智能可视建模与仿真实验室

Intelligent Visual Modeling & Simulation (iGame) Lab

第七讲
基于等几何分析的线弹性问题求解及GIFT方法

杭州电子科技大学

徐岗

1. 线弹性问题方程及其弱形式

线弹性问题是一个有着丰富研究历史的经典目标，使用IGA方法可以很好的求解。IGA求解此问题的优势不仅仅在于精确的几何描述，也因为使用了C^1连续的基函数得到了高质量的应力场。本次研究二维线弹性问题并求其平衡解，仅涉及平面。

- 在下文中，索引A,B,C,D可以取值1,…,d(在二维中d=2)，其中u_A指的是$\boldsymbol{u}$的第A个元素，并且其微分将用逗号表示，如：$u_{A,B} = u_{A,x_B} = \frac{\partial u_A}{\partial x_B}$
- 用重复的下标表示求和，如：$u_{A,BB} = u_{A,11} + u_{A,22}$
- 在一般的非对称张量A=$[A_{AB}]$中，使用$A_{(AB)}$表示对称部分，使用$A_{[AB]}$表示反对称部分。因此

  $A_{AB} = A_{(AB)} + A_{[AB]}$; $A_{(AB)} = A_{(BA)} = \frac{1}{2}(A_{(AB)} + A_{(BA)})$； $A_{[AB]} = -A_{[BA]} = \frac{1}{2}(A_{[AB]} - A_{[BA]})$
- 如果A是非对称张量并且B是对称张量，则$A_{\mathrm{ij}}B_{\mathrm{ij}} = A_{(\mathrm{ij})}B_{\mathrm{ij}}$; $A_{[\mathrm{ij}\}}B_{\mathrm{ij}} = 0$.
- σ_{AB}表示（Cauchy）应力张量的元素，u_A表示位移向量的元素， f_A表示每次单位体积体力的元素，$\varepsilon = [\varepsilon_{AB}]$表示小形变应力张量（工程应变），则$\varepsilon_{AB} = u_{(A,B)} = \frac{1}{2}(u_{A,B} + u_{B,A})$;
- 此问题的本构方程满足胡克定理，即$\sigma_{AB} = C_{ABCD}\varepsilon_{CD}$, 其中C为弹性材料系数矩阵；

1. 线弹性问题方程及其弱形式

下面给出线弹性混合边界值问题的**强形式**：

$$
(S)\quad \begin{cases}
\text{Find } \vec{u}: \bar{\Omega} \to R^{\mathrm{d}} \;\; s,t \\
\sigma_{AB} + \mathrm{f}_A = 0 \quad \text{in } \Omega \qquad A = 1,\ldots,d \\
u_A = g_A \quad \text{on } \Gamma_{D_A} \quad A = 1,\ldots,d \\
\sigma_{AB}\, n_B = h_A \quad \text{on } \Gamma_{N_A} \quad A = 1,\ldots,d
\end{cases}
$$

g_A为给定的边界位移

h_A为给定的边界traction

Γ_{D_A}为Dirichlet边界

Γ_{N_A}为Neuman边界

其中 $\overline{I^2_{D_A} \cup I^2_{N_A}} \equiv I^2 \equiv \partial\Omega$ ，且 $\Gamma_{D_A} \cap \Gamma_{N_A} = \emptyset$

1. 线弹性问题方程及其弱形式

为得到线弹性问题的弱形式，需要给每个方向设置一组trial函数和weighting函数：

$$\Phi_A := \{u_A : \bar{\Omega} \to R | u_A \in H^1(\Omega)\ and\ u_{A|\Gamma_{D_A}} = g_A\}$$

$$\Upsilon_A := \{w_A : \bar{\Omega} \to R | w_A \in H^1(\Omega)\ and\ w_{A|\Gamma_{D_A}} = 0\ \}$$

通过乘以w函数并使用分步积分，得到线弹性混合边界值问题的**强形式**：

$$\textbf{(w)}\begin{cases} \text{Find } u_A \in \Phi_A \quad A = 1,\ldots,d,\ \text{s.t.} \\ \displaystyle\int_\Omega w_{(A,B)}\,\sigma_{AB}\,d\Omega = \int_\Omega w_A\,f_A\,d\Omega + \sum_{A=1}^{d}\left(\int_{\Gamma_{N_A}} w_A\,h_A\,d\Gamma\right) \quad \forall w_A \in \Upsilon_A \quad A = 1,\ldots,d \end{cases}$$

1. 线弹性问题方程及其弱形式

为了方便，将(w)式变为简写，定义一下符号：

$$\Phi := \{\vec{u}: u_A \in \Phi_A\}$$

$$\Upsilon := \{\vec{w}: w_A \in \Upsilon_A\}$$

$$a(\vec{u},\vec{w}) = \int_{\Omega} w_{(A,B)}\, C_{ABCD} u_{(C,D)} d\Omega$$

$$L(\vec{w}) = \int_{\Omega} w_A f_A d\Omega + \sum_{A=1}^{d} \left(\int_{\Gamma_{N_A}} w_A h_A d\Gamma \right)$$

将上述符号代入(w)，得简写形式：

$$\textbf{(w)} \begin{cases} \text{Find } \vec{u} \in \Phi \text{ s.t.} \\ \quad a(\vec{u},\vec{w}) = L(\vec{w}) \quad \forall \vec{w} \in \Upsilon \end{cases}$$

解释：在固体力学中，(w)被称作虚功原理，$\vec{w}$被称为虚位移。L(w)是体力f乘以虚位移，再加上边界traction得到的虚功，$a(\vec{u},\vec{w})$代表的是应力做的功。(w)描述了静态的内力功和外力功的平衡。

2. Voigt Notation

将对称张量写作向量可以简化之后的形式，也可以降低其阶数，这种方法称为Voigt Notation。

对于线弹性问题：

- 应变向量：$\vec{\varepsilon}(\vec{u}) := [u_{1,1} \quad u_{2,2} \quad u_{1,2} + u_{2,1}]^T$
- 虚应变向量：$\vec{\varepsilon}(\vec{w}) := [w_{1,1} \quad w_{2,2} \quad w_{1,2} + w_{2,1}]^T$
- 应力向量：$\vec{\sigma}(\vec{u}) := [\sigma_{11} \quad \sigma_{22} \quad \sigma_{12}]^T$
- 弹性系数矩阵：$\vec{D} := \begin{bmatrix} D_{11} & D_{12} & D_{13} \\ D_{21} & D_{22} & D_{23} \\ D_{31} & D_{32} & D_{33} \end{bmatrix}$，其中$D_{IJ} = C_{ABCD}$

I/J	A/C	B/D
1	1	1
2	2	2
3	1	2
3	2	1

已知 $\vec{\sigma}(\vec{u}) = \vec{D}\,\vec{\varepsilon}(\vec{u})$，可得：

$$a(\vec{u}, \vec{w}) = \int_{\Omega} (\vec{\varepsilon}(\vec{w}))^T D\, \vec{\varepsilon}(\vec{u}) d\Omega$$

3. 各向同性体，平面应变和应力

对于各向同性体（Isotropic bodies）,有如下关系式：

$$C_{ABCD}(\vec{x}) = \mu(\vec{x})(\delta_{AC}\delta_{BD} + \delta_{AD}\delta_{BC}) + \lambda(\vec{x})\delta_{AB}\delta_{CD} \qquad A, B, C, D = 1, \dots, 3$$

其中λ和μ是拉梅常数，可以使用如下公式转化为杨氏模量E和泊松系数v：

$$\lambda = \frac{\mathrm{v}E}{(1+\mathrm{v})(1-2v)}, \qquad \mu = \frac{E}{2(1+\mathrm{v})}$$

所以此时的应力和应变有如下关系式：

$$\begin{bmatrix} \sigma_{11} \\ \sigma_{22} \\ \sigma_{33} \\ \sigma_{23} \\ \sigma_{13} \\ \sigma_{12} \end{bmatrix} = \begin{bmatrix} 2\mu+\lambda & \lambda & \lambda & 0 & 0 & 0 \\ \lambda & 2\mu+\lambda & \lambda & 0 & 0 & 0 \\ \lambda & \lambda & 2\mu+\lambda & 0 & 0 & 0 \\ 0 & 0 & 0 & \mu & 0 & 0 \\ 0 & 0 & 0 & 0 & \mu & 0 \\ 0 & 0 & 0 & 0 & 0 & \mu \end{bmatrix} \begin{bmatrix} \varepsilon_{11} \\ \varepsilon_{22} \\ \varepsilon_{33} \\ 2\varepsilon_{23} \\ 2\varepsilon_{13} \\ 2\varepsilon_{12} \end{bmatrix}$$

3. 各向同性体，平面应变和应力

此时的应力和应变的关系式也可以是：

$$\begin{bmatrix} \varepsilon_{11} \\ \varepsilon_{22} \\ \varepsilon_{33} \\ 2\varepsilon_{23} \\ 2\varepsilon_{13} \\ 2\varepsilon_{12} \end{bmatrix} = \frac{1}{E} \begin{bmatrix} 1 & -\nu & -\nu & 0 & 0 & 0 \\ -\nu & 1 & -\nu & 0 & 0 & 0 \\ -\nu & -\nu & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 2(1+\nu) & 0 & 0 \\ 0 & 0 & 0 & 0 & 2(1+\nu) & 0 \\ 0 & 0 & 0 & 0 & 0 & 2(1+\nu) \end{bmatrix} \begin{bmatrix} \sigma_{11} \\ \sigma_{22} \\ \sigma_{33} \\ \sigma_{23} \\ \sigma_{13} \\ \sigma_{12} \end{bmatrix}$$

对于二维的应力，要忽略z方向的应变值，即$\varepsilon_{33} = \varepsilon_{23} = \varepsilon_{13} = 0$，可得如下式：

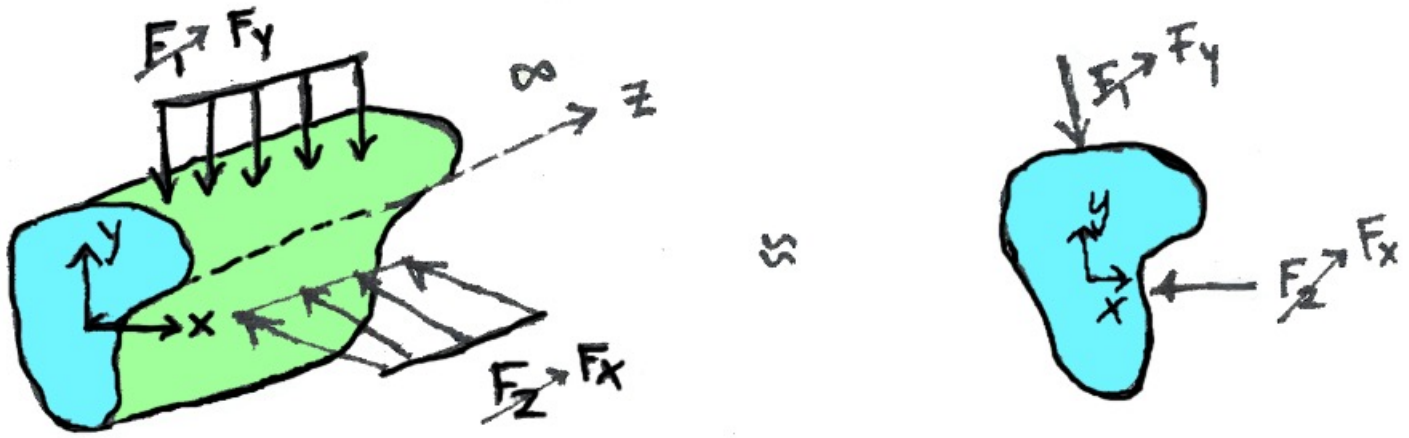
$$\begin{bmatrix} \sigma_{11} \\ \sigma_{22} \\ \sigma_{12} \end{bmatrix} = \begin{bmatrix} 2\mu+\lambda & \lambda & 0 \\ \lambda & 2\mu+\lambda & 0 \\ 0 & 0 & \mu \end{bmatrix} \begin{bmatrix} \varepsilon_{11} \\ \varepsilon_{22} \\ 2\varepsilon_{12} \end{bmatrix}$$

因此，对于平面应力，有如下式：

$$\vec{\sigma}(\vec{u}) = \vec{D}\,\vec{\varepsilon}(\vec{u})\text{，其中}\vec{D} = \begin{bmatrix} 2\mu+\lambda & \lambda & 0 \\ \lambda & 2\mu+\lambda & 0 \\ 0 & 0 & \mu \end{bmatrix}$$

3. 各向同性体，平面应变和应力

当z方向的尺寸与其他方向的尺寸相比大的多时，会发生平面应变。在这种情况下，z方向上的主应变被约束并假设为零。



平面应变假设常用于大坝、漏斗和其他岩土工程的分析。

3. 各向同性体，平面应变和应力

同样对于平面应变也可以得出相同的结论，忽略z方向的应力，即$\sigma_{33} = \sigma_{23} = \sigma_{13} = 0$

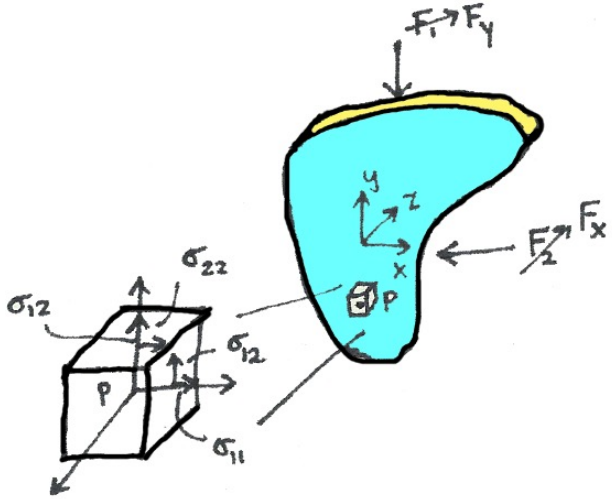
$$\begin{bmatrix} \varepsilon_{11} \\ \varepsilon_{22} \\ 2\varepsilon_{12} \end{bmatrix} = \frac{1}{E} \begin{bmatrix} 1 & -v & 0 \\ -v & 1 & 0 \\ 0 & 0 & 2(1+v) \end{bmatrix} \begin{bmatrix} \sigma_{11} \\ \sigma_{22} \\ \sigma_{12} \end{bmatrix}$$

对上式逆置得：

$$\begin{bmatrix} \sigma_{11} \\ \sigma_{22} \\ \sigma_{12} \end{bmatrix} = \begin{bmatrix} E/(1-v^2) & vE/(1-v^2) & 0 \\ vE/(1-v^2) & E/(1-v^2) & 0 \\ 0 & 0 & E/2(1+v) \end{bmatrix} \begin{bmatrix} \varepsilon_{11} \\ \varepsilon_{22} \\ 2\varepsilon_{12} \end{bmatrix}$$

因此，对于平面应变有：

$$\vec{\sigma}(\vec{u}) = \vec{D}\,\vec{\varepsilon}(\vec{u})\text{，其中}\vec{D} = \frac{E}{1-v^2} \begin{bmatrix} 1 & v & 0 \\ v & 1 & 0 \\ 0 & 0 & (1-v)/2 \end{bmatrix}$$



平面应力通常发生在薄平板上，这些平板只受到与其平行的载荷力作用。

3. 各向同性体，平面应变和应力

平面应变和平面应力是二维弹性的经典设定，这也引出了弹性的平面理论，下面是对之前的总结。

平面弹性理论：

应力-应变关系： $\vec{\sigma}(\vec{u}) = \vec{D}\,\vec{\varepsilon}(\vec{u})$ （其中$D_{IJ} = C_{ABCD}$ ）

各向同性体的平面应变：

$$\vec{D} = \begin{bmatrix} 2\mu+\lambda & \lambda & 0 \\ \lambda & 2\mu+\lambda & 0 \\ 0 & 0 & \mu \end{bmatrix}$$

平面之外的法向应力：

$$\sigma_{33} = \lambda(\varepsilon_{11} + \varepsilon_{22})$$

各向同性体的平面应力：

$$\vec{D} = \frac{E}{1-v^2}\begin{bmatrix} 1 & v & 0 \\ v & 1 & 0 \\ 0 & 0 & (1-v)/2 \end{bmatrix}$$

