

## Trabalho sobre Computação Gráfica Interativa Bidimensional

### *Visualização de elementos finitos isoparamétricos*

#### Objetivo

O objetivo deste trabalho é a criação da interface gráfica (menus, botões, janelas de desenho, etc.) de um programa para visualização e manipulação de elementos finitos isoparamétricos planos. Este trabalho é uma continuação do primeiro trabalho, isto é, as funções implementadas no primeiro trabalho serão utilizadas para desenhar um elemento finito na janela de desenho e calcular o determinante do Jacobiano da transformação isoparamétrica. As posições dos nós do elemento finito podem ser manipuladas com a utilização do *mouse*. Para cada configuração deformada do elemento, o programa desenha, através de *franjas coloridas* (também chamadas de *iso-faixas*), a variação do determinante do Jacobiano no domínio do elemento.

#### Especificação

Assim como no primeiro trabalho, é fornecido um programa incompleto e o objetivo do trabalho é implementar as funções que complementam o programa. O entendimento do programa fornecido faz parte deste trabalho.

A estrutura de dados utilizada para armazenar em memória um elemento finito é a mesma utilizada no primeiro trabalho. A solução do primeiro trabalho deve ser copiada para este trabalho.

O aluno terá que implementar funções que, através de chamadas a funções do sistema gráfico CD, desenhem o elemento finito na janela principal do programa e funções que controlam a visualização do elemento (dar *zoom*, mover enquadramento, etc.). Os pontos onde devem ser introduzidas as modificações estão indicados nos arquivos fornecidos (apenas os arquivos *prj.c* e *dsp.c* devem ser alterados neste trabalho). A modificação de maior vulto é a criação de uma tabela que associa valores do determinante do Jacobiano aos limites entre as iso-faixas coloridas utilizadas para mostrar a sua variação no domínio do elemento. Este tabela pode ser desenhada no próprio *canvas* de desenho ou pode ser desenhada em um diálogo de interface à parte.

O texto de uma variável *string* de informação, definida dentro do arquivo *prj.c*, deve ser alterado de forma a indicar o autor do trabalho.

A parte do trabalho que compreende a criação da interface gráfica (menus, botões, janelas de desenho, etc.) do programa deve ser implementada com a complementação do arquivo *trab2.led*, que especifica a configuração da interface gráfica na linguagem LED. O arquivo fornecido cria um diálogo de interface onde já são especificados uma área de desenho (*canvas*), uma área de mensagens para o usuário e a associação de teclas do teclado a funções do programa. O que se pede é a criação de menus e botões para acionar as mesmas funções que já são acionadas pelo teclado. Obrigatoriamente devem ser criados menus de barra (no topo do diálogo) e botões adjacentes.

Abaixo estão relacionadas as teclas usadas e as correspondentes funções acionadas:

- “h” (*Help*): Exibe a mensagem de *help* ou sai do *help*.
- “q” (*Quit*): Saida do programa.
- “i” (*Info*): Informações sobre o trabalho.
- “o” (*Open*): Abre um novo arquivo contendo dados de um elemento finito
- “w” (*Write*): Salva dados do elemento finito corrente em um arquivo.
- “p” (*Print*): Imprime o conteúdo da janela de desenho.
- “c” (*Copy*): Copia para o *clipboard* o conteúdo da janela de desenho.
- “s” (*Subdivide*): Define subdivisão do lado do elemento finito para visualização.
- “f” (*Fit*): Ajusta a imagem do elemento de forma a caber totalmente na janela.
- “+” (*Zoom +*): Amplia o modelo na janela de desenho.
- “-” (*Zoom -*): Reduz o modelo a janela de desenho.
- “←” (*Pan Left*): Move o enquadramento da janela de desenho para a esquerda.
- “→” (*Pan Right*): Move o enquadramento da janela de desenho para a direita.
- “↓” (*Pan Down*): Move o enquadramento da janela de desenho para baixo.
- “↑” (*Pan Up*): Move o enquadramento da janela de desenho para cima.

Observe que a complementação da interface do programa (criação de botões, etc.) é para ser feita apenas com modificações no arquivo *trab2.led*, utilizando o programa executável fornecido. Não há necessidade de se recompilar o programa para fazer isso, pois estas complementações são apenas reconfigurações do programa.

## Fornecido e Pedido

Os arquivos para execução do trabalho estão disponíveis dentro de um arquivo comprimido (*civ2801trab2vc6.zip*), que está disponível no *site* da disciplina na *web*. Os seguintes arquivos são fornecidos:

- *trab2.led*: Arquivo em linguagem LED que especifica a configuração da interface gráfica. Este arquivo deve ser complementado como parte deste trabalho.
- *drv.c*: *Driver* que faz a interface do programa para o sistema de interface IUP e que dirige o fluxo de controle do programa. Todas as funções de *callback* registradas junto ao IUP se encontram neste arquivo, que já está completo.
- *prj.c*: Arquivo onde estão localizadas as funções de projeto do programa. Este arquivo deve ser complementado como parte deste trabalho.
- *prj.h*: Definições do módulo *prj.c*.
- *dsp.c*: Arquivo onde estão localizadas as funções para desenhar o elemento finito. Este arquivo deve ser complementado como parte deste trabalho.
- *dsp.h*: Definições do módulo *dsp.c*.
- *contour.c*: Arquivo onde estão localizadas as funções para criar os polígonos correspondentes às iso-faixas de determinante do Jacobiano. Este arquivo já está completo (não precisa ser modificado).
- *contour.h*: Arquivo com protótipos das funções do módulo *contour.c*.
- *elm.c*: Módulo de elemento finito (praticamente igual ao do primeiro trabalho; veja comentários). Este arquivo deve conter as implementações do primeiro trabalho.
- *elm.h*: Definições de *elm.c* (praticamente igual ao do primeiro trabalho).
- *shp.c*: Módulo de funções de forma e jacobiano (igual ao do primeiro trabalho). Este arquivo deve conter as implementações do primeiro trabalho.
- *shp.h*: Definições de *shp.c* (igual ao do primeiro trabalho).
- Alguns arquivos com dados de coordenadas nodais de elementos finitos isoparmétricos.
- Bibliotecas de funções do sistema de interface IUP e do sistema gráfico CD, e os correspondentes arquivos de definições e protótipos, que também estão disponíveis no *site* da disciplina.

Os arquivos *prj.c*, *dsp.c*, *elm.c* e *shp.c* estão incompletos. O trabalho consiste em complementar estes arquivos conforme indicado dentro do código. O entendimento do código existente nestes arquivos é considerado parte do trabalho. Também são fornecidos arquivos contendo dados de um elemento finito para serem lidos pelo programa. Pede-se também, em papel, somente as linhas que foram adicionadas nos arquivos. Estas linhas devem ser indicadas da seguinte forma:

**COMPLETE AQUI: XX**  
**LINHAS ADICIONADAS**

O arquivo *trab2.led* complementado também deve ser entregue.