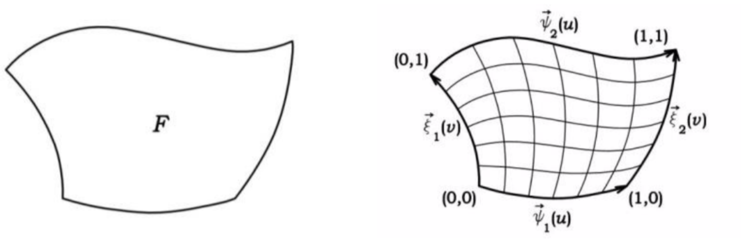
INF2604 – Geometria Computacional

Prática 5: Projetor bilinear

Prof. Waldemar Celes
Departamento de Informática, PUC-Rio

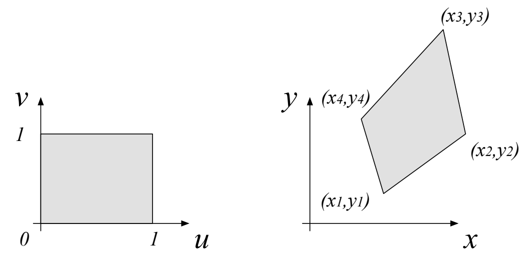
11 de setembro de 2025

O objetivo deste exercício é implementar um projetor bilinear para geração de malhas de quadriláteros, conforme ilustra a figura abaixo.



O domínio F é definido por quatro curvas *Psi*, ψ_1 e ψ_2, e *Csi*, ξ_1 e ξ_2. Neste exercício, essas curvas serão representadas por linhas poligonais. Dada uma curva *Zeta* qualquer $\zeta = \{x_0, y_0, x_1, y_1, \cdots, x_{n-1}, y_{n-1}\}$, prevê-se a implementação de uma função que, dado um parâmetro $u \in [0, 1]$, retorna o ponto correspondente na curva $\zeta(u)$. Como caso particulares, tem-se $\zeta(0) = \{x_0, y_0\}$ e $\zeta(1) = \{x_{n-1}, y_{n-1}\}$.

Para entender o projetor bilinear, deve-se primeiro considerar o mapeamento paramétrico de um quadrilátero. Um ponto no interior de quadrilátero no espaço paramétrico $u, v \in [0, 1]$ é mapeado para o ponto correspondente no quadrilátero no espaço físico $x, y \in \mathbb{R}$ através de funções de forma, com base nos vértices do quadrilátero físico.



$$\mathbf{p} = (1-u)(1-v)\mathbf{p}_1 + u(1-v)\mathbf{p}_2 + uv\mathbf{p}_3 + (u-1)v\mathbf{p}_4$$

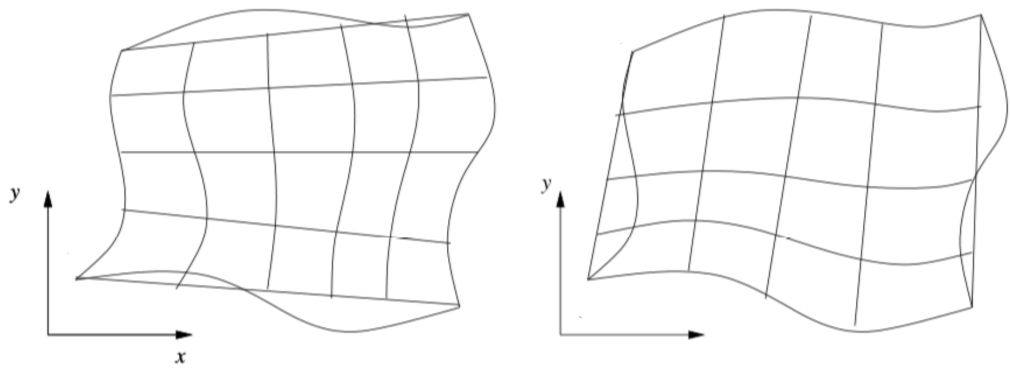
Considerando uma região quadrilateral delimitada pelas quatro curvas, podemos fazer uma interpolação entre as curvas ψ e outra entre as curvas ξ:

$$\mathbf{p} = (1-v)\psi_1(u) + v\psi_2(u)$$

e

$$\mathbf{p} = (1-u)\xi_1(v) + u\xi_2(v)$$

obtendo, respectivamente, as malhas ilustradas abaixo.



Para se chegar ao projetor bilinear, basta somar as duas interpolações e subtrair o mapeamento paramêtrico:

$$
\begin{aligned}
\mathbf{p} =&(1-v)\psi_1(u) + v\psi_2(u) + (1-u)\xi_1(v) + u\xi_2(v) - \\
&((1-u)(1-v)\mathbf{p}_1 + u(1-v)\mathbf{p}_2 + uv\mathbf{p}_3 + (u-1)v\mathbf{p}_4)
\end{aligned}
$$

Pede-se: Implemente um gerador de malha quadrilateral baseado no projetor bilinear. Como entrada, tem-se as quatro curvas que delimitam o domínio ψ_1, ψ_2, ξ_1 e ξ_2, fornecidas em um arquivo texto que indica o número de pontos n e as coordenadas dos pontos x, y de cada curva, em sequência:

n

$x_0 \;\; y_0$

$\cdots$

$x_{n-1} \;\; y_{n-1}$

n

$x_0 \;\; y_0$

$\cdots$

$x_{n-1} \;\; y_{n-1}$

n

$x_0 \;\; y_0$

$\cdots$

$x_{n-1} \;\; y_{n-1}$

n

$x_0 \;\; y_0$

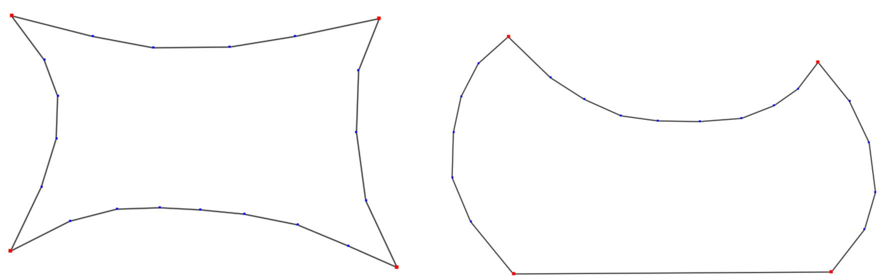
$\cdots$

$x_{n-1} \;\; y_{n-1}$

Seu programa deve salvar as coordenadas dos vértices da malha (não é necessário salvar as incidências dos elementos):

$$
\begin{array}{l}
n \\
x_0 \;\; y_0 \\
x_1 \;\; y_1 \\
x_2 \;\; y_n \\
\cdots \\
x_{n-1} \;\; y_{n-1}
\end{array}
$$

Para testar, considere as curvas ilustradas abaixo, descritas nos arquivos `curvas1.txt` e `curvas2.txt`.



Seu programa de teste deve salvar os vértices da malha nos arquivos `malha1.txt` e `malha2.txt`, respectivamente.

Entrega: Os códigos fontes do algoritmo e do teste devem ser enviados via página da disciplina no EAD. O aluno também deve mandar os arquivos de saída teste (`malha1.txt` e `malha2.txt`). O prazo final para envio é **segunda-feira, dia 22 de novembro**.