平面之外的法向应变：

$$\varepsilon_{33} = -\frac{v}{E}(\sigma_{11} + \sigma_{22})$$

4. Galerkin方法

（1）为了将线性弹性问题的弱形式转变为代数方程组，可以使用Galerkin方法并且在有限集$\vartheta^{\mathrm{h}} \subset \vartheta$和$\Upsilon^{\mathrm{h}} \subset \Upsilon$中计算，这些集合由等参的NURBS基定义。

$$\Upsilon^{\mathrm{h}} := \{\vec{w}^h \in \Upsilon : \vec{w}^h(\vec{x}) = \sum_{i=1}^{n} N_i(\vec{x})\overrightarrow{c_i}\}$$

$$\vartheta^{\mathrm{h}} := \{\vec{u}^h \in \vartheta : \vec{u}^h(\vec{x}) = \sum_{i=1}^{n} N_i(\vec{x})\overrightarrow{d_i}\}$$

（2）进一步描述ϑ^{h}，要求如果一个给定函数$\vec{g}^h \in \vartheta^{\mathrm{h}}$ (其中$(\vec{g}^h)_{A|\Gamma_{D_A}} = g_A$ for A = 1,…,d)，对于每一个$\vec{u}^h \in \vartheta^{\mathrm{h}}$都存在与之对应的$\vec{v}^h \in \Upsilon^{\mathrm{h}}$满足：

$$\vec{u}^h = \vec{v}^h + \vec{g}^h$$

将此方法应用到线弹性上可得：

$$\textbf{(G)} \begin{cases} \text{Find } \vec{u}^h = \vec{v}^h + \vec{g}^h, \text{ where } \vec{v}^h \in \Upsilon^{\mathrm{h}} \text{ s.t.} \\ a(\vec{w}^h, \vec{u}^h) = L(\vec{w}^h) \quad \forall \vec{w}^h \in \Upsilon^{\mathrm{h}} \end{cases}$$

4. Galerkin方法

然后可以根据定义将上式中已知量和未知量分离：

$$a(\vec{w}^h, \vec{v}^h) = L(\vec{w}^h) + a(\vec{w}^h, \vec{g}^h)$$

（3）定义$\eta = \{1, \dots, n\}$是包含所有NURBS基函数索引的集合，并且定义$\eta_{g_A} \subset \eta$ 是包含所有在Γ_{D_A}上非0的NURBS基函数索引的集合，可得：

$$supp(N_i) \cap \Gamma_{D_A} \neq \emptyset \quad if\ and\ only\ if \quad i \in \eta_{g_A}$$

$$\Upsilon^{\mathrm{h}} := \{\vec{w}^h : (\vec{w}^h(\vec{x}))_A = \sum_{i \in \eta - \eta_{g_A}} N_i(\vec{x})(\vec{c_i})_A \quad , \qquad A = 1, \dots, d\}$$

选择$\vec{g}^h(\vec{x})$，其满足$(\vec{g}_i)_A = 0 \quad for\ i \in \eta - \eta_{g_A}$：

$$(\vec{g}^h(\vec{x}))_A = \sum_{i \in \eta_{g_A}} N_i(\vec{x})(\vec{g_i})_A$$

然后，可以写出$\vec{u}^h \in \vartheta^{\mathrm{h}}$的第A个部分：

$$(\vec{u}^h(\vec{x}))_A = \sum_{i \in \eta} N_i(\vec{x})(\vec{d_i})_A = \sum_{i \in \eta - \eta_{g_A}} N_i(\vec{x})(\vec{d_i})_A + \sum_{i \in \eta_{g_A}} N_i(\vec{x})(\vec{g_i})_A$$

4. Galerkin方法

（4）如果将$\vec{u}^h$和$\vec{w}^h$的表达式插入到Galerkin公式中，可以得到：

$$a\left(\sum_{A=1}^{d}\sum_{i\in\eta-\eta_{g_A}} N_i\,(\overrightarrow{c_i})_A\hat{e}_A\,,\ \sum_{B=1}^{d}\sum_{j\in\eta-\eta_{g_B}} N_j\,(\overrightarrow{d_j})_B\hat{e}_B\right)$$

$$= L\left(\sum_{A=1}^{d}\sum_{i\in\eta-\eta_{g_A}} N_i\,(\overrightarrow{c_i})_A\hat{e}_A\right) - a\left(\sum_{A=1}^{d}\sum_{i\in\eta-\eta_{g_A}} N_i\,(\overrightarrow{c_i})_A\hat{e}_A\,,\ \vec{g}^h\right)$$

在二维中， $\hat{e}_1 = \left[\begin{smallmatrix}1\\0\end{smallmatrix}\right], \hat{e}_2 = \left[\begin{smallmatrix}0\\1\end{smallmatrix}\right]$

（5）使用a()的线性，可以得：

$$\sum_{A=1}^{d}\sum_{i\in\eta-\eta_{g_A}} (\overrightarrow{c_i})_A\Big(\sum_{B=1}^{d}\sum_{j\in\eta-\eta_{g_B}} a(N_i\,\hat{e}_A\,, N_j\hat{e}_B)(\overrightarrow{d_j})_B - L(N_i\hat{e}_A) + a(N_i\hat{e}_A, \vec{g}^h) = 0$$

对于任意$(\overrightarrow{c_i})_A$, $i \in \eta - \eta_{g_A}$, A=1,…,d, 可以得到：

$$\sum_{B=1}^{d}\sum_{j\in\eta-\eta_{g_B}} a(N_i\,\hat{e}_A\,, N_j\hat{e}_B)(\overrightarrow{d_j})_B = L(N_i\hat{e}_A) - a(N_i\hat{e}_A, \vec{g}^h)$$

4. Galerkin方法

对于上面的n_{eq}个方程，共有n_{eq}个未知数，其中$n_{eq} = \sum_{A=1}^{d} |\eta - \eta_{g_A}|$.

（6）根据虚功原理，对于每一个全局基函数都有一个方程；而在这里，对于每一个全局基函数都有d个方程。因此，引入一个ID数组来表示自由度A=1,…,d和全局函数索引$i \in \eta - \eta_{g_A}$联系在一起的方程索引P：P = ID(A , i).

最终的矩阵形式方程：

$$\boldsymbol{Kd = F}$$

$$K = [K_{PQ}] = stiffness\ matrix$$

$$d = [d_Q] = displacement\ vector$$

$$F = [F_P] = force\ vector$$

其中： P = ID(A , i); Q = ID(B , j).

$$K_{PQ} = a(N_i \hat{e}_A, N_j \hat{e}_B)$$

$$F_P = L(N_i \hat{e}_A) - a(N_i \hat{e}_A, \vec{g}^h)$$

$$d_Q = (\overrightarrow{d_j})_B$$

5. 装配矩阵系统

根据虚功原理，构建了大量的局部刚度矩阵和局部载荷向量，将它们装配到全局刚度矩阵和全局载荷向量中，就得到了全局系统。在d维空间和n_{loc}个局部基函数的条件下，可以计算出单元Ω^{e}的局部刚度矩阵：

$$K_{\mathrm{pq}}^{e} = \int_{\Omega^{\mathrm{e}}} (\vec{\varepsilon}(N_a^e \hat{e}_A))^T \, D \, \vec{\varepsilon}(N_b^e \hat{e}_B) d\Omega^{\mathrm{e}}$$

其中：p=d(a-1)+A ; q=d(b-1)+B

将应变记作： $\vec{\varepsilon}(N_a^e \hat{e}_A) = B_a^e \hat{e}_A$，其中$B_a^e = \begin{bmatrix} N_{a,1}^e & 0 \\ 0 & N_{a,2}^e \\ N_{a,2}^e & N_{a,1}^e \end{bmatrix}$ for d=2

可以将局部刚度矩阵简写为：

$$\boldsymbol{K_{pq}^{e}} = \int_{\boldsymbol{\Omega^e}} (\boldsymbol{B_a^e})^{\boldsymbol{T}} \boldsymbol{D} (\boldsymbol{B_b^e}) \boldsymbol{d\Omega^e} \hat{\boldsymbol{e}}_{\boldsymbol{B}}$$

5. 装配矩阵系统

同样可以得出局部载荷向量的计算公式：

$$\boldsymbol{f_p^e} = \int_{\boldsymbol{\Omega^e}} \boldsymbol{N_a^e f_A d\Omega^e} + \int_{\boldsymbol{\Gamma_{N_A}} \cap \boldsymbol{\Gamma^e}} \boldsymbol{N_a^e h_A d\Gamma^e}$$

为了将局部刚度矩阵和载荷向量装配到全局系统中，需要用到下面的数组：

$$P = LM(A, a, e) = ID\big(A, IEN(a, e)\big)$$

其中P是全局公式索引，A是自由度，a是局部基索引，e是单元编号。

$$BC(A, \mathrm{i}) = \begin{cases} 0 & if\ N_i|_{\boldsymbol{\Gamma_{D_A}}} = 0 \\ 1 & otherwise \end{cases}$$

其中i是全局基索引，A是自由度。若$BC(A, \mathrm{i}) = 1$，设定：$K_{PQ} = \delta_{PQ}$， $F_P = (\vec{\mathrm{g}}_i)_A$。

其中P=ID(A , i),表示Dirichlet条件$(\vec{d}_i)_A = (\vec{\mathrm{g}}_i)_A$。当遇到Dirichlet边界条件约束时不需要更新全局刚度矩阵，只需要根据Dirichlet条件更新全局载荷向量。

5. 装配矩阵系统

根据上面的思路，可以写出单元的装配过程：

```
for A=1,…,d
    for a=1,..,n_loc
        set p=d(a-1)+A
        set P=LM(A,a,e)
        set i=IEN(a,e)
        if BC(A,i) = 0
            左边；
            F_P += f_p^e
        endif
    endloop
endloop
```

```
for B=1,…,d
    for b=1,…, n_loc
        set q=d(b-1)+B
        set Q=LM(B , b , e)
        set j=IEN(b , e)
        if BC(B,j)=0
            Update K_PQ += K_pq^e
        else
            Update F_P -= K_pq^e (g_i)_B
        endif
    endloop
endloop
```

6. 构造单元刚度矩阵和载荷向量

构建单元刚度矩阵和载荷向量的方法使用到了等参，Bezier extraction和四边形的概念。

（1）使用这种思想，<u>单元刚度矩阵</u>可用如下公式计算：

$$K_{pq}^{e} = \int_{\Omega^e} (\vec{\varepsilon}(N_a^e \hat{e}_A))^T D\, \vec{\varepsilon}(N_b^e \hat{e}_B) d\Omega^e = \hat{e}_A{}^T \int_{\Omega^e} (B_a^e)^T D (B_b^e) d\Omega^e \hat{e}_B$$

$$\cong \hat{e}_A{}^T \left(\sum_{q_1=1}^{n_q} \sum_{q_2=1}^{n_q} (B_a^e(\vec{x}^e(\tilde{\xi}_{q_1}, \tilde{\xi}_{q_2}))^T D \left(\vec{x}^e(\tilde{\xi}_{q_1}, \tilde{\xi}_{q_2})\right) \left(B_b^e\left(\vec{x}^e(\tilde{\xi}_{q_1}, \tilde{\xi}_{q_2})\right)\right) * \tilde{w}_{q_1} * \tilde{w}_{q_2} * jac(\tilde{\xi}_{q_1}, \tilde{\xi}_{q_2}) \right) \hat{e}_B$$

其中使用到了二维的高斯积分方法，进行了等参的空间转化。

6. 构造单元刚度矩阵和载荷向量

（2）构建单元载荷向量时将其分为两个部分：体积力（如重力、离心力）+表面力（如牵引力）

$$f_p^e = \int_{\Omega^e} N_a^e f_A d\Omega^e + \int_{\Gamma_{N_A}\cap\Gamma^e} N_a^e h_A d\Gamma^e = V + T$$

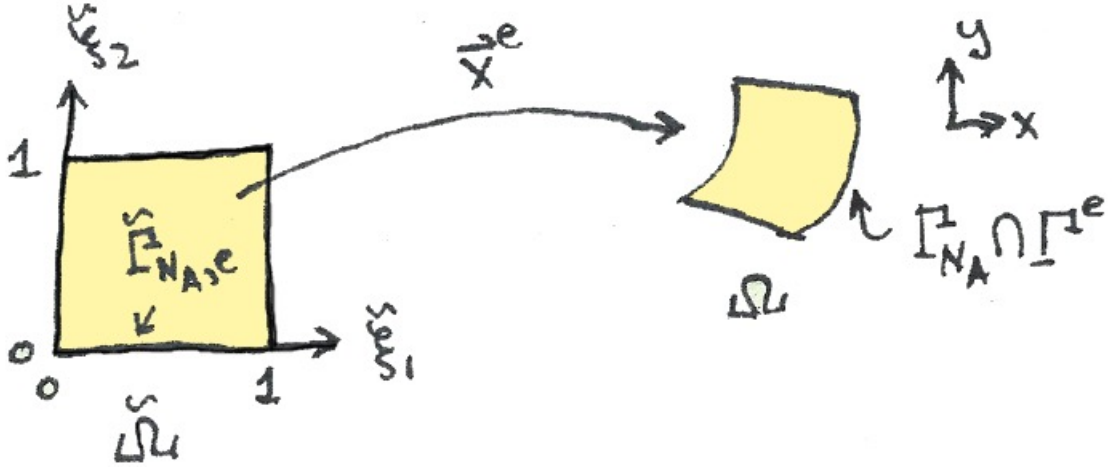
①构建体积力部分和构建刚度矩阵相同：

$$V \cong \sum_{q_1=1}^{n_q}\sum_{q_2=1}^{n_q} N_a^e\left(\vec{x}^e\left(\tilde{\xi}_{q_1},\tilde{\xi}_{q_2}\right)\right) * f_A\left(\vec{x}^e\left(\tilde{\xi}_{q_1},\tilde{\xi}_{q_2}\right)\right) * \tilde{w}_{q_1} * \tilde{w}_{q_2} * jac\left(\tilde{\xi}_{q_1},\tilde{\xi}_{q_2}\right)$$

②构建表面力部分，需要精确获取边界$\Gamma_{N_A}\cap\Gamma^e$，定义：

$$\tilde{\Gamma}_{N_A,e} := \vec{x}^{e^{-1}}\left(\Gamma_{N_A}\cap\Gamma^e\right) \quad (preimage\ of\ Neumann\ boundary)$$

假设： $\tilde{\Gamma}_{N_A,e} = \{\check{\xi} = (t,0): t \in (0,1)\}$



6. 构造单元刚度矩阵和载荷向量

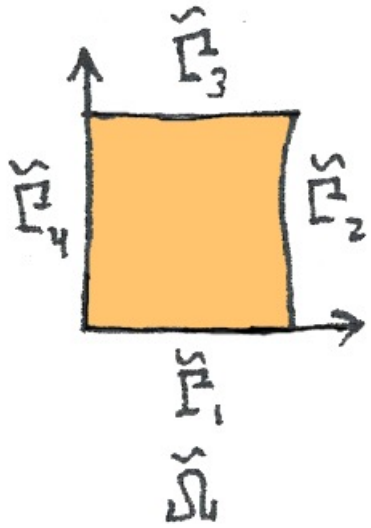
所以表面力构建公式近似为：

$$T \cong \sum_{q=1}^{n_q} N_a^e\left(\vec{x}^e\left(\tilde{\xi}_q, 0\right)\right) * h_A\left(\vec{x}^e\left(\tilde{\xi}_q, 0\right)\right) * \tilde{w}_q * jac_\Gamma\left(\tilde{\xi}_q, 0\right)$$

其中： $jac_\Gamma\left(\tilde{\xi}_q, 0\right) = \left|J\left(\tilde{\xi}_q, 0\right)\tilde{t}\right| = \left|J\left(\tilde{\xi}_q, 0\right)\begin{bmatrix}1\\0\end{bmatrix}\right|$

对于其他边的边界积分均为类似的表达，为了方便，创建一个Neumann数组标识Neumann边界对应的Bezier单元边：

$$Neumann(A, a, e) = \begin{cases} 1 & if \ \ \vec{x}^e\left(\tilde{\Gamma}_a\right) \subseteq \Gamma_{N_A} \\ 0 & other \end{cases}$$



