4
Usando o wx2x2

Em linha gerais, dada uma função F cordata, o **wx2x2** calcula inicialmente seu conjunto crítico $\mathcal{C}$ e coloca-se à disposição do usuário para realizar a tarefa seguinte, que pode ser o cálculo da flor $\mathcal{F}$ ou o cálculo das pré-imagens de algum ponto (ou conjunto de pontos), ou ainda a interação gráfica que permite ao usuário obter uma variedade de informações geométricas e numéricas.

Não seria exagerado afirmar que a quase totalidade dos procedimentos computacionais do programa dedica-se à resolução do sistema

$$F(x) = q, \quad x,\, q \in \mathbb{R}^2. \tag{4-1}$$

Quando o programa resolve esse sistema, ou seja, quando todas as pré-imagens de q são obtidas, dizemos que o ponto q foi invertido ou que q é *um ponto resolvido*.

No caso de funções sabidamente não cordatas, como é o caso de funções que não têm conjunto crítico limitado, o **wx2x2** fornece alternativas, que levam ao estudo detalhado da função em um domínio limitado do plano, determinado pelo usuário.

4.1
A área de trabalho do wx2x2

Na tela inicial do programa, sua *área de trabalho*, o usuário deve apenas fornecer as funções coordenadas da função $F = (F_1, F_2)$ em termos das variáveis x e y e clicar no botão "Get Ready!". À direita da área de trabalho, na área de log, o programa registra um conjunto de informações específicas da função, como mostrado no Capítulo 2. Duas novas janelas são sobrepostas à área de trabalho. A janela frontal à esquerda representa o domínio da função. Nela, vê-se o conjunto crítico $\mathcal{C}$ e as cúspides de F (Figura 4.2a). Na segunda, representando o contradomínio da função, vê-se a imagem do conjunto crítico $F(\mathcal{C})$ (Figura 4.2b).

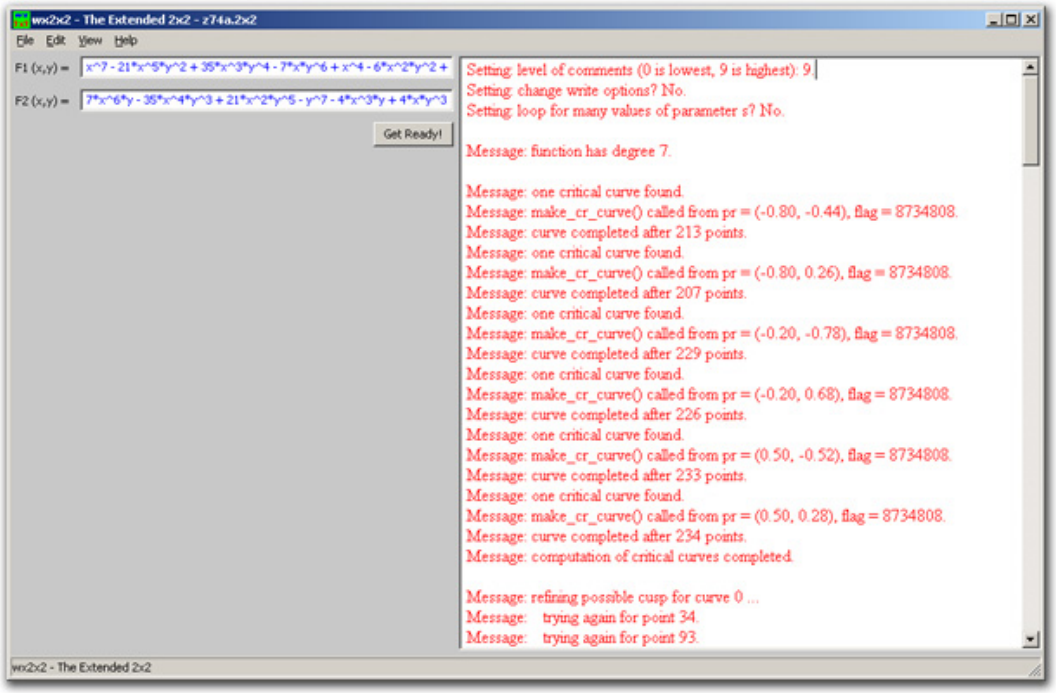
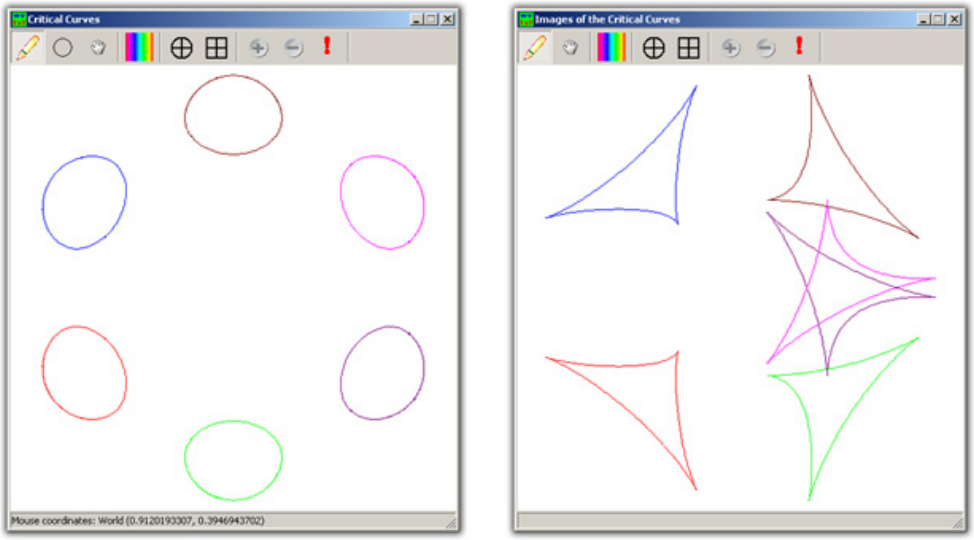


