on $\partial\Omega_D$ and $\partial\Omega_N$ respectively, T_0 and Φ_0 being the imposed temperature and thermal flux (n unit vector normal to the boundary). f is a user-defined function that allows to generate problems with an analytical solution, by adding a source term to the classical heat conduction equation.

Isogeometric solving

- weak form

$$\int_{\Omega} \nabla(\kappa(x)\nabla U(x))\ \psi(x)\ d\Omega = \int_{\Omega} f(x)\ \psi(x)\ d\Omega \quad \forall \psi \in H^1_{\partial\Omega_D}(\Omega),$$

- integration by parts

$$-\int_{\Omega} \kappa(x)\nabla U(x)\ \nabla\psi(x)\ d\Omega + \int_{\partial\Omega_N} \Phi_0(x)\ \psi(x)\ d\Gamma = \int_{\Omega} f(x)\ \psi(x)\ d\Omega.$$

- temperature field and test function

$$\begin{aligned} T(\xi,\eta) &= \sum_{i=1}^{n_i} \sum_{j=1}^{n_j} \hat{N}_i^{p_i}(\xi)\, \hat{N}_j^{p_j}(\eta)\, T_{ij}, \\ \psi(x) &= N_{ij}(x) = \hat{N}_{ij} \circ \sigma^{-1}(x) = \hat{N}_{ij}(\xi,\eta) = \hat{N}_i^{p_i}(\xi)\, \hat{N}_j^{p_j}(\eta). \end{aligned}$$

where σ is the map from parametric domain $\mathcal{P}$ to computational domain Ω.

Isogeometric solving—continued

$$\sum_{k=1}^{n_k}\sum_{l=1}^{n_l} T_{kl}\int \kappa(x)\nabla N_{kl}(x)\nabla N_{ij}(x)d\Omega = \int \Phi_0(x)N_{ij}(x)d\Gamma - \int f(x)N_{ij}(x)d\Omega.$$

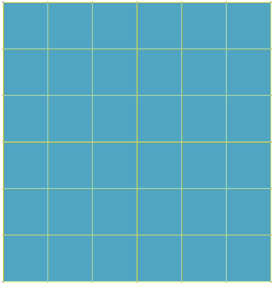
- rewritten weak formulation
- stiffness matrix and right-hand term

$$\begin{aligned} M_{ij,kl} &= \int_\Omega \kappa(x)\nabla N_{kl}(x)\ \nabla N_{ij}(x)\ d\Omega \\ &= \int_{\mathcal{P}} \kappa(T(u))\nabla_u \tilde{N}_{kl}(u)B(u)^T B(u)\ \nabla_u \tilde{N}_{kl}(u)J(u)\ d\mathcal{P} \\ S_{ij} &= \int_{\partial\mathcal{P}_N} \Phi_0(T(u))\ \tilde{N}_{kl}(u)J(u)\ d\tilde{\Gamma} - \int_{\mathcal{P}} f(T(u))\ \tilde{N}_{kl}(u)J(u)\ d\mathcal{P}. \end{aligned}$$

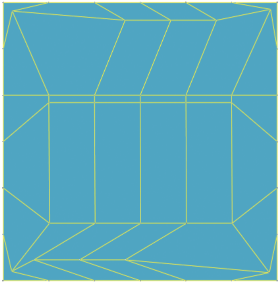
where J is the Jacobian of the transformation, B^K is the transposed of the inverse of the Jacobian matrix.

Given ....

- source term : $\boldsymbol{f}(x,y) = -\frac{4\pi^2}{9}\sin(\frac{\pi x}{3})\sin(\frac{\pi y}{3})$.
- boundary condition : $\boldsymbol{U}_0(\boldsymbol{x}) = 0$ and $\Phi_0(\boldsymbol{x}) = 0$
- exact solution : $\boldsymbol{U}(x,y) = 2\sin(\frac{\pi x}{3})\sin(\frac{\pi y}{3})$.
- computational domain : $\Omega(x,y) = [0,6]\times[0,6]$