6. 构造单元刚度矩阵和载荷向量

for q_1=1,…n_q

for q_2=1,…n_q

call shapeFunction(e, $\tilde{\xi}_{q_1}$,$\tilde{\xi}_{q_2}$)

compute R_a^e,∇R_a^e,$\vec{x}$,Jac(e, $\tilde{\xi}_{q_1}$,$\tilde{\xi}_{q_2}$)

set j=det(J)

set D_{loc} =D($\vec{x}$)

set f_{loc} =f($\vec{x}$)

for a=1,…, n_{loc}

Form $B_a^e = \begin{bmatrix} N_{a,1}^e & 0 \\ 0 & N_{a,2}^e \\ N_{a,2}^e & N_{a,1}^e \end{bmatrix}$

for b=1,…, n_{loc}

Form $B_b^e = \begin{bmatrix} N_{b,1}^e & 0 \\ 0 & N_{b,2}^e \\ N_{b,2}^e & N_{b,1}^e \end{bmatrix}$

set $K_{loc} = (B_a^e)^T D_{\text{loc}} (B_b^e) * \tilde{w}_{q_1} * \tilde{w}_{q_2} * j$

for A=1,…,D

for B=1,…,d

set p=d(a-1)+A

set p=d(b-1)+B

Update K_{pq}^e+= $\hat{e}_A{}^T K_{loc} \hat{e}_B$

endloop

endloop

endloop

for A=1,…,d

set p=d(a-1)+A

Update f_p^e+= R_a^e*$(f_{\text{loc}})_A$* $\tilde{w}_{q_1}$ * $\tilde{w}_{q_2}$ * j

endloop

endloop

endloop

endloop

总结

第一步：初始化问题

① Set p1,p2,n1,n2,节点向量1、2,控制点集,权重集

② 调用函数获得n_{el},$\{c^e\}_{e=1}^{n_{el}}$,IEN,$\{\{\vec{p}_b^{b,e}\}\}$, $\{\{w_a^{b,e}\}\}$, $\{\{\vec{p}_a^{a,e}\}\}$, $\{\{w_a^{a,e}\}\}$

③ Set n_q,$\{\check{\xi}_q\}_{q=1}^{n_q}$,$\{\check{w}_q\}_{q=1}^{n_q}$

④ 构建 ID和LN数组

⑤ 构建 g_A,A=1,…d (Dirichlet条件数据)

第二步：构建刚度矩阵

```
Set K=0 and F=0
for e=1,…, n_el
    调用单元计算函数得到k^e和f^e
    调用单元装配函数更新K和F
endloop
```

第2.1步：添加边界约束条件

```
for i=1,…,n
    for A=1,…,d
        if BC(A,i)=1
            set P=ID(A,i)
            set K_PP = 1
            set F_p = (g_i)_A
        endif
    endloop
endloop
```

第3步：求解矩阵系统

Solve d=K.inverse()*F

第4步：分析结果

ID数组的替代结构

ID数组定义如下：P=ID(A , i)　Q=ID(B , j)

其中P是方程编号，Q是未知量编号，A,B是自由度编号，i , j是基函数编号。

构建ID数组有两个明显的做法：（1）通过基函数索引将未知量组织起来；（2）使用自由度索引将未知量组织起来。

（1）通过基函数索引将未知量组织起来

$$\mathrm{d} = \begin{bmatrix} d_1 \\ d_2 \\ d_3 \\ d_4 \\ \vdots \\ d_{2n-1} \\ d_{2n} \end{bmatrix} = \begin{bmatrix} (\vec{d}_1) \\ (\vec{d}_2) \\ \vdots \\ (\vec{d}_n) \end{bmatrix} \qquad K = \begin{bmatrix} (\mathrm{k})_{11} & (\mathrm{k})_{12} & \cdots & (\mathrm{k})_{1n} \\ (\mathrm{k})_{21} & (\mathrm{k})_{22} & \cdots & (\mathrm{k})_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ (\mathrm{k})_{n1} & (\mathrm{k})_{n2} & \cdots & (\mathrm{k})_{nn} \end{bmatrix}$$

$$\Rightarrow \quad ID(B,j) = (j-1) * d + B$$

其中k_ij是和基函数i，j相关的d*d矩阵。做法1的刚度矩阵会比做法2的刚度矩阵更具有稀疏性。

ID数组的替代结构

（2）使用自由度索引将未知量组织起来

$$\mathrm{d} = \begin{bmatrix} d_1 \\ \vdots \\ d_{\mathrm{n}} \\ d_{n+1} \\ \vdots \\ d_{2n} \end{bmatrix} = \begin{bmatrix} (\vec{d}_1)_1 \\ \vdots \\ (\vec{d}_n)_1 \\ (\vec{d}_1)_1 \\ \vdots \\ (\vec{d}_n)_1 \end{bmatrix} \qquad K = \begin{bmatrix} K_{11} & K_{12} \\ K_{21} & K_{22} \end{bmatrix}$$

$$\Rightarrow \quad ID(B,j) = (B-1) * n + j$$

其中K_{AB}是和自由度A、B相关的n*n矩阵。

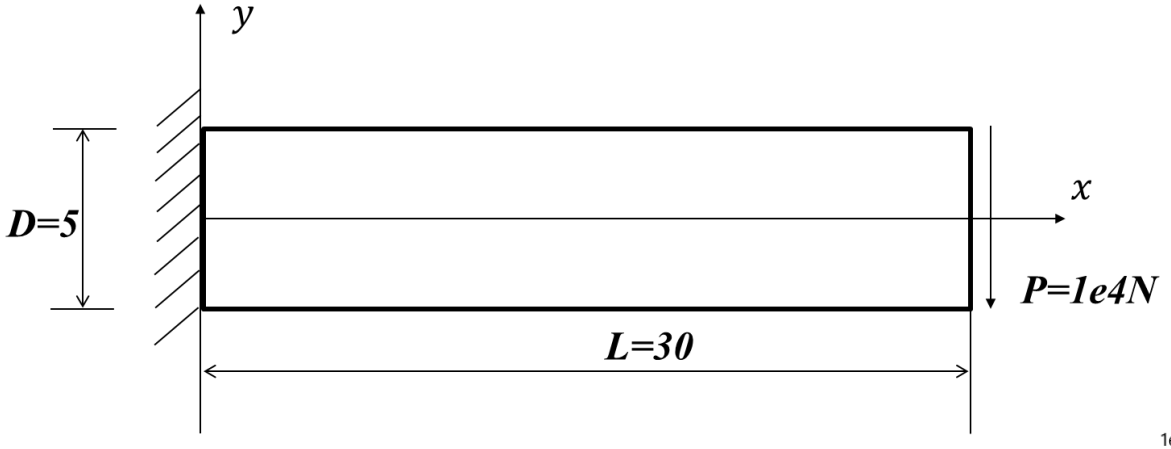
全局矩阵系统：$\begin{bmatrix} K_{11} & K_{12} \\ K_{21} & K_{22} \end{bmatrix} \begin{bmatrix} d_1 \\ d_2 \end{bmatrix} = \begin{bmatrix} F_1 \\ F_2 \end{bmatrix}$

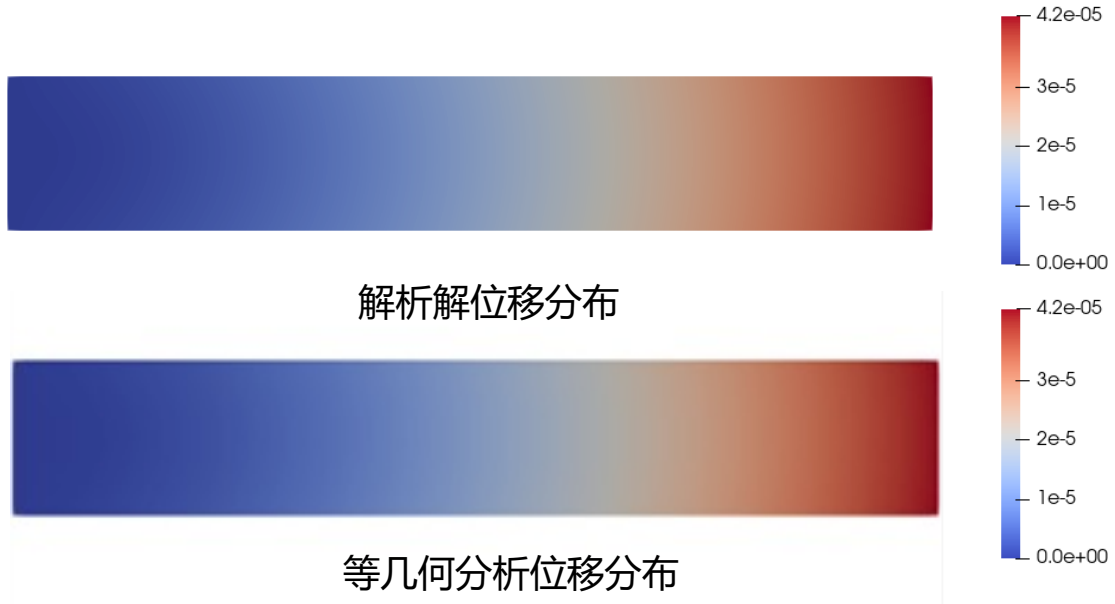
因此，方法2使得矩阵系统基于物理进行排序。这样的线性系统可以高效的使用基于物理的线性求解器和预处理器。

实验

线弹性问题单元离散平衡方程

$$\int_{\Omega_e} \boldsymbol{B}^T \boldsymbol{D} \boldsymbol{B} \boldsymbol{u}_e \mathrm{d}\Omega_e = \int_{\Omega_e} \boldsymbol{N}^T \boldsymbol{p} \mathrm{d}\Omega_e + \int_{\Gamma_N^e} \boldsymbol{N}^T \boldsymbol{t} \, \mathrm{d}\Gamma_N^e$$



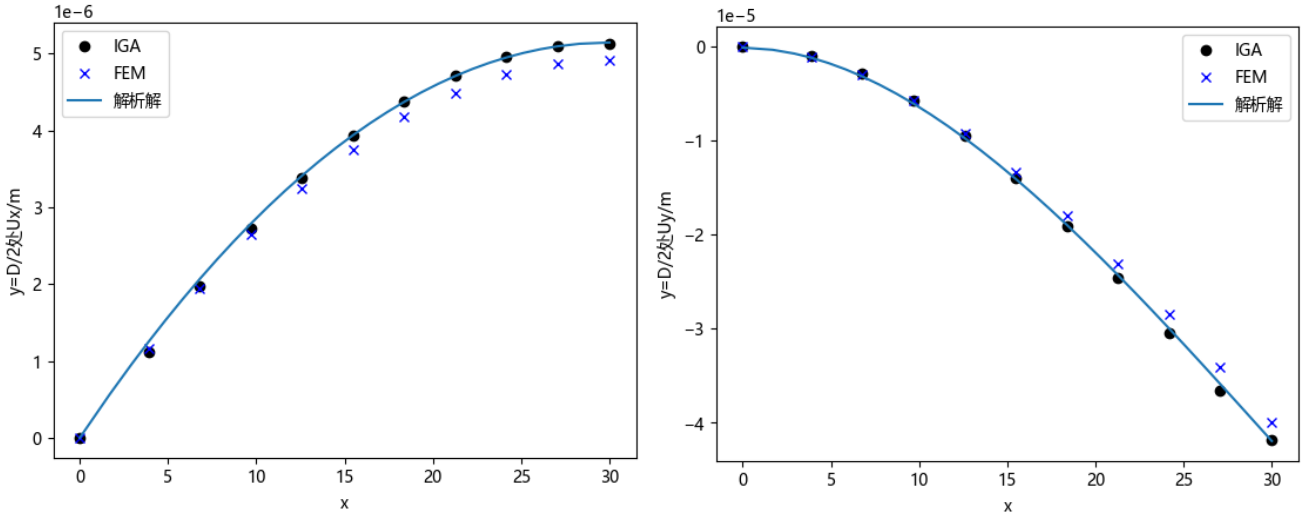


悬臂梁问题解析解

$$u_x = \frac{Py}{6EI}\left[(6L-3x)x + (2+\mu)\left(y^2 - \frac{D^2}{4}\right)\right]$$

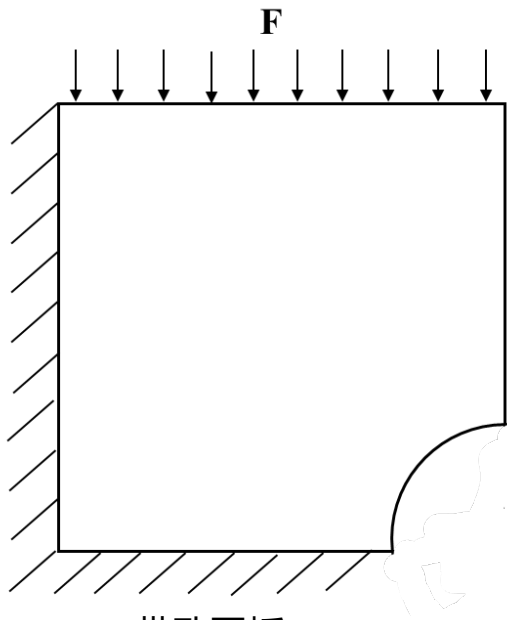
$$u_y = -\frac{P}{6EI}\left[3\mu y^2(L-x) + (4+5\mu)\frac{D^2 x}{4} + (3L-x)x^2\right]$$

$$\sigma_x = -\frac{P(L-x)y}{I}, \quad \sigma_y = 0, \quad \tau_{xy} = \frac{P}{2I}\left(\frac{D^2}{4} - y^2\right)$$

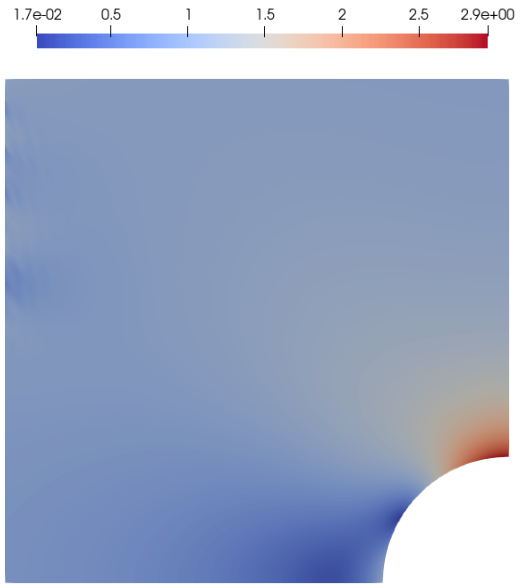


端点处有限元分析、等几何分析和解析解的位移大小

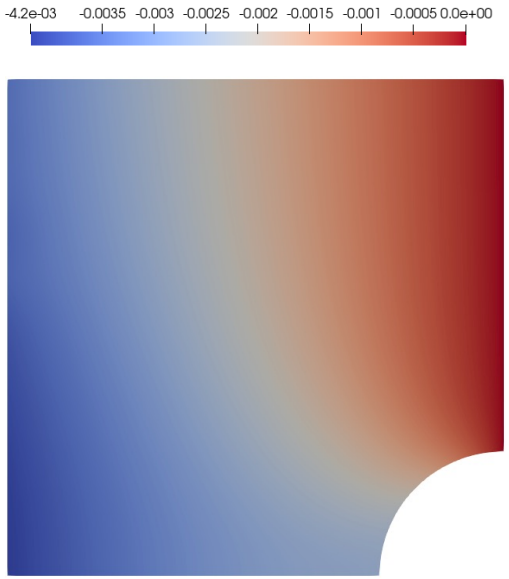
实验



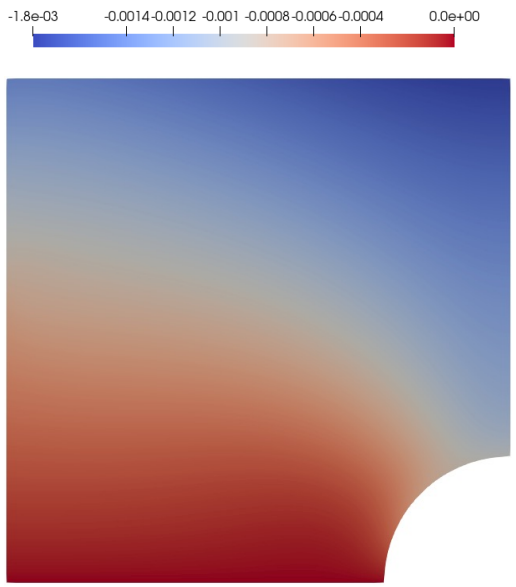
带孔平板



冯米斯应力分布

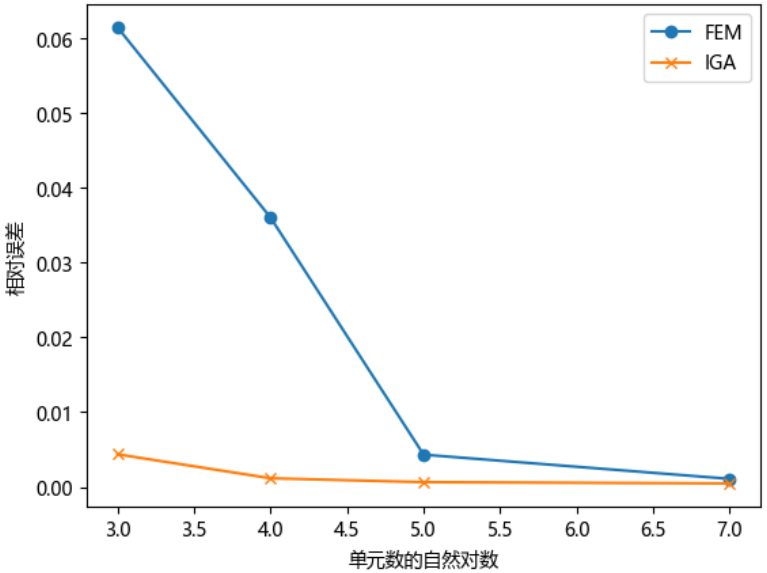


X方向位移分布



Y方向位移分布

$$error = \frac{\left\| \boldsymbol{s}^{h} - \boldsymbol{s}^{*} \right\|_{L2(\Omega)}}{\left\| \boldsymbol{s}^{*} \right\|_{L2(\Omega)}}$$



- 相同单元下等几何比有限元数值精度更高
- 粗网格下等几何也有极高的精度，可通过减少单元规模加速仿真过程

智能可视建模与仿真实验室

Intelligent Visual Modeling & Simulation (iGame) Lab

Thank You !