Figura 4.1: Área de trabalho do wx2x2



(a) Conjunto Crítico (b) Imagem do Conjunto Crítico

Figura 4.2: Janelas do Domínio e do Contradomínio

4.2
Modos de Análise: funções próprias e domínios limitados

O **wx2x2** possui três modos distintos para analisar funções do plano no plano. O primeiro, o modo *standard*, cuja interface é exibida na Figura 4.1, trata de funções *cordatas*. Já o segundo, o *modo bump*, exibido na Figura 4.3,

trabalha com funções da forma

$$H(\mathbf{x}) = B(\mathbf{x}) \cdot F(\mathbf{x}) + [1 - B(\mathbf{x})] \cdot G(\mathbf{x}), \quad \mathbf{x} \in \mathbb{R}^2,$$

onde $B : \mathbb{R}^2 \to \mathbb{R}$ é uma *função bump* e H é *cordata*. Finalmente, o terceiro modo é o *mask*, exibido na Figura 4.6, no qual o usuário restringe o domínio da função F a um disco de sua preferência.

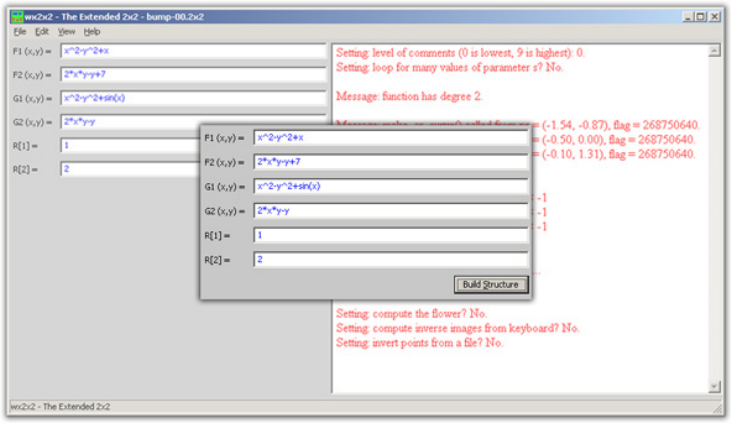


Figura 4.3: Modo Bump

Em princípio, o usuário não sabe que a função de seu interesse é cordata. A propriedade é genérica, para uma topologia adequada, mas ainda assim, para fins de análise numérica, até uma função cordata pode se comportar como se não o fosse — em vários procedimentos, o programa envia uma mensagens de erro causadas por esse fenômeno. Existem parâmetros reguláveis pelo usuário que fazem com que os algoritmos procedam com maior cautela, justamente para poder recuperar, dispondo de precisão suficiente, o fato da função original ser cordata.

Existem casos em que o usuário tem de fato interesse numa função F não cordata. Isso ocorre por exemplo quando o conjunto crítico $\mathcal{C}$ não é limitado: um exemplo quase trivial é a dobra em sua forma normal, $(x, y) \mapsto (x, y^2)$. Nessa situação, convém usar o modo *bump*. Essencialmente, o usuário justapõe a função F à outra função G de sua escolha, obtendo uma função H de forma que, dentro de um disco escolhido pelo usuário, H comporta-se como F, e fora de outro disco devidamente escolhido, como G. O procedimento é transparente ao usuário, depois de fornecidos F, G e os raios dos discos (centrados na origem). O programa define H como a combinação descrita acima de F e G

com pesos determinados por um *bump* B dado por

$$B(\mathbf{x}) = \begin{cases} 1 & 0 \leq ||\mathbf{x}|| \leq R_1 \\ \sum_{i=0}^{7} \alpha_i ||\mathbf{x}||^i & R_1 < ||\mathbf{x}|| \leq R_2 \\ 0 & ||\mathbf{x}|| > R_2 \end{cases}$$

onde os coeficientes α_i's dependem dos raios dos discos e $||\cdot||$ é a norma euclidiana de $\mathbb{R}^2$. A partir daí, o programa opera com H como se fosse uma função cordata.