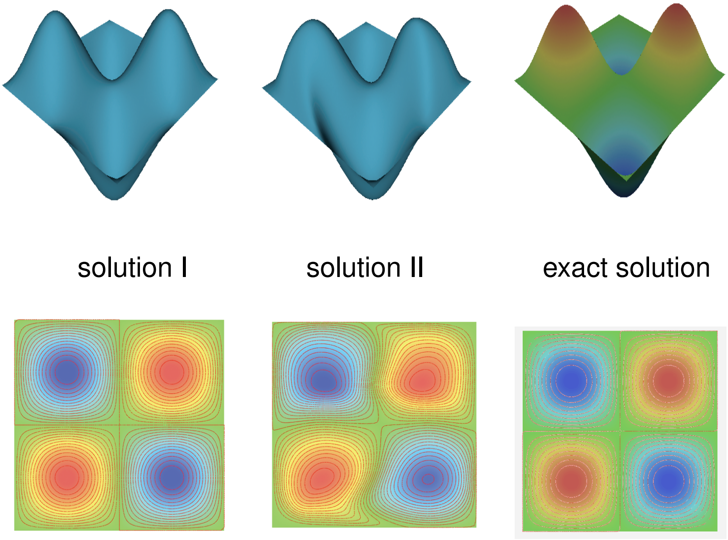


control point placement I



control point placement II

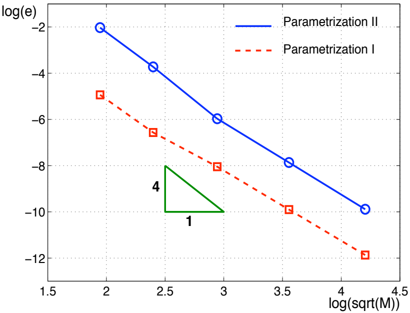
Solution comparison



Error analysis

- relative L_2 norm:

$$e = \sqrt{\frac{\int_\Omega (\boldsymbol{T} - \tilde{\boldsymbol{T}})^T (\boldsymbol{T} - \tilde{\boldsymbol{T}}) d\Omega}{\int_\Omega \boldsymbol{T}^T \boldsymbol{T} d\Omega}}$$



Remark

- Model quality has some effect on analysis results
- accuracy and efficiency
- How to improve model quality for isogeometric analysis ?

Outline

Problem statement

r-refinement

given initial placement of control points of computational domain, reposition the inner control points to achieve more accurate simulation results in isogeometric analysis

Main idea

- Inspired from shape optimization
- Shape optimization: optimize the boundary model to minimize the error function (cost function) of simulation

Main idea

Let the inner control points, rather than boundary control points, be the design variables for the shape optimization, and find the best placement of inner control points to make the value of a cost function as small as possible.

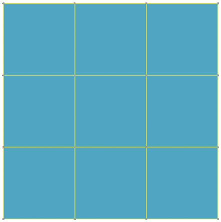
Main approach

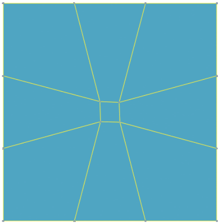
- optimization variables: the coordinates of the inner control points
- cost function: error of the IGA solution
- optimization algorithm: steepest-descent method in conjunction with a back-tracking line-search
  1. Evaluation of perturbed points $x_k + \epsilon e_k$
  2. Estimation of the gradient $\nabla f(x_k)$ by finite-difference
  3. Define search direction $d_k = -\nabla f(x_k)$
  4. Line search : find ρ such as $f(x_k + \rho d_k) < f(x_k)$

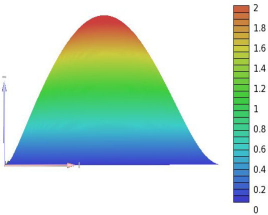
Example 1: given...