Q & A

Weakening the tight coupling between geometry and simulation in IGA

徐 岗

杭州电子科技大学计算机学院

Email: gxu@hdu.edu.cn
WEB: http://xugang.hdu.edu.cn

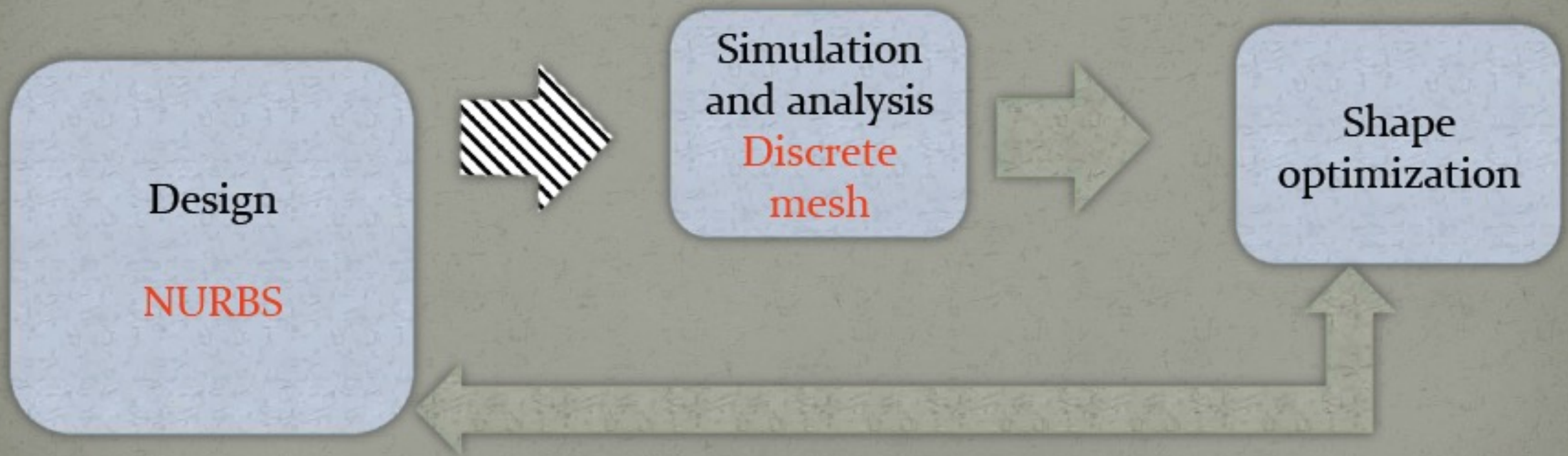
Joint work with S. Tomar, E. Atroshchenko, S. Bordas

Outline

- Main idea
- A general framework for GIFT
- Patch tests for GIFT
- Adaptive GIFT with PHT-spline
- Conclusion and future work

Seamless integration of CAD/CAE

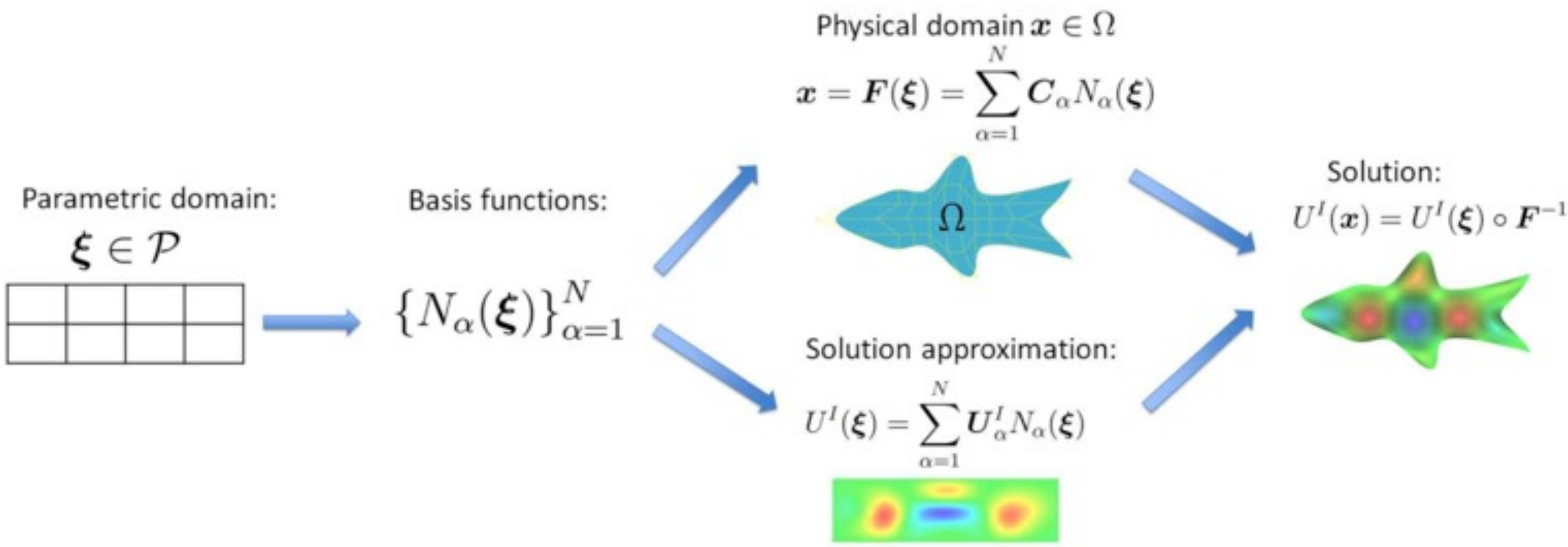
- One of “ Ten CAD Challenges ”
- Different geometric representation
- New analysis scheme ?



- IGA is an isoparametric, exact geometry approach, which is recently providing very promising results as an alternative to 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).

Main idea of IGA

Representing geometry and solution field in the same spline space



Related work

- Application in various simulation problems
- Application of various spline in IGA
  - NURBS
  - T-spline
  - Subdivision surfaces and volumes
  - PHT-spline
  - Powell-Sabin spline
  - Triangular B-spline
- Geometric modeling issue in IGA (Isogeometric Modeling)
  - Analysis-suitable spline
  - Analysis-suitable volume parameterization
- Shape optimization based on IGA

Motivation of GIFT

- This consistent spline representation has the potential to provide closer integration between geometric design and numerical analysis
- Some constraints which may be unwanted
- Conversion from the given NURBS geometry into the specified spline model is required as presented in the following paper

H. Speleers ,C. Manni , F. Pelosi. From NURBS to NURPS geometries.

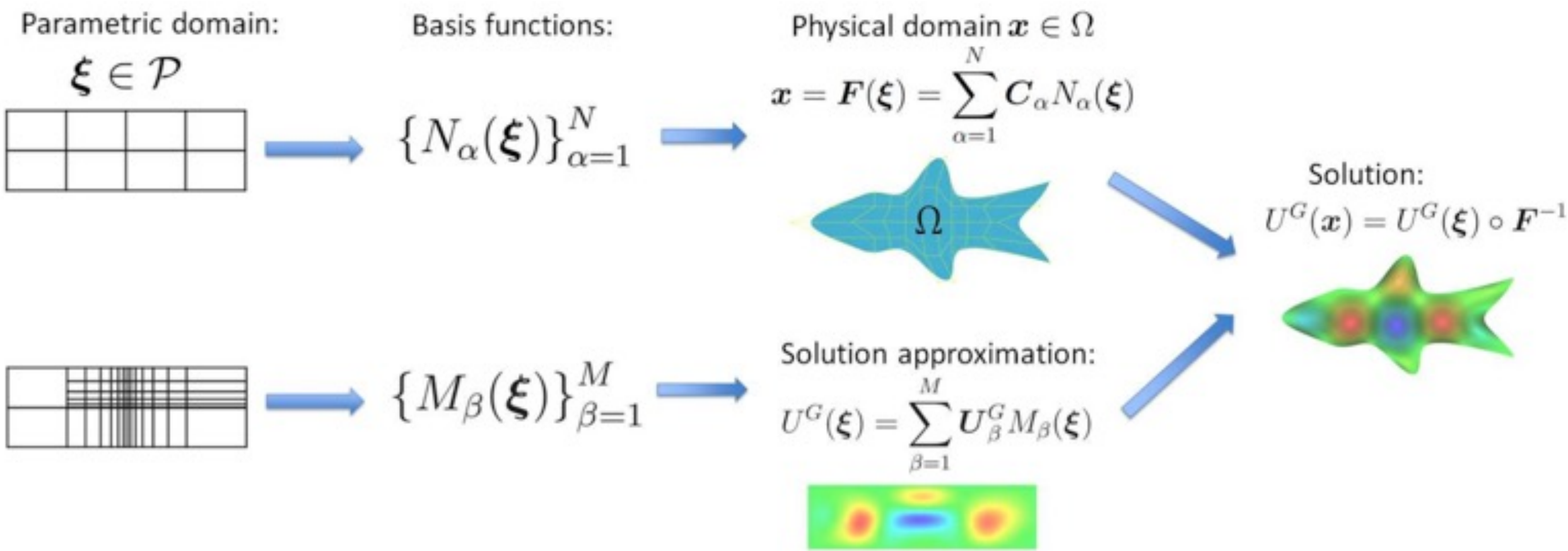
CMAME, 255(2013) 238-254

- In most cases, the exact conversion is impossible

Main idea of GIFT

Representing geometry and solution field in different spline space

***G**eometry-**I**ndependent **F**ield approxima**T**ion*



Comparison of IGA and GIFT

	IGA	**GIFT**
Solution field	Same spline space as the geometry	Geometry-independent spline space
Computational domain	Varies during h/p-refinements	Fixed during h/p-refinement
Integration element	Knot span of the parametric domainof the computational domain	Knot span of the parametric domain of the solution field
Degree of freedom	Fixed by the computational domain	Flexible
Continuity of solution field	Fixed by the computational domain	Flexible

Main features of GIFT

- Flexibility for solution representation: solution field with local strong gradient
- Independent control of continuity conditions : in case of multi-patches, the continuity condition between neighboring patches in the solution field can be automatically guaranteed without setting additional continuity constraints
- Independent field refinement: refinement operations by knot insertion and degree elevation are performed directly on the spline space of the solution field
- Efficient processing with pre-computing: Jacobian can be pre-computed to save computational expense

Outline

- Main idea
- A general framework for GIFT
- Patch tests for GIFT
- Adaptive GIFT with PHT-spline
- Shape optimization with GIFT
- Conclusion and future work

Model problem

$$
\begin{aligned}
\mathcal{A}u &= f, && \boldsymbol{x} \in \Omega \\
u &= 0, && \boldsymbol{x} \in \Gamma_D \\
\frac{\partial u}{\partial \boldsymbol{n}_A} &= g, && \boldsymbol{x} \in \Gamma_N,
\end{aligned}
$$

in which $\mathcal{A}$ is an elliptical partial differential operator

$$
\mathcal{A}u = -\partial_j \left(a_{ij}(\boldsymbol{x}) \partial_i u \right) + b_j(\boldsymbol{x}) \partial_j u + c(\boldsymbol{x}) u
$$

Weak form

Find $u \in H_0^1(\Omega)$, such that for any $v \in H_0^1(\Omega)$

$$a(u, v) = l(v), \tag{1}$$

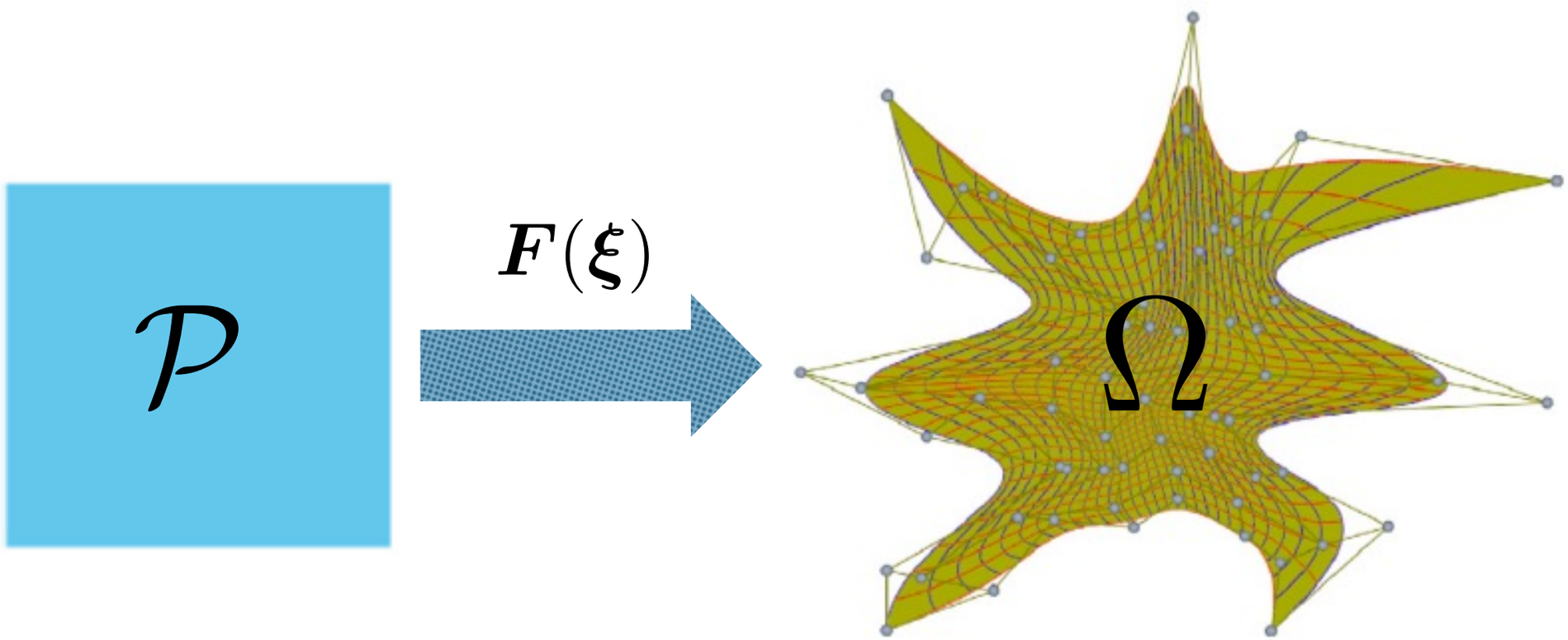
where

$$\begin{aligned} a(u, v) &= \int_{\Omega} \left\{ \left(a_{ij}(\boldsymbol{x}) \partial_i u \partial_j v + b_j(\boldsymbol{x}) \partial_j u v + c(\boldsymbol{x}) u v \right) \right\} d\Omega, \\ l(v) &= \int_{\Omega} f(\boldsymbol{x}) v(\boldsymbol{x}) d\Omega + \int_{\Gamma_N} g(\boldsymbol{x}) v(\boldsymbol{x}) d\Gamma \end{aligned} \tag{2}$$

NURBS computational domain

$$\boldsymbol{F} : \mathcal{P} \rightarrow \Omega, \quad \boldsymbol{x} = \boldsymbol{F}(\boldsymbol{\xi})$$

$$\boldsymbol{F}(\boldsymbol{\xi}) = \sum_{\alpha=1}^{N} \boldsymbol{C}_{\alpha} N_{\alpha}(\boldsymbol{\xi})$$



Main idea of GIFT

The main idea of the GIFT method is to seek for a solution in a geometry-independent spline space

$$V = \{u \in \mathrm{span}\{M_{i_1,i_2,\ldots i_d}(\boldsymbol{\xi})\} \circ \boldsymbol{F}^{-1}, u|_{\Gamma_D} = 0\},$$

where $M_{i_1,i_2,\ldots i_d}(\boldsymbol{\xi})$ can be a tensor product of NURBS, B-splines, T-splines, PHT-splines, etc.