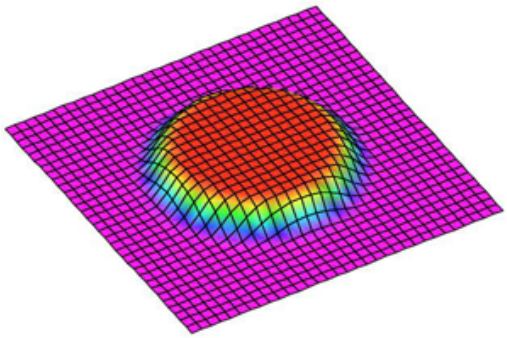


Figura 4.4: Gráfico de uma bump

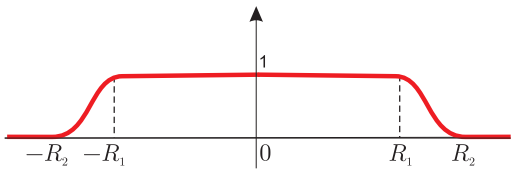


Figura 4.5: Traço no plano $x = 0$

No modo *mask*, o programa invoca o modo *bump* a partir de uma função H obtida amalgamando a função F original com a função G dada pela identidade. O raio interno é dado pelo usuário e o externo vale 1.5 vezes o interno. Nas janelas gráficas, são mostradas apenas as informações contidas no disco escolhido pelo usuário.

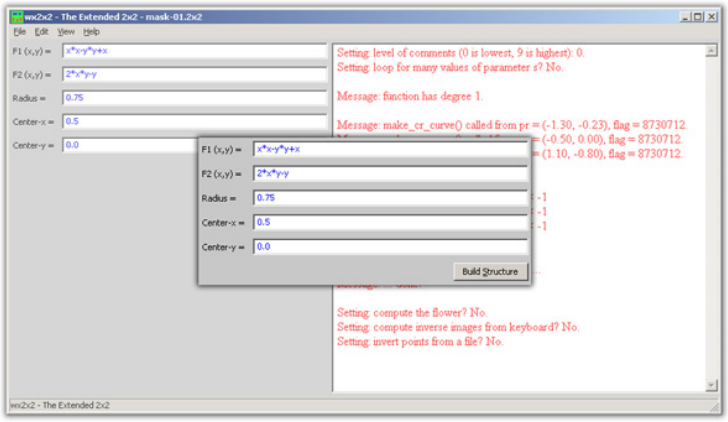


Figura 4.6: Modo mask

4.3 Barras de ferramenta nas janelas de domínio e contradomínio

No alto das figuras 4.2a e 4.2b encontra-se a barra de ferramentas das janelas do domínio e contradomínio. Suas funções são descritas a seguir.

(a) Barra de Ferramentas da janela do domínio

(b) Barra de Ferramentas da janela do contradomínio

Figura 4.7: Barras de Ferramentas

1- Pencil: No domínio, é usada para desenhar um ponto e, automaticamente, sua imagem é calculada pelo programa. Na imagem, desenha um ponto e desenha suas pré-imagens, também calculadas pelo programa.

2- Circle: Desenha uma circunferência e sua imagem. O círculo é dado clicando seu centro e arrastando o cursor a um ponto da circunferência.

3- Hand: Translada a janela de visualização.

4- Forecolor: Permite escolher uma cor para o *Pencil* e para *Circle*.

5- Center: Exibe ou oculta uma pequena marca que indica o centro da janela. É útil para posicionar objetos ao realizar *zooms*.

6- Axes/Grid: Na janela do domínio, exibe a malha utilizada na detecção das curvas críticas (v. seção 5.3); no contradomínio, exibe os eixos cartesianos (v. Figura 4.8).

7, 8- Zooms In e Out: Amplia/Reduz a janela de visualização, fixando o centro da janela.

9- Reset: Exclui tudo o que foi desenhando através do *Pencil*, nas duas janelas ao mesmo tempo.

Todas as ferramentas são acessíveis por meio de teclas de atalho, indicadas na Tabela 4.1.