- source term : $\boldsymbol{f}(x,y) = -\frac{4\pi^2}{9}\sin(\frac{\pi x}{3})\sin(\frac{\pi y}{3})$.
- boundary condition : $\boldsymbol{U}_0(\boldsymbol{x}) = 0$ and $\Phi_0(\boldsymbol{x}) = 0$
- exact solution : $\boldsymbol{U}(x,y) = 2\sin(\frac{\pi x}{3})\sin(\frac{\pi y}{3})$.
- computational domain : $\Omega(x,y) = [0,3]\times[0,3]$

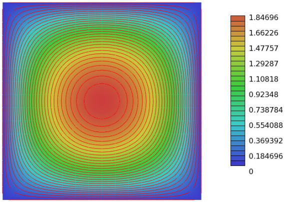
Example 1: results

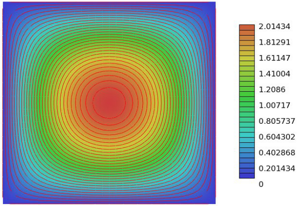


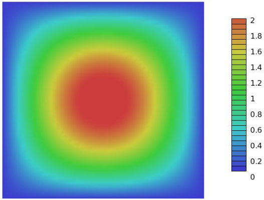




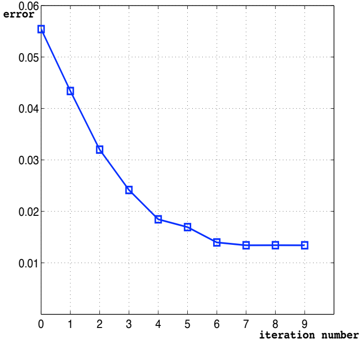
initial solution | final solution | exact solution

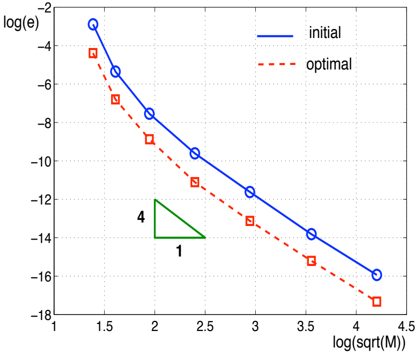






Example 1: error analysis





error history of optimization(24.52%)

error history of h-refinement

Example 2: given...

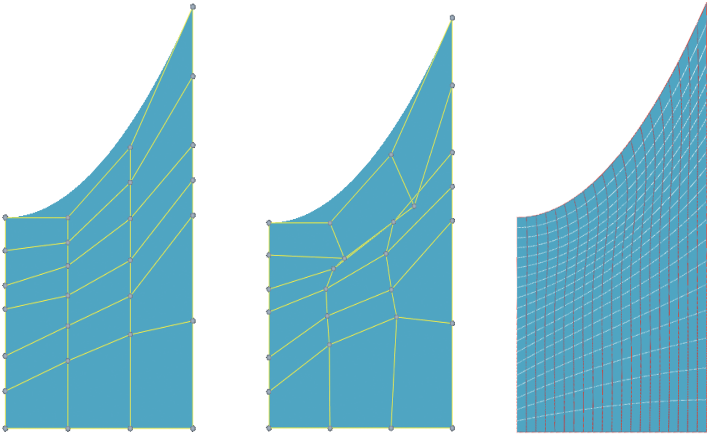
- boundary condition : $\boldsymbol{U}_0(\boldsymbol{x}) = 0$ and $\Phi_0(\boldsymbol{x}) = 0$
- exact solution :

$$\boldsymbol{U}(x, y) = \sin(\pi(y - x^2)) \sin(\pi x) \sin(\pi y)$$

- computational domain :

$$\Omega(x, y) = \{(x, y) | -1 \leq y \leq x^2, 0 \leq x \leq 1\}$$

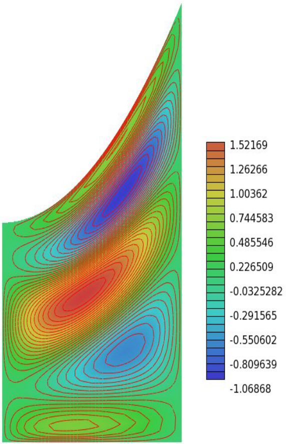
Example 2: results



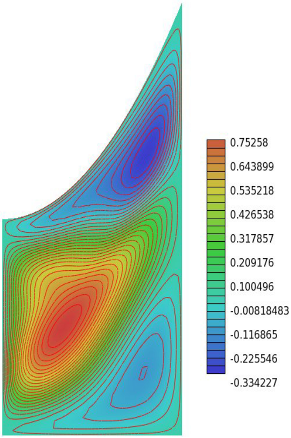
initial domain   final domain   final isoparametric net