The problem then consists in finding $u^G \in V$ such that for any $v^G \in V$

$$a(u^G, v^G) = l(v^G)$$

The solution is sought in the form

$$u^G(\boldsymbol{x}) = \sum_{\beta=1}^{M} U_\beta^G M_\beta(\boldsymbol{x}),$$

Poisson equation----

Annulus example with smooth solution

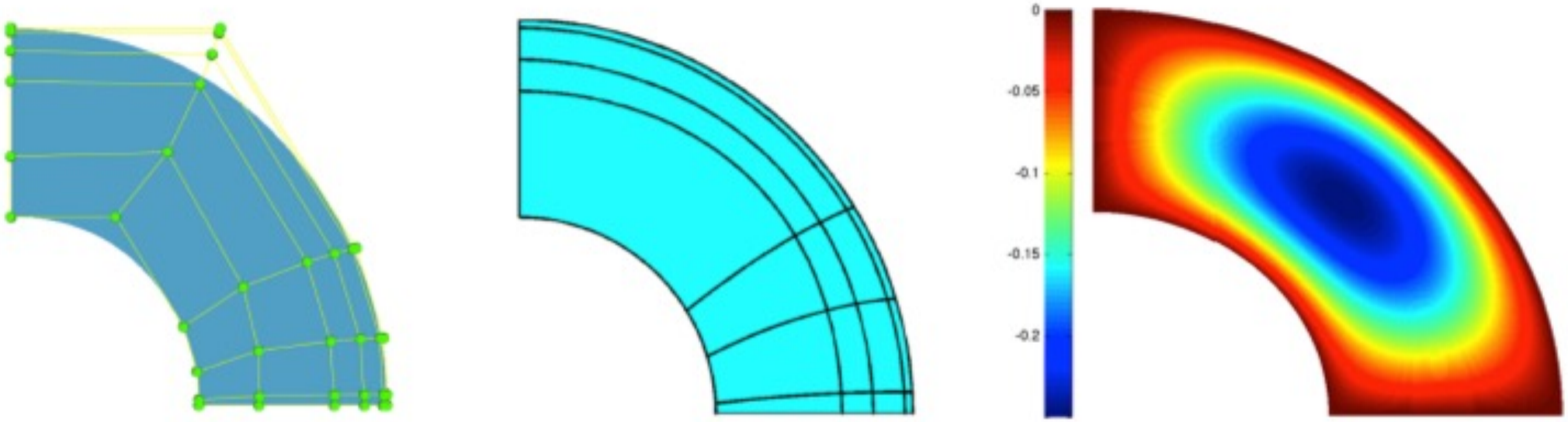
$$\begin{aligned} -\Delta u &= f \quad \text{in } \Omega \\ u &= 0 \quad \text{on } \partial\Omega \end{aligned}$$

- Source function

$$f(\boldsymbol{x}) = \frac{8 - 9\sqrt{x^2+y^2}}{x^2+y^2}\sin(2\arctan(y/x)),$$

- Exact solution

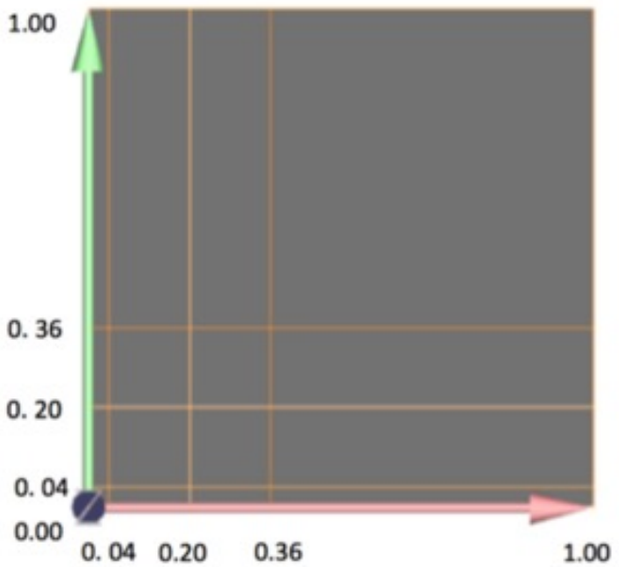
$$u(\boldsymbol{x}) = (x^2+y^2-3\sqrt{x^2+y^2}+2)\sin(2\arctan(y/x))$$



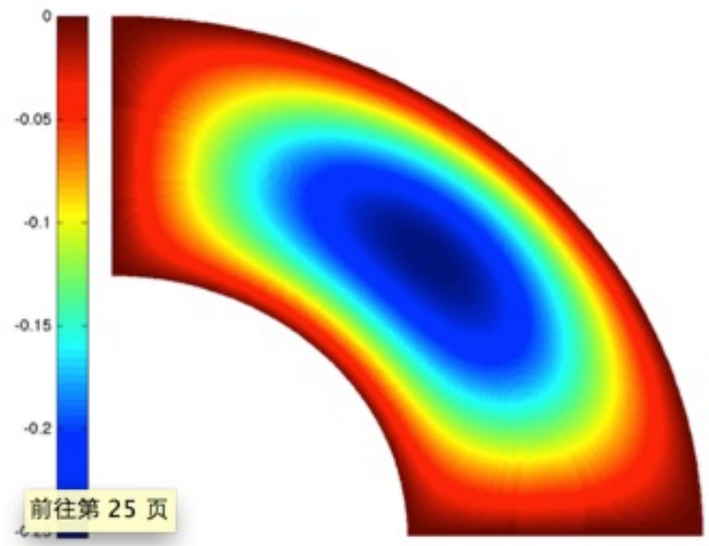
(a) computational domain (b) knot elements (c) exact solution

IGA (top) vs. GIFT (bottom)

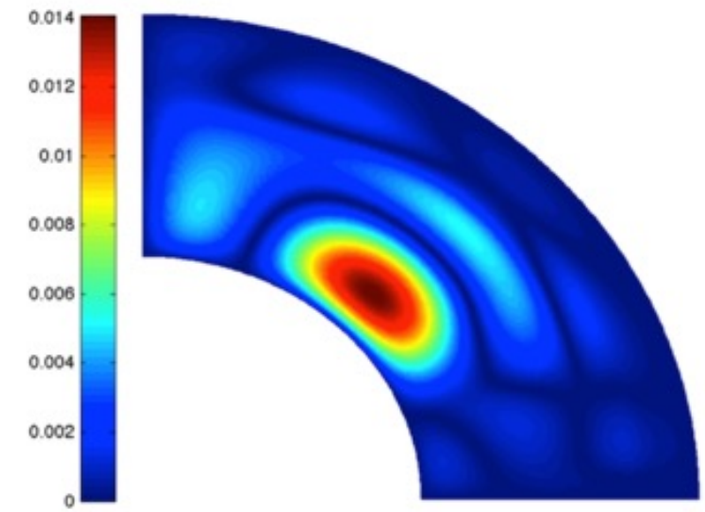
Representing geometry and solution field in spline spaces with different knot vectors



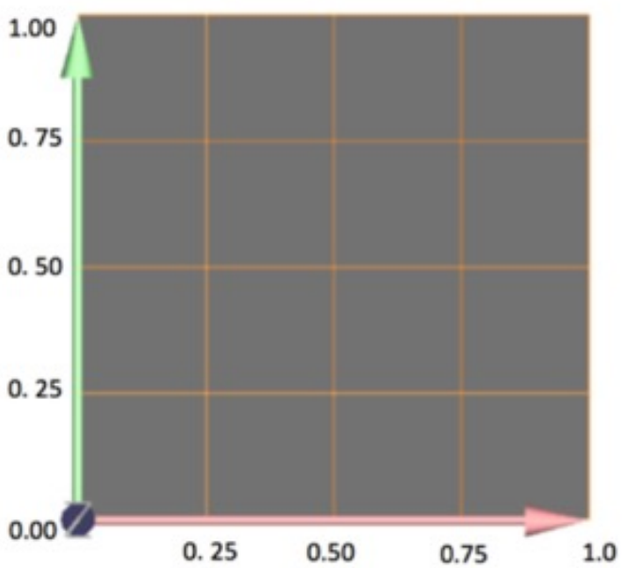
(d) IGA parametric domain



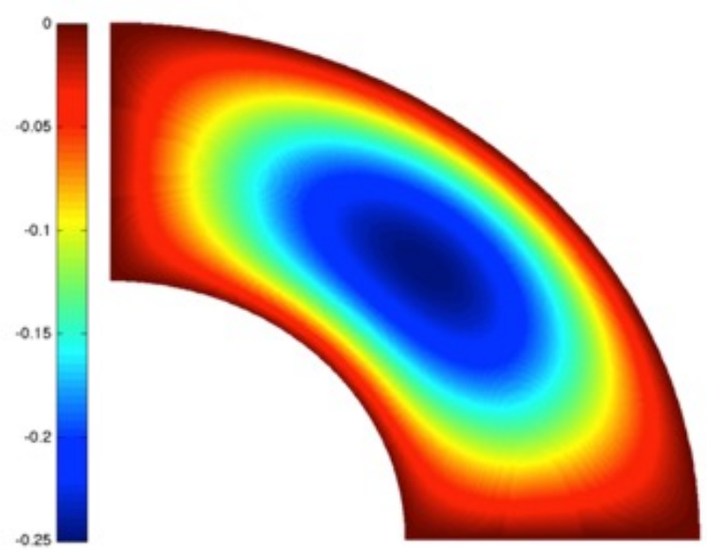
(e) IGA solution with quadratic NURBS



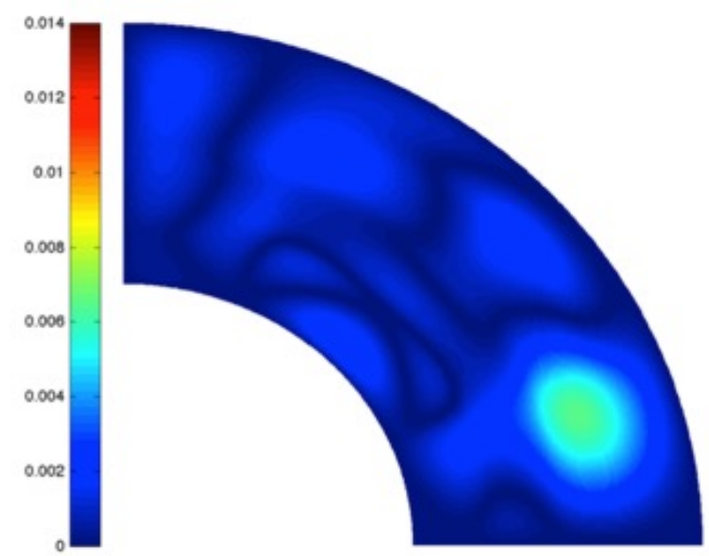
(f) IGA error color-map



(g)GIFT parametric domain



(h) GIFT solution with quadratic B-spline



(i)GIFT error color-map

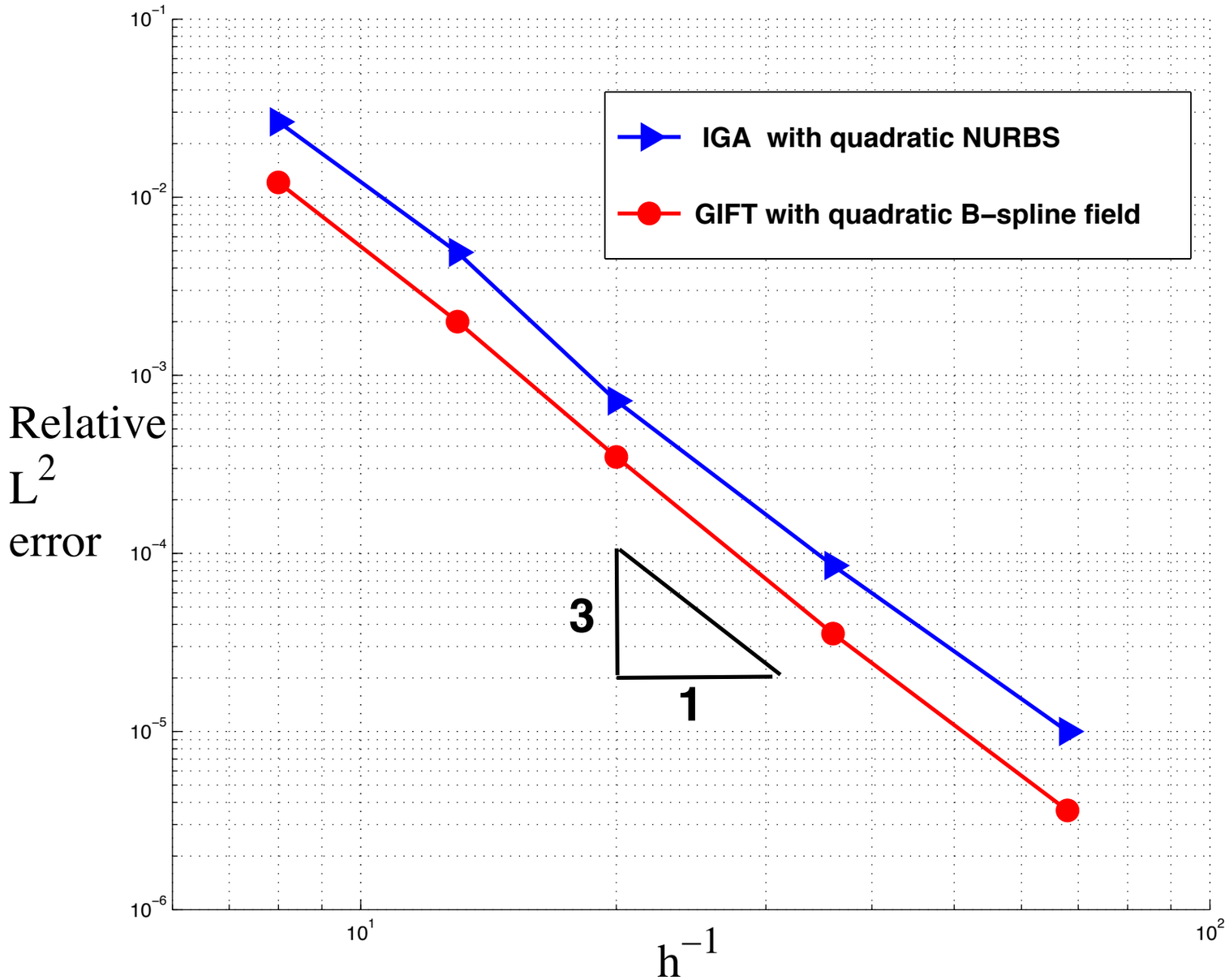
Quantitative data

	IGA	**GIFT**
SSSF	NURBS	polynomial B-spline
# DegF.	2×2	2×2
# ConF.	6×6	6×6
Ξ	$[0, 0, 0, 0.04, 0.2, 0.36, 1, 1, 1]$	$[0, 0, 0, 0.25, 0.5, 0.75, 1, 1, 1]$
$\mathcal{H}$	$[0, 0, 0, 0.04, 0.2, 0.36, 1, 1, 1]$	$[0, 0, 0, 0.25, 0.5, 0.75, 1, 1, 1]$
MeanErr	0.0022	0.001
MaxErr	0.0141	0.0065

Remark

By using uniform knot structure of the parametric domain, GIFT method leads to improved accuracy than IGA for the same parameterization of the computational domain.