Example 2: simulation results

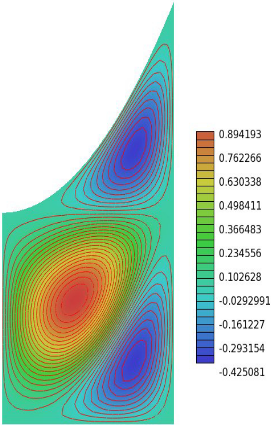
initial solution



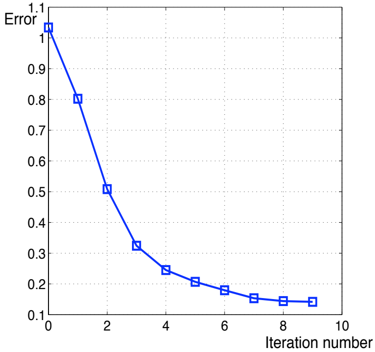
final solution

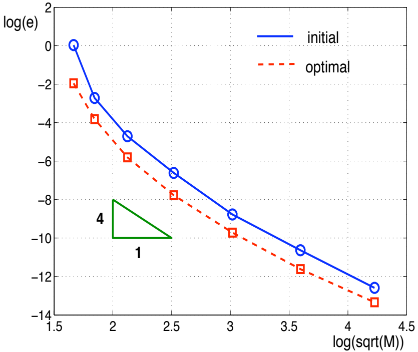


exact solution



Example 2: error analysis





error history of optimization(14.65%)

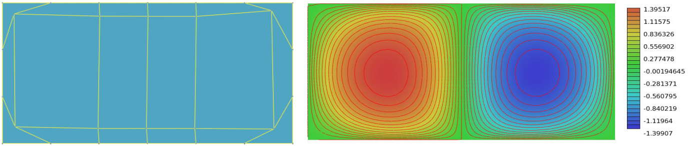
error history of h-refinement

Example 3: given...

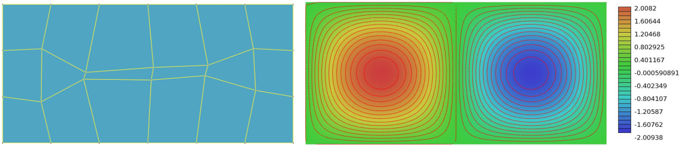
- source term : $\boldsymbol{f}(x, y) = -\frac{4\pi^2}{9} \sin(\frac{\pi x}{3}) \sin(\frac{\pi y}{3})$.
- boundary condition : $\boldsymbol{U}_0(\boldsymbol{x}) = 0$ and $\Phi_0(\boldsymbol{x}) = 0$
- exact solution : $\boldsymbol{U}(x, y) = 2 \sin(\frac{\pi x}{3}) \sin(\frac{\pi y}{3})$.
- computational domain : $\Omega(x, y) = [0, 6] \times [0, 3]$

Example 3: results

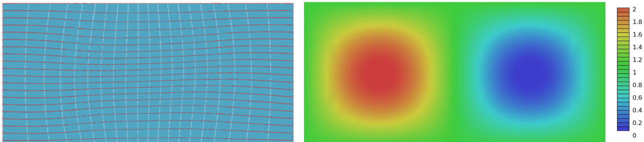
initial:



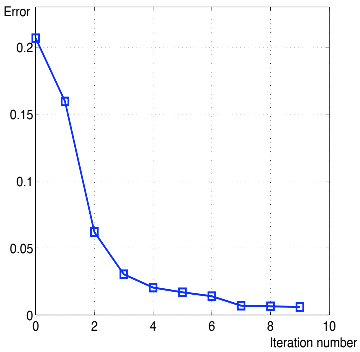
final:

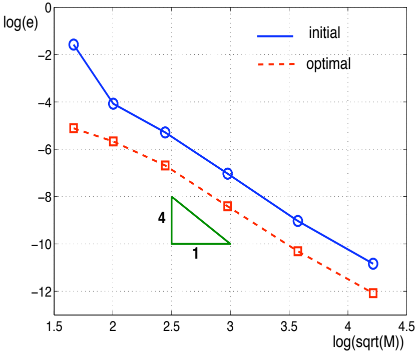


exact:



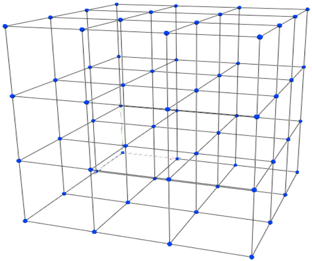
Example 3: error analysis



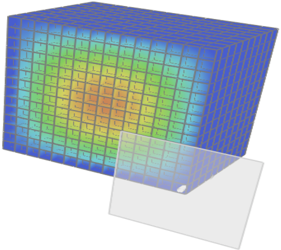


error history of optimization(3.31%) error history of h-refinement

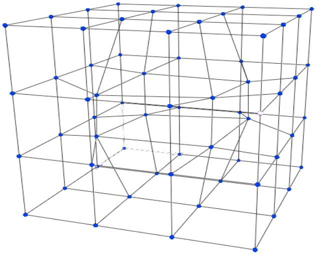
3D Example with known exact solution



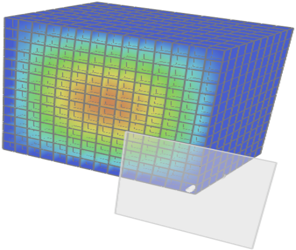
initial CD



initial solution



final CD



final solution

Outline

Outline

Residual function

- model problem:

$$\begin{aligned} \Delta U &= f \quad \text{in } \Omega \\ U &= U_0 \text{ on } \partial\Omega_D \end{aligned} \tag{2}$$

- U_h is the IGA solution
- error: $e = U - U_h$
- residual function

$$\begin{aligned} \mathcal{R}(\psi) &= \int_\Omega f\ \psi\ d\Omega + \int_\Omega \nabla U_h\ \nabla\psi\ d\Omega \\ &= \sum_{K\in\Omega} \int_K (f\ \psi - \Delta U_h\ \psi)\ dK \end{aligned}$$

A posteriori error estimate

$$\|e\|^2 \leq C \sum_{K \in \Omega} h_K \int_K (f - \Delta U_h)^2 dK \qquad (3)$$

Main idea for r-refinement

reposition inner control points to minimize $\sum_{K\in\Omega} h_K \int_K (f - \Delta U_h)^2 dK$.

Key point

- $$\Delta U_h = \frac{\partial^2 U_h}{\partial^2 x} + \frac{\partial^2 U_h}{\partial^2 y} \tag{4}$$
- Solution field in isogeometric analysis

$$U_h(x, y) = \mathcal{T}_h(\xi, \eta) = \sum_{i=1}^{n_i} \sum_{j=1}^{n_j} \hat{N}_i^{p_i}(\xi)\, \hat{N}_j^{p_j}(\eta)\, T_{ij}$$

Key point

How to compute ΔU_h?

Main idea of IGA

- Use the same mathematical representation for the computational domain and solution field.
- Computational domain Ω is parameterized by the following planar B-spline surface:

$$P(\xi,\eta) = (x(\xi,\eta), y(\xi,\eta)) = \sum_{i=1}^{n_i} \sum_{j=1}^{n_j} N_i^{d_i}(\xi)\, N_j^{d_j}(\eta)\, p_{ij},$$

- Solution field over the computational domain Ω has the following form,

$$U_h(x(\xi,\eta), y(\xi,\eta)) = \mathcal{T}(\xi,\eta) = \sum_{i=1}^{n_i} \sum_{j=1}^{n_j} N_i^{d_i}(\xi)\, N_j^{d_j}(\eta)\, T_{ij},$$

  Here T_{ij} are the unknown variables in isogeometric analysis to be solved.