Convergence result



$$e_{L_2} = \sqrt{\frac{\int_{\mathcal{P}} (u - u_{ext})^{\mathrm{T}} (u - u_{ext}) d\mathcal{P}}{\int_{\mathcal{P}} u_{ext}^{\mathrm{T}} u_{ext} d\mathcal{P}}}$$

Outline

- Main idea
- A general framework for GIFT
- Patch tests for GIFT
- Adaptive GIFT with PHT-spline
- Conclusion and future work

Patch tests (分片检验)

- 分片检验，是由 Irons 1965年首先提出的。它是评价弹性问题中形状函数在不满足连续性要求条件下位移型单元收敛性的一种方法，被广泛地使用。
- 由于 Stummel 给出了一个通过分片检验但不收敛的单元，以及 Sander 和 Beckers 给出了两个不通过分片检验但却收敛的单元，因而分片检验是收敛的充要条件受到怀疑。即便如此，分片检验仍是测试单元收敛性的有效工具
- 通过分片检验的一般标准：对于有精确解的边值问题，保证在边界条件施加，积分计算、求解线性系统等步骤中无任何误差的前提下，计算误差可以达到机器精度。

Patch test

Some historical background of the patch test

- I. Babuska and R. Narasimhan. The Babuska-Brezzi condition and the patch test: an example. *Comput. Methods Appl. Mech. Engrg.*, **140**, 183-199, 1997.
- G.P. Bazeley, Y.K. Cheung, B.M. Irons, and O.C. Zienkiewicz. Triangular elements in plate bending - conforming and nonconforming solutions, in *Proceedings of the Conference on Matrix Methods in Structural Mechanics*, Wright Patterson Air Force Base, Dayton, Ohio, 547-576, 1965.
- F. Stummel. The generalized patch test. *SIAM J. Numer. Anal.*, **16(3)**, 449-471, 1979.
- M. Wang. On the necessity and sufficiency of the patch test for convergence of nonconforming finite elements. *SIAM J. Numer. Anal.*, **39(2)**, 363-384, 2001.
- O.C. Zienkiewicz, and R.L. Taylor. The finite element patch test revisited: A computer test for convergence, validation and error estimates. *Comput. Methods Appl. Mech. Engrg.*, **149**, 223-254, 1997.
- T.J.R. Hughes. *The Finite Element Method: Linear Static and Dynamic Finite Element Analysis*. Prentice-Hall Inc., 1987.

Original geometry parametrization of the domain

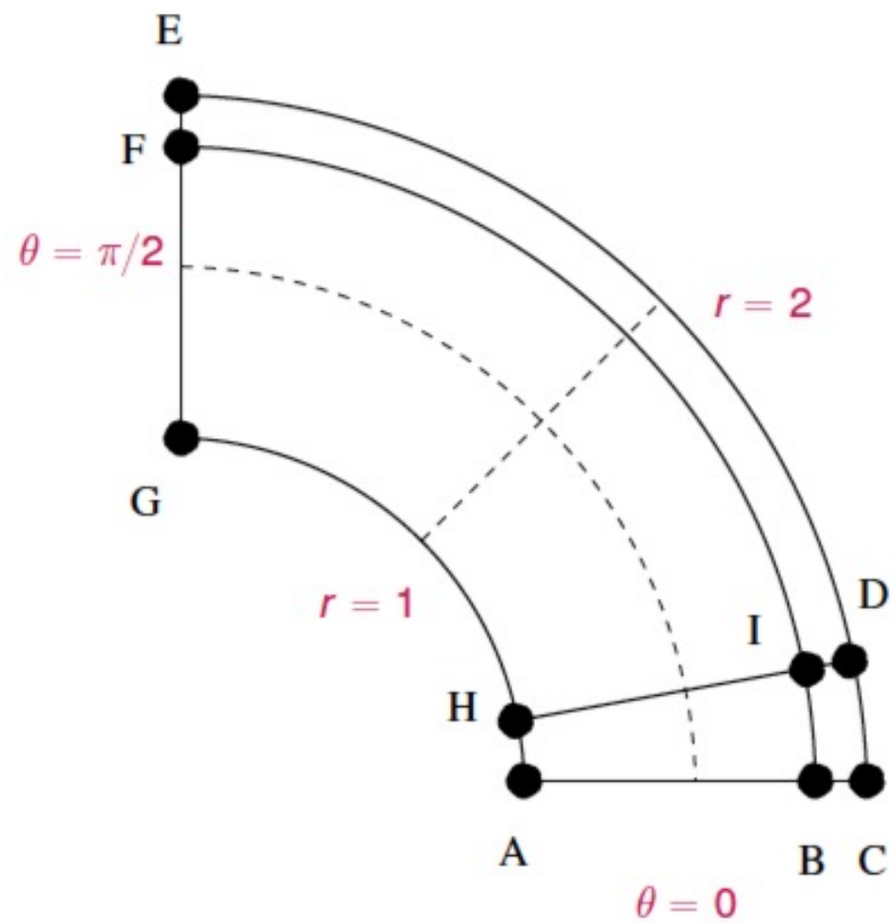
- The geometry is exactly represented by NURBS of degrees $1x2$
- Basic parametrization by one element, defined by 2 knot vectors

$$\Sigma = \{0, 0, 1, 1\}, \quad \Pi = \{0, 0, 0, 1, 1, 1\}.$$

- Together with NURBS basis, this is given by the following set of 6 control points, where the third value denotes the weight.

$$P[0,0] := \{1, 0, 1\}, \qquad P[1,0] := \{2, 0, 1\},$$
$$P[0,1] := \{1, 1, 1/\sqrt{2}\}, \qquad P[1,1] := \{2, 2, 1/\sqrt{2}\},$$
$$P[0,2] := \{0, 1, 1\}, \qquad P[1,2] := \{0, 2, 1\}.$$

Parametrization of the domain for the patch-test I



Quarter annulus region

- For patch-test in 2D, one-time h-refinement in both directions
- Consider the refined knot vectors

$$\Sigma = \{0, 0, s, 1, 1\}, \quad \Pi = \{0, 0, 0, t, 1, 1, 1\}.$$

Parametrization of the domain for the patch-test II

Shape A Uniform curvilinear elements $s = t = 1/2$

Shape B For non-uniform curvilinear elements, shift the points B, D, F, H and I. Set

$$t_1 := 1 - t + t/\sqrt{2}, \quad t_2 := t + \sqrt{2}t_1,$$

Updated set of control points in non-homogenized form

$$\{1, 0, 1\}, \quad \{1 + s, 0, 1\}, \quad \{2, 0, 1\}$$

$$\{1, \frac{t}{\sqrt{2}t_1}, t_1\}, \quad \{(1 + s), \frac{(1 + s)t}{\sqrt{2}t_1}, t_1\}, \quad \{2, \frac{\sqrt{2}t}{t_1}, t_1\}$$

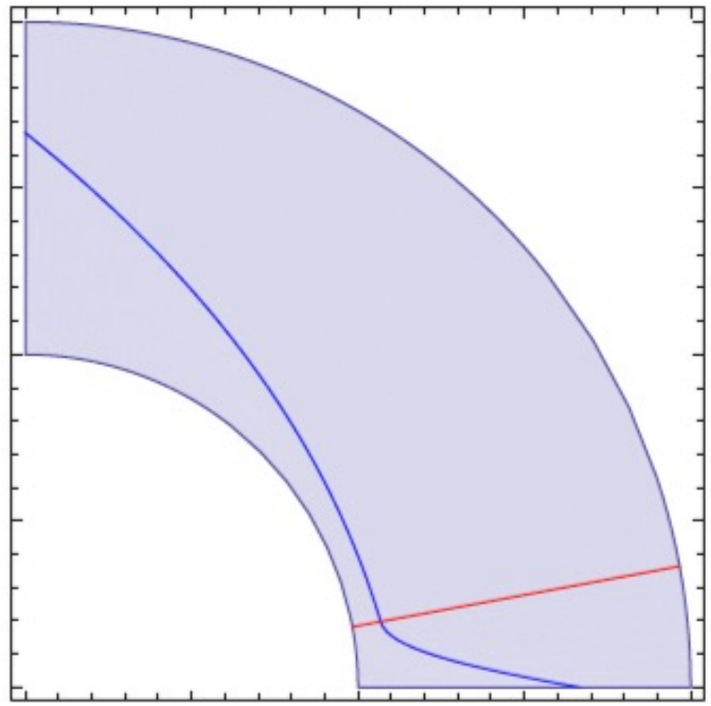
$$\{\frac{\sqrt{2}(1 - t)}{t_2}, 1, \frac{t_2}{2}\}, \{\frac{\sqrt{2}(1 + s)(1 - t)}{t_2}, (1 + s), \frac{t_2}{2}\}, \{\frac{2\sqrt{2}(1 - t)}{t_2}, 2, \frac{t_2}{2}\}$$

$$\{0, 1, 1\}, \quad \{0, 1 + s, 1\}, \quad \{0, 2, 1\}$$

Parametrization of the domain for the patch-test III

Shape C Add another parameter δ, two interior points changed as

$$\{\frac{(1+s)t_1}{t_1+\delta}, \frac{(1+s)t}{\sqrt{2}(t_1+\delta)}, t_1+\delta\}, \{\frac{\sqrt{2}(1+s)(1-t)}{t_2+2\delta}, \frac{(1+s)t_2}{t_2+2\delta}, \frac{t_2}{2}+\delta\}$$



Quarter annulus region with non-uniform elements,
($s = 2/3, t = 1/8, \delta = 1/2$)

Various combinations of degrees and knots/weights

- $p_u = p_g$, and $\Sigma_u = \Sigma_g$ (isogeometric case)
- $p_u = p_g$, and $\Sigma_u \neq \Sigma_g$
- $p_u < p_g$, and $\Sigma_u = \Sigma_g$
- $p_u < p_g$, and $\Sigma_u \neq \Sigma_g$
- $p_u > p_g$, and $\Sigma_u = \Sigma_g$
- $p_u > p_g$, and $\Sigma_u \neq \Sigma_g$

Total number of cases

- 3 choices of element shapes, and 6 choices of degrees/knots, total of 18 cases !!

Results for various choices of $p_u, p_g, \Sigma_u, \Sigma_g$

- L^2 error in the solution $u = 1 + x + y$
- $p_u = p_g$, i.e. $p_u = p_g = 1x2$
- $p_u < p_g$, i.e. $p_u = 1x2$, $p_g = 2x3$
- $p_u > p_g$, i.e. $p_u = 2x3$, $p_g = 1x2$

p and Σ	Shape A	Shape B	Shape C
$p_u = p_g$, $\Sigma_u = \Sigma_g$	8.2469e-016	8.5695e-016	1.5017e-015
$p_u < p_g$, $\Sigma_u = \Sigma_g$	8.5489e-016	8.8080e-016	1.5172e-015
$p_u > p_g$, $\Sigma_u = \Sigma_g$	1.1979e-015	1.8185e-015	3.6019e-015
$p_u < p_g$, $\Sigma_u \neq \Sigma_g$	1.1401e-015	1.0907e-015	0.20155
$p_u > p_g$, $\Sigma_u \neq \Sigma_g$	1.5010e-015	1.4193e-015	0.03019
$p_u = p_g$, $\Sigma_u \neq \Sigma_g$	7.4477e-016	1.0611e-015	0.20155

Concluding remarks

- When knot data are same (same representation/basis)
  - various combinations of polynomial degrees pass the test for all kind of parametric elements
- When knot data are different (allowing different basis)
  - various combinations of polynomial degrees pass the test on rectangular elements in parametric domain

One message

- Without any fancy/weird parametric elements, various combinations of different basis and polynomial degrees pass the test, and can be used in practice

Outline

- Main idea
- A general framework for GIFT
- Patch tests for GIFT
- Adaptive GIFT with PHT-spline
- Conclusion and future work

PHT-spline [Deng et al., 2008]

- Piecewise polynomials over a hierarchical T-mesh
- Linear independence
- Simple local refinement

[Deng et al., 2008]

Main idea of PHT-GIFT

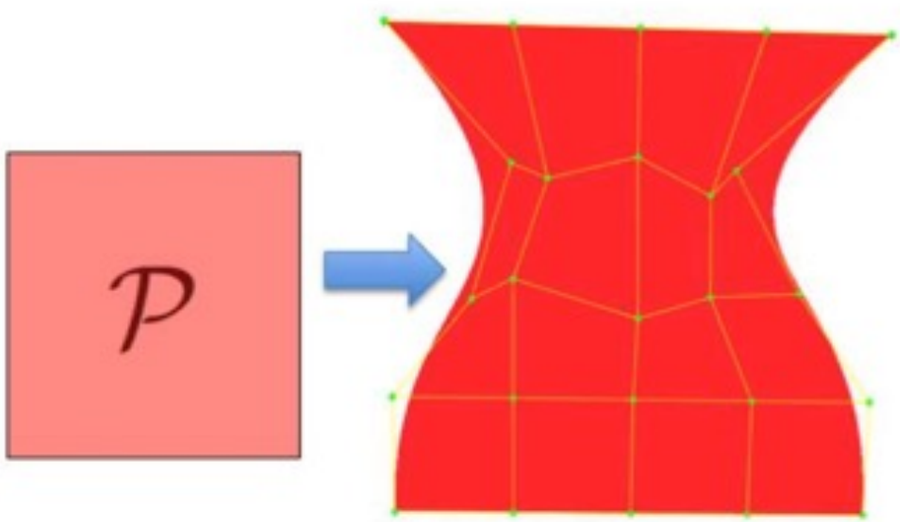
- Computational domain: NURBS space
- Solution field: PHT-spline space with local-refinement property
- Independent field refinement:

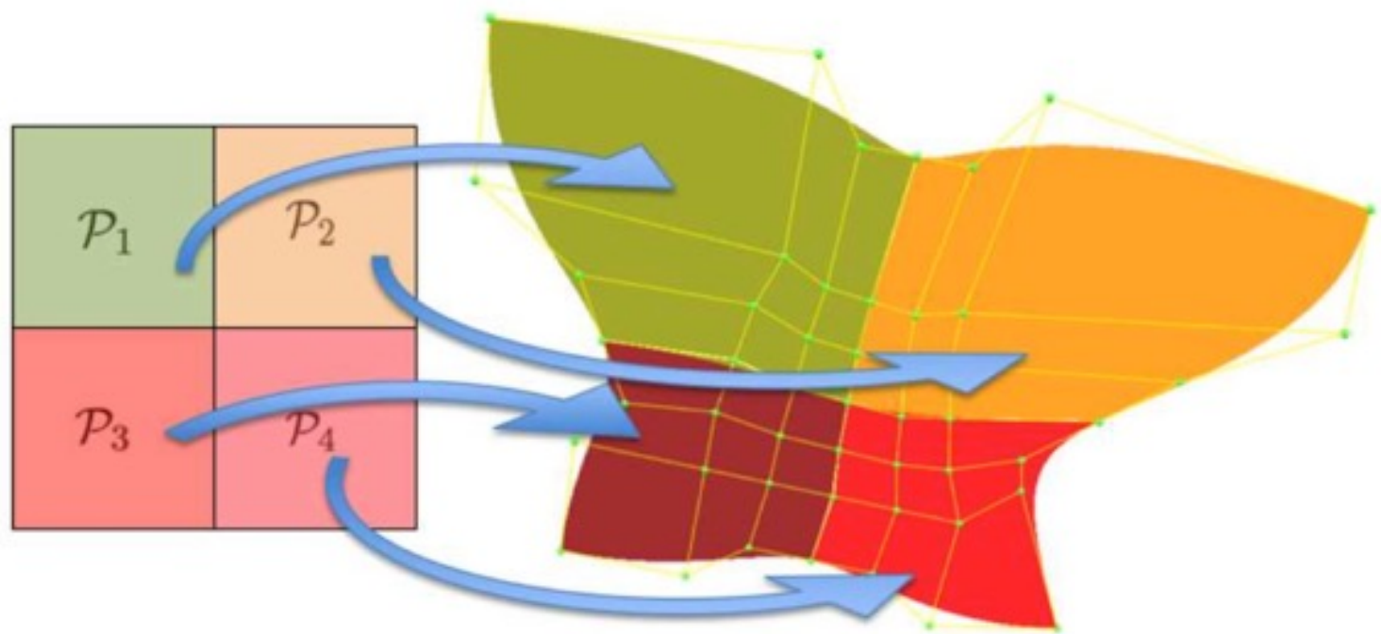
local refinement is only performed on the solution field

Pre-processing

- Construct the common parametric domain of geometry and solution field

- Case I : If the computational domain is made of a single NURBS or multi-patches in which the parametric domain of each patch forms a quad-mesh, then the parametric domain of the solution field is constructed as the partition-mesh formed by the knot lines on the parametric domain of the NURBS patches.