Computation of ΔU_h

From $U_h(x(\xi,\eta), y(\xi,\eta)) = \mathcal{T}(\xi,\eta)$, we have

$$\frac{\partial \mathcal{T}}{\partial \xi} = \frac{\partial U_h}{\partial x}\frac{\partial x}{\partial \xi} + \frac{\partial U_h}{\partial y}\frac{\partial y}{\partial \xi} \tag{5}$$

$$\frac{\partial \mathcal{T}}{\partial \eta} = \frac{\partial U_h}{\partial x}\frac{\partial x}{\partial \eta} + \frac{\partial U_h}{\partial y}\frac{\partial y}{\partial \eta} \tag{6}$$

Then we can obtain

$$\frac{\partial U_h}{\partial x} = \left(\frac{\partial \mathcal{T}}{\partial \xi} y_\eta - \frac{\partial \mathcal{T}}{\partial \eta} y_\xi\right)/J \tag{7}$$

$$\frac{\partial U_h}{\partial y} = \left(\frac{\partial \mathcal{T}}{\partial \eta} x_\xi - \frac{\partial \mathcal{T}}{\partial \xi} x_\eta\right)/J \tag{8}$$

where $J = x_\xi y_\eta - y_\xi x_\eta$.

Computation of ΔU_h — continued

$$\frac{\partial^2 \mathcal{T}}{\partial^2 \xi} = \frac{\partial^2 U_h}{\partial^2 x}(\frac{\partial x}{\partial \xi})^2 + \frac{\partial U_h}{\partial x}\frac{\partial^2 x}{\partial^2 \xi} + \frac{\partial^2 U_h}{\partial^2 y}(\frac{\partial y}{\partial \xi})^2 + \frac{\partial U_h}{\partial y}\frac{\partial^2 y}{\partial^2 \xi} \quad (9)$$

$$\frac{\partial^2 \mathcal{T}}{\partial^2 \eta} = \frac{\partial^2 U_h}{\partial^2 x}(\frac{\partial x}{\partial \eta})^2 + \frac{\partial U_h}{\partial x}\frac{\partial^2 x}{\partial^2 \eta} + \frac{\partial^2 U_h}{\partial^2 y}(\frac{\partial y}{\partial \eta})^2 + \frac{\partial U_h}{\partial y}\frac{\partial^2 y}{\partial^2 \eta} \quad (10)$$

From (9)(10)

$$\frac{\partial^2 U_h}{\partial^2 x} = [y_\eta^2(\mathcal{T}_{\xi\xi} - \frac{\partial U_h}{\partial x}x_{\xi\xi} - \frac{\partial U_h}{\partial y}y_{\xi\xi}) - y_\xi^2(\mathcal{T}_{\eta\eta} - \frac{\partial U_h}{\partial x}x_{\eta\eta} - \frac{\partial U_h}{\partial y}y_{\eta\eta})]/K \quad (11)$$

$$\frac{\partial^2 U_h}{\partial^2 y} = [x_\xi^2(\mathcal{T}_{\eta\eta} - \frac{\partial U_h}{\partial x}x_{\eta\eta} - \frac{\partial U_h}{\partial y}y_{\eta\eta}) - x_\eta^2(\mathcal{T}_{\xi\xi} - \frac{\partial U_h}{\partial x}x_{\xi\xi} - \frac{\partial U_h}{\partial y}y_{\xi\xi})]/K \quad (12)$$

where $K = (x_\xi y_\eta)^2 - (x_\eta y_\xi)^2$

$$\Delta U_h = [(x_\xi^2 - y_\xi^2)(\mathcal{T}_{\eta\eta} - \frac{\partial U_h}{\partial x}x_{\eta\eta} - \frac{\partial U_h}{\partial y}y_{\eta\eta}) - (x_\eta^2 - y_\eta^2)(\mathcal{T}_{\xi\xi} - \frac{\partial U_h}{\partial x}x_{\xi\xi} - \frac{\partial U_h}{\partial y}y_{\xi\xi})]/K$$

Outline

Overview of r-refinement in IGA

1. Solve IGA problem over given computational domain
2. Compute $\sum_{K\in\Omega} h_K \int_K (f - \Delta U_h)^2 dK$
3. Reposition inner control points by minimizing $\sum_{K\in\Omega} h_K \int_K (f - \Delta U_h)^2 dK$
4. Output final placement of inner control points

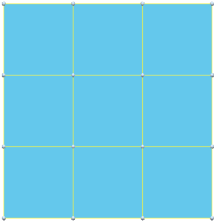
Error assessment

- $e = U - U_h$
- A posteriori error assessment by resolving IGA problem:

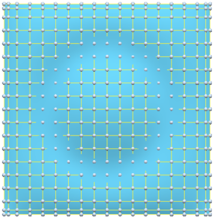
$$\begin{aligned} \Delta e &= f - \Delta U_h \quad &\text{in } \Omega \\ e &= 0 \quad &\text{on } \partial\Omega_D \end{aligned} \tag{13}$$

- Error field e has a B-spline form
- Perform h-refinement to achieve a good approximation
- More accurate but much more expensive
- Used for error assessment in r-refinement method

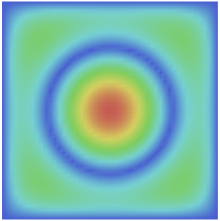
Error assessment: an example



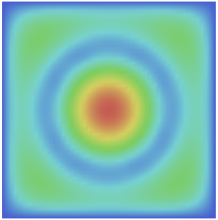
computational domain (CD)



resolved error surface



exact error colormap (EC)



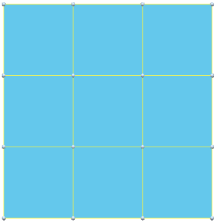
resolved EC

Outline

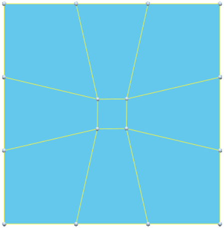
Model problem

$$\begin{aligned} \Delta U &= -\frac{4\pi^2}{9}\sin(\frac{\pi x}{3})\sin(\frac{\pi y}{3}) \quad \text{in } \Omega \\ U &= 0 \quad \text{on } \partial\Omega_D \end{aligned} \tag{14}$$

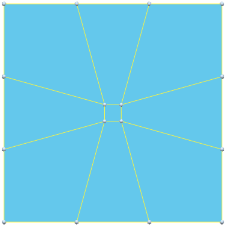
Example with known exact solution



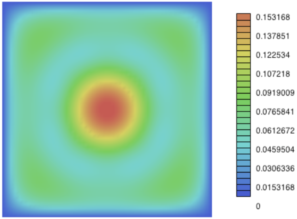
initial CD



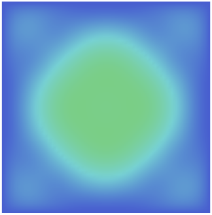
CD after r-refinement



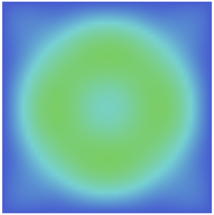
CD after exact r-refinement



initial EC



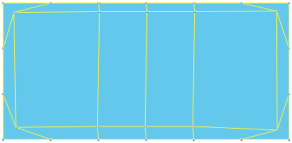
EC after r-refinement

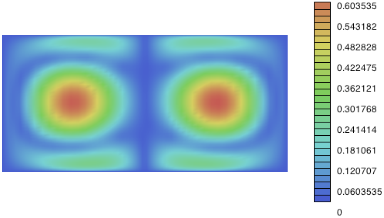


EC after exact method

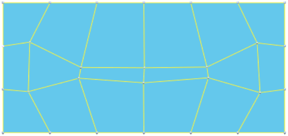
Example II with known exact solution

Initial CD and EC:

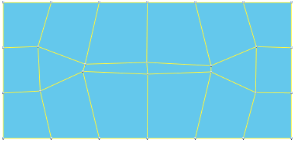


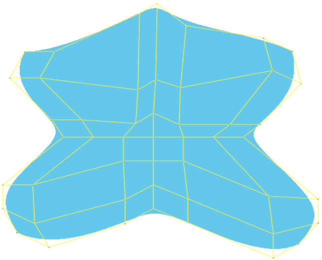


Final CD and EC:

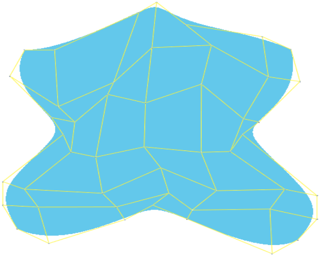


Exact method:

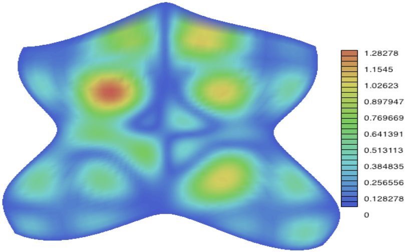




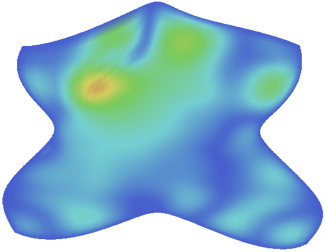
initial CD



CD after r-refinement

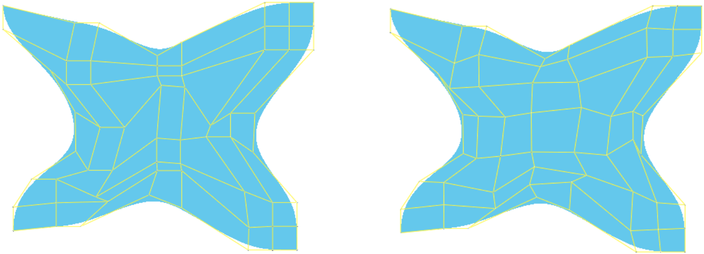


initial EC

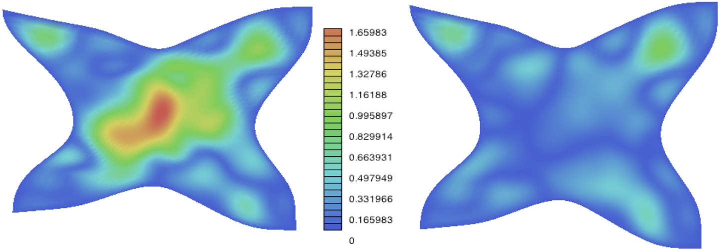


EC after r-refinement

Example IV with unknown exact solution



initial CD | CD after r-refinement



initial EC | EC after r-refinement

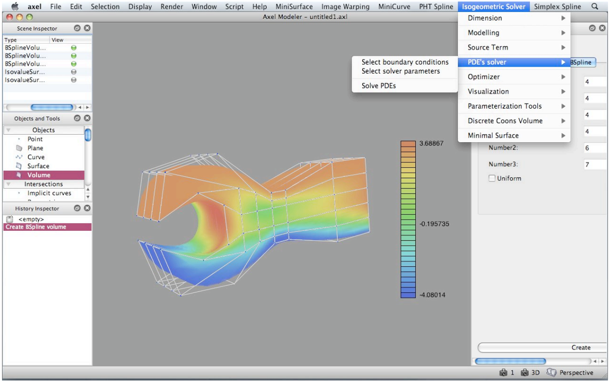
Outline

Conclusion

- Model quality has some effect on accuracy and efficiency
- r-refinement method in isogeometric version: reducing the error value while keeping number of freedom as constant
- Parameterization of computational domain is changed without changing the boundary geometry
- Residual-based a posteriori error estimation for IGA
- Error assessment method for r-refinement in IGA
- Implementation in algebraic-geometric modeler AXEL
  http://axel.inria.fr/

Future work

- Local r-refinement: reparameterization of patches where error value exceeds a specified tolerance.
- r-h-refinement for isogeometric analysis
- 3D cases:



Thanks for your attention!

Question?