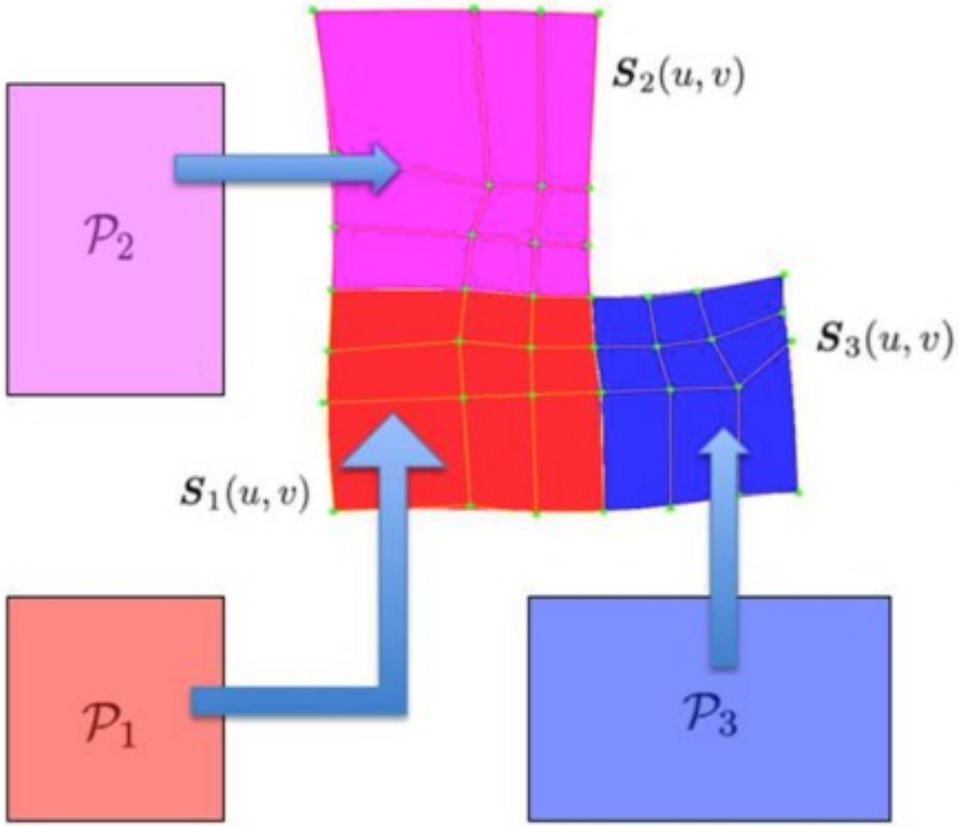


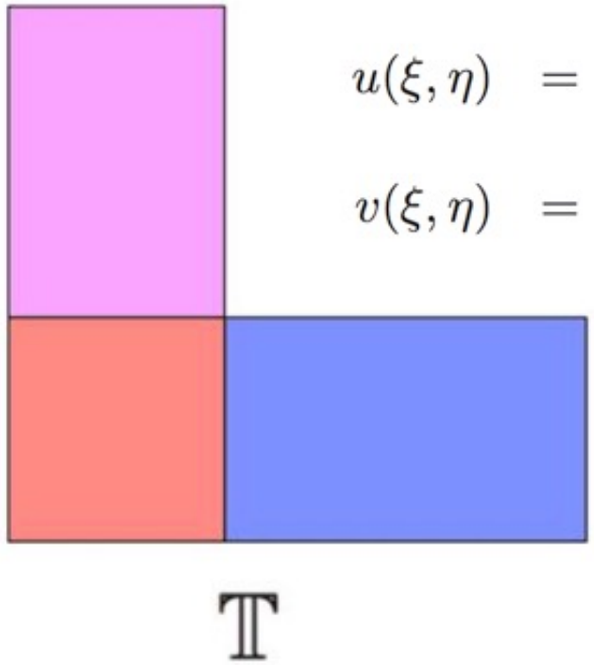


Pre-processing (cont.)

- For other cases, we should construct the initial parametric mesh according to the topological connection information of the NURBS patches. A reparameterization operation should be performed.

- Remark: There is actually no need to derive the explicit parametric representation of the reparameterized surface. The derivative information required in the solving stage can be evaluated from the original parameterization through the Jacobian transformation matrix.





$$u(\xi,\eta) = \frac{1}{f-e}[(f-\xi)a + (\xi-e)b]$$

$$v(\xi,\eta) = \frac{1}{h-g}[(h-\eta)c + (\eta-g)d]$$

Pipeline

Algorithm 1 Local refinement scheme in GIFT using PHT-splines

Input: Planar NURBS parameterization $\boldsymbol{F}(\xi, \eta)$ of computational domain Ω

Output: PHT-spline solution field

1: Compute the PHT-spline solution $\mathcal{T}(\xi, \eta)$ of model problem (17) over the given NURBS parameterization $\boldsymbol{F}(\xi, \eta)$ by GIFT .

2: Calculate the local error indicator $e_{\mathcal{K}}$ patch by patch on the solution field

3: Mark the parametric cells to be refined by a mean-value marking algorithm

4: Subdivide the marked cells into four sub-celles on the parametric domain of the PHT-spline field.

5: Construct the PHT-spline basis functions over the refined T-mesh of the parametric domain.

6: Compute the new solution field $\mathcal{T}(\xi, \eta)$ in the refined PHT-spline space.

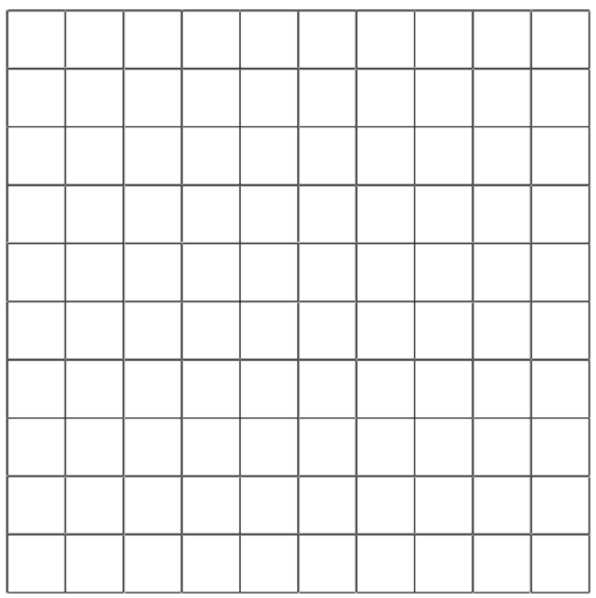
7: Repeat the above refinement steps until the estimated error is less than a given threshold.

Poisson equation----

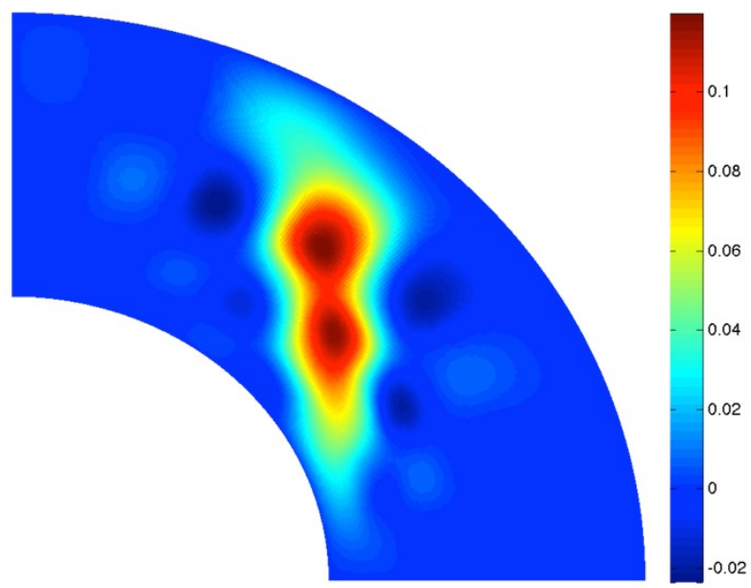
Annulus solution field with sharp peak area

NURBS geometry with PHT-spline solution field

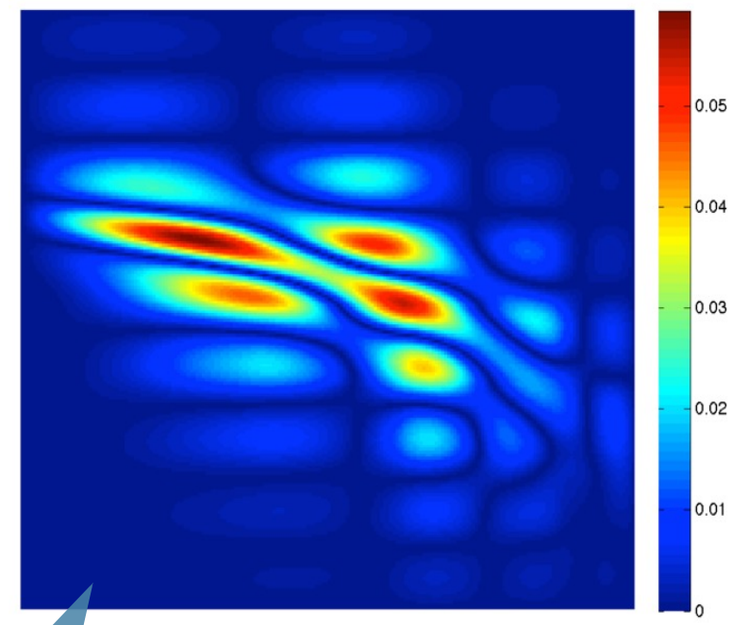
$$\boldsymbol{u} = (r-1)(r-2)\theta(\theta - \frac{\pi}{2})\exp(-100(r\cos\theta - 1)^2)$$



T-partition
on the parametric domain



GIFT solution over
NURBS geometry



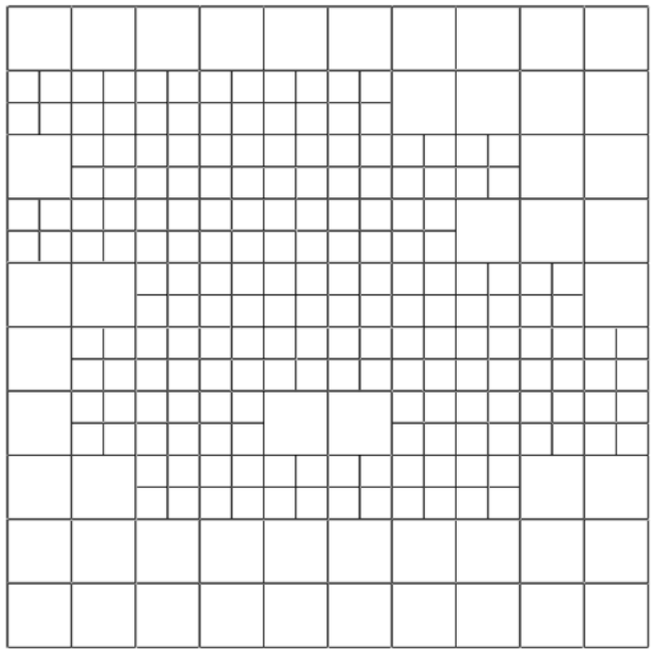
GIFT error
on the parametric domain

Guide the local refin-
ement of T-partition

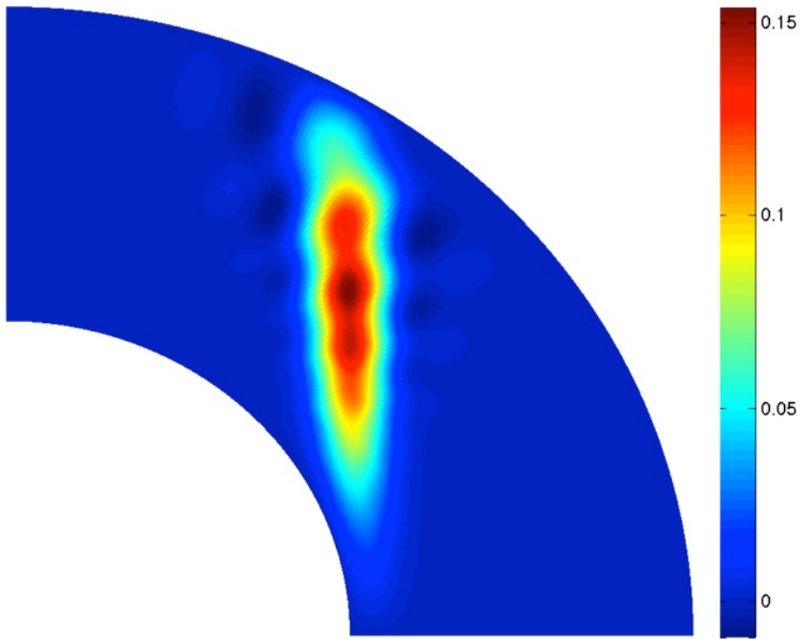
Poisson equation---- (Cont.)

Annulus solution field with sharp peak area

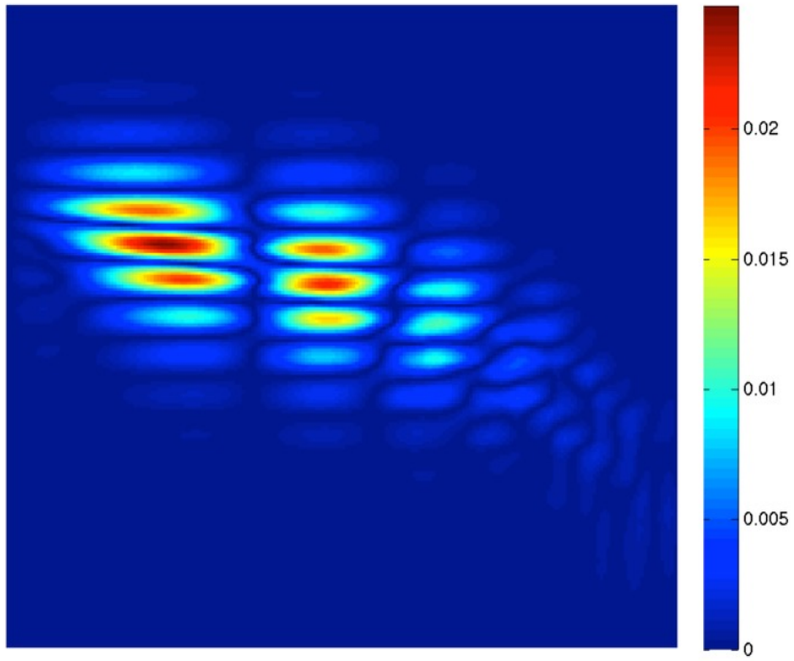
NURBS geometry with PHT-spline solution field



T-partition
on the parametric domain



GIFT solution over
NURBS geometry

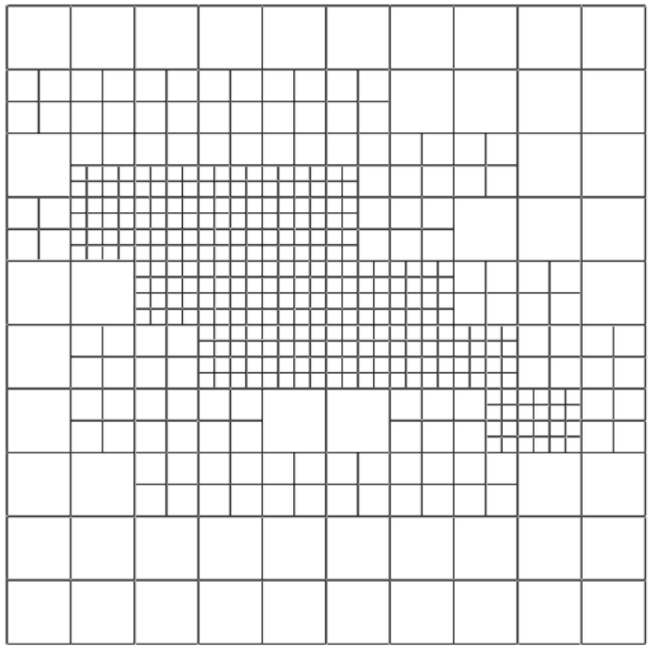


GIFT error
on the parametric domain

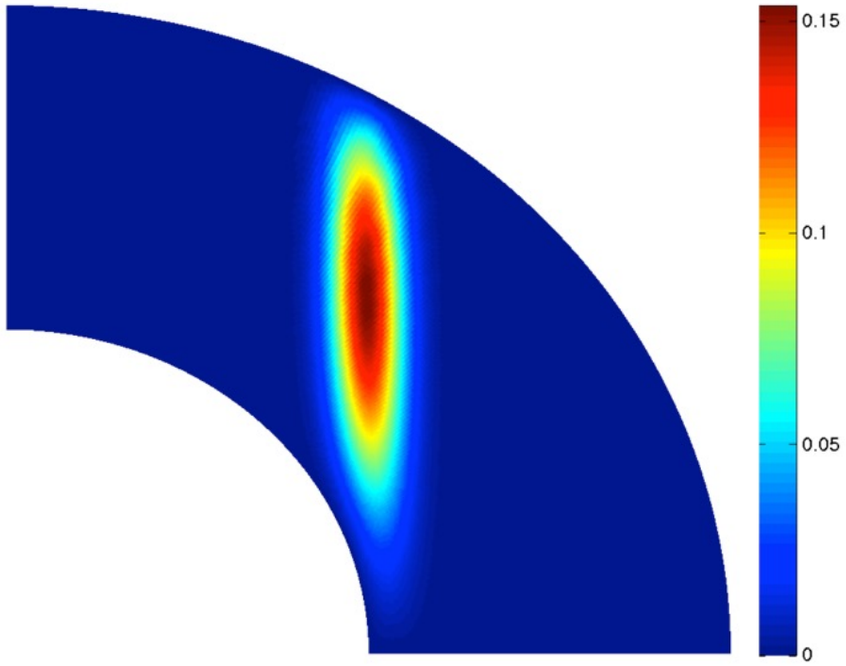
Poisson equation---- (Cont.)

Annulus solution field with sharp peak area

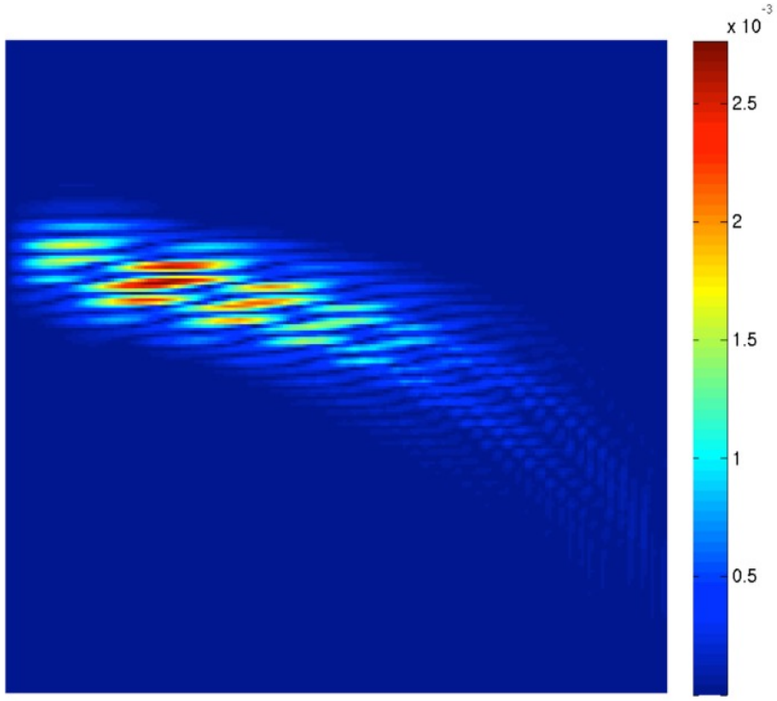
NURBS geometry with PHT-spline solution field



T-partition
on the parametric domain



GIFT solution over
NURBS geometry

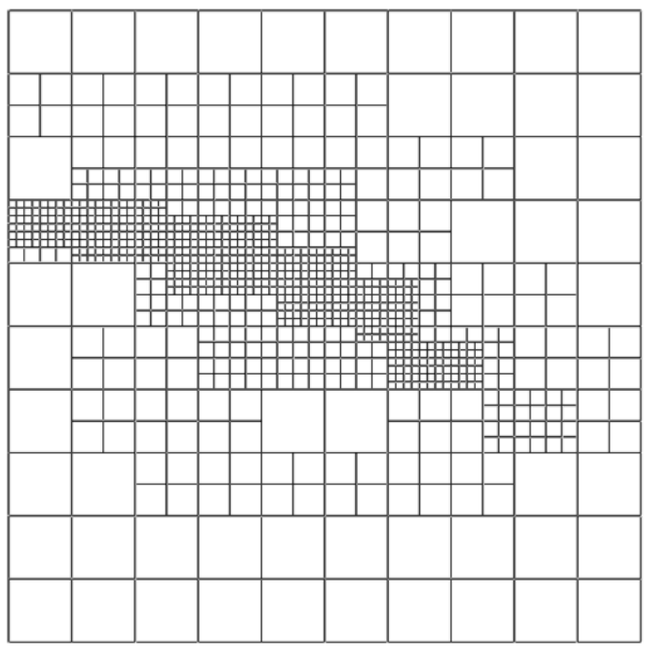


GIFT error
on the parametric domain

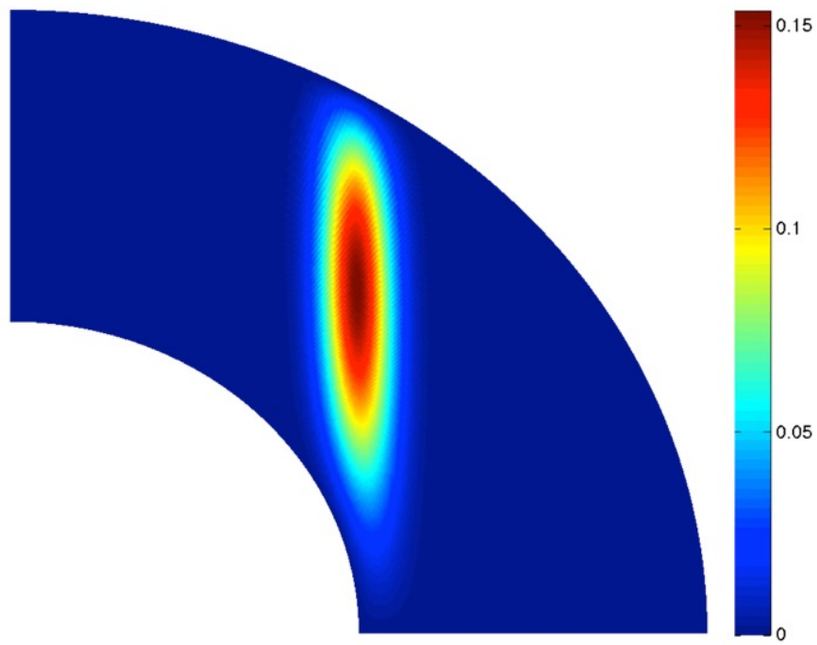
Poisson equation---- (Cont.)

Annulus solution field with sharp peak area

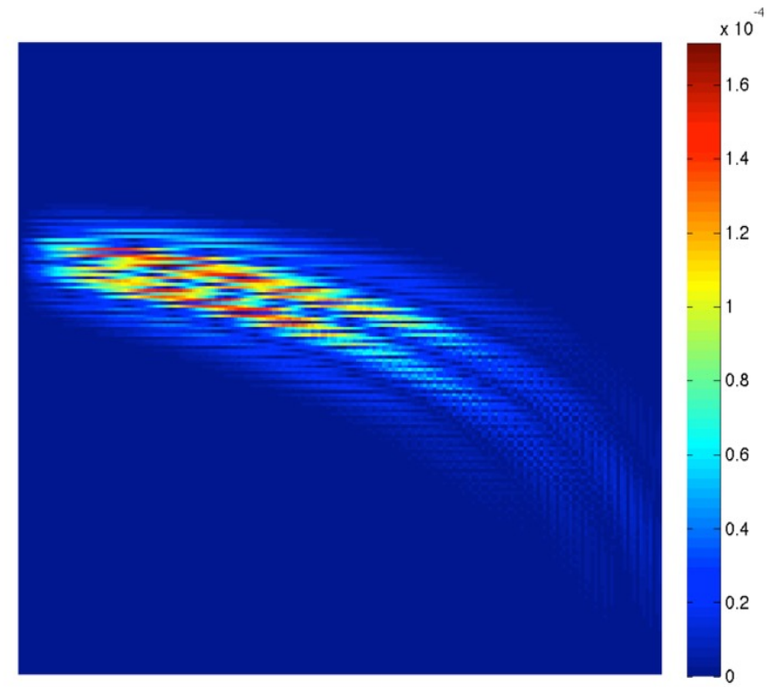
NURBS geometry with PHT-spline solution field



T-partition
on the parametric domain

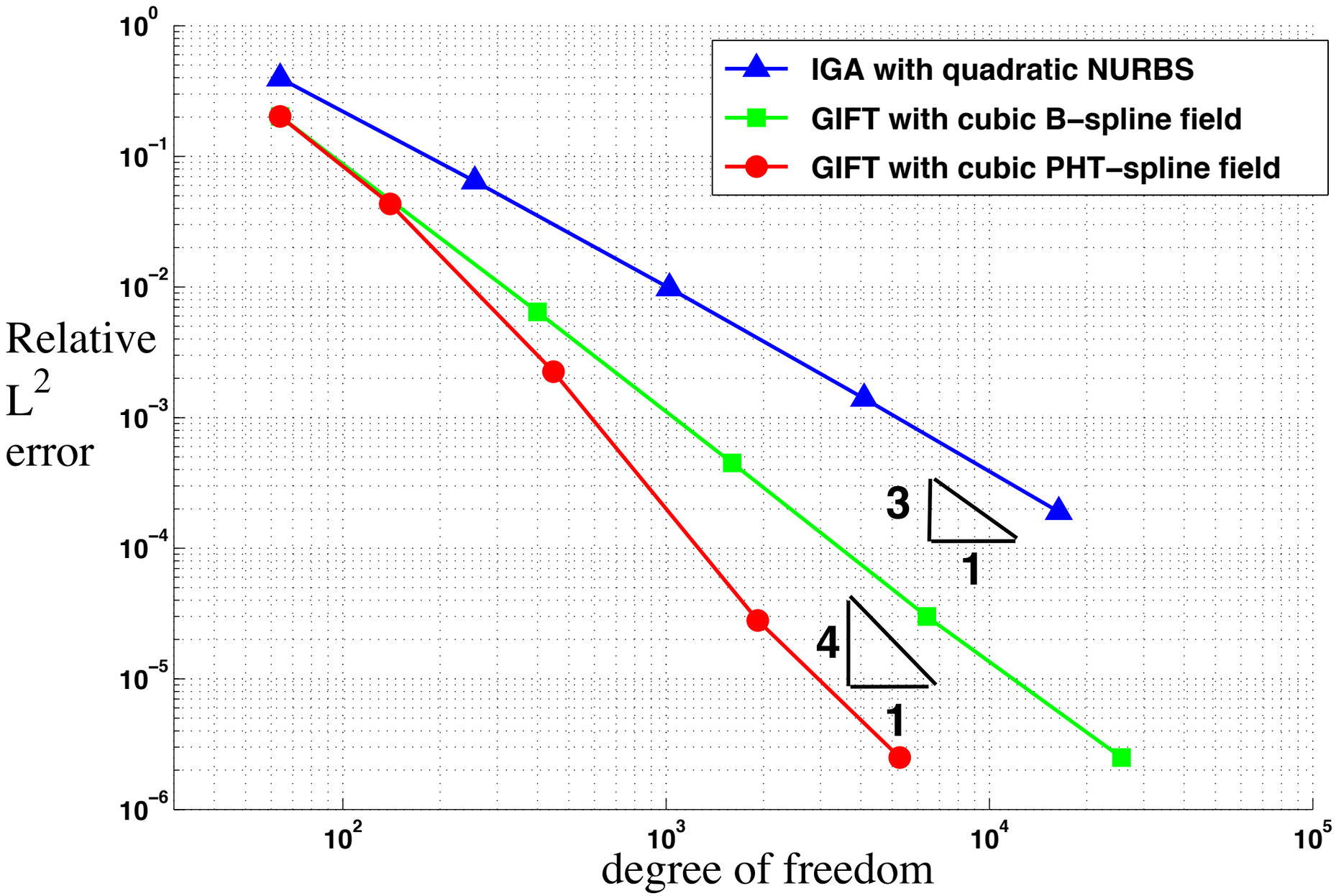


GIFT solution over
NURBS geometry



GIFT error
on the parametric domain

Convergence result



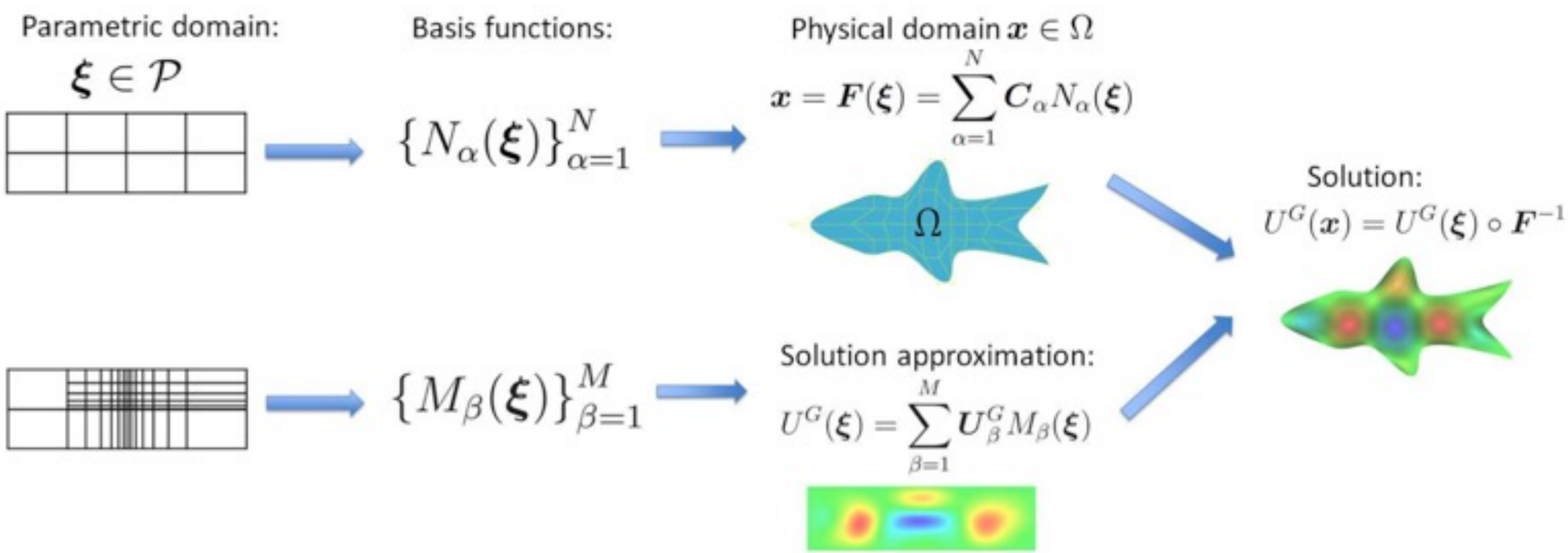
$$e_{L_2} = \sqrt{\frac{\int_{\mathcal{P}} (u - u_{ext})^{\mathrm{T}} (u - u_{ext}) d\mathcal{P}}{\int_{\mathcal{P}} u_{ext}^{\mathrm{T}} u_{ext} d\mathcal{P}}}$$

Outline

- Main idea
- A general framework for GIFT
- Patch tests for GIFT
- Adaptive GIFT with PHT-spline
- Conclusion and future work

Conclusion

- GIFT: the spline spaces of the solution field and the geometry can be chosen independently
- Adaptive GIFT: NURBS Geometry + PHT Simulation



Future work

- Detailed studies of the stability and convergence properties
- Efficient and general construction of common parametric domain (Pre-processing)
- Automatic selection of the most suitable field spline space
- GIFT in Crack growth or other free boundary problems
- GIFT for isogeometric shape and topology optimization
- GIFT within an isogeometric boundary element framework
- Parallelization on GPUs