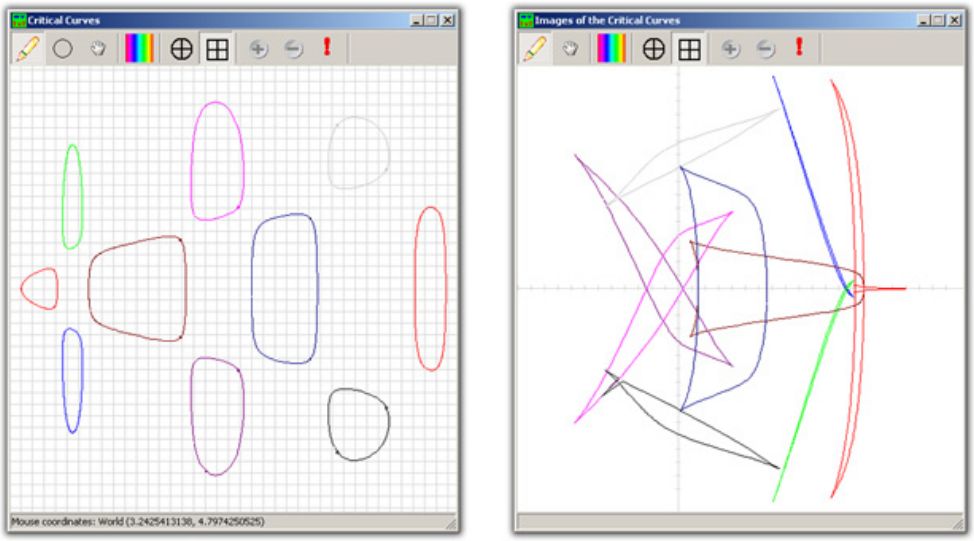


Figura 4.8: A ferramenta Axes/Grid e seus comportamentos

Ferramenta	Tecla de Atalho
Pencil	P ou p
Circle	E ou e
Hand	H, h ou a Barra de espaços pressionada
Forecolor	F ou f
Center	V ou v
Axes/Grid	A ou a
Zoom In	+
Zoom Out	−
Reset	R ou r

Tabela 4.1: Teclas de Atalho

4.4 Parâmetros Globais

O programa permite o ajuste de vários parâmetros globais, que determinam a robustez de vários processos numéricos. Na opção *Preferences* do menu *Edit*, o programa exibe uma janela de diálogo onde o usuário pode ajustar esses parâmetros, descritos a seguir. Na guia *General*, encontram-se

Level of comments: Um número inteiro entre 0 e 9, que determina o nível de detalhe das mensagens exibidas na área de log. No Capítulo 2, mostramos um exemplo onde este parâmetro estava definido com o valor máximo.

Blank: Vale 0 ou 1. Habilita ou desabilita os testes de *Blank-Troyer* (veja Capítulo 4 de [MST2] para detalhes).

Number of tries: Quando o teste de contagem ou o de Blank-Troyer reprova uma placa X do domínio, assim estabelecendo que falta uma curva crítica nessa placa, o programa passa a procurar dois pontos distintos p e q em X nos quais F tem orientações opostas. Isso garante a existência de um ponto crítico no segmento entre p e q. O parâmetro indica o número de tentativas para localizar p e q (veja 5.3.4).

Banksys: O programa armazena o conjunto de pré-imagens de alguns pontos num banco de pontos resolvidos. Banksys igual a -1 indica que todo ponto p cujas pré-imagens são calculadas serão armazenados nesse banco. Se Banksys é igual 1, um ponto p é armazenado só quando o segmento terminando em p invertido na imagem intercepta $F(\mathcal{C})$. Quando Banksys é 0, o banco de pontos resolvidos guarda apenas quatro pontos especiais. (v. Capítulo 5).

Number of neigh: Um número inteiro positivo menor ou igual à constante *MAX_NUM_NEIGH*. Especifica, no processo de inversão de um ponto, a quantidade de caminhos de inversão em forma de L, contados a partir de comprimento mínimo, que serão considerados na rotina que procura um caminho adequado de inversão (v. Capítulo 5). Por *default*, *MAX_NUM_NEIGH*$= 40$.

Os parâmetros da guia *Step Settings* são utilizados nas rotinas que constroem curvas críticas. No **wx2x2** existem duas rotinas principais para essa tarefa, *make_cr_curve()* e *next_zero()*, que são rotinas numéricas de passo variado controladas por:

1- Curve Sharpness: um valor no intervalo $[0, 1)$. Quanto mais próximo de 1, mais fina é a discretização no cálculo de curvas críticas. A Figura 4.9 exibe os conjuntos $\mathcal{C}$ e $F(\mathcal{C})$, da função $(x, y) \mapsto (x^3 - 3xy^2 + 2.5x^2 - 2.5y^2 + x, 3x^2y - y^3 - 5xy + y)$, $\widetilde{F}(z) = z^3 + 2.5\bar{z}^2 + z$, para dois valores distintos do parâmetro.

2- Minimum step e Maximum step: passos extremos empregados na rotina *next_zero()*.

3- Min step in make_cr_curve(): menor passo na rotina *make_cr_curve()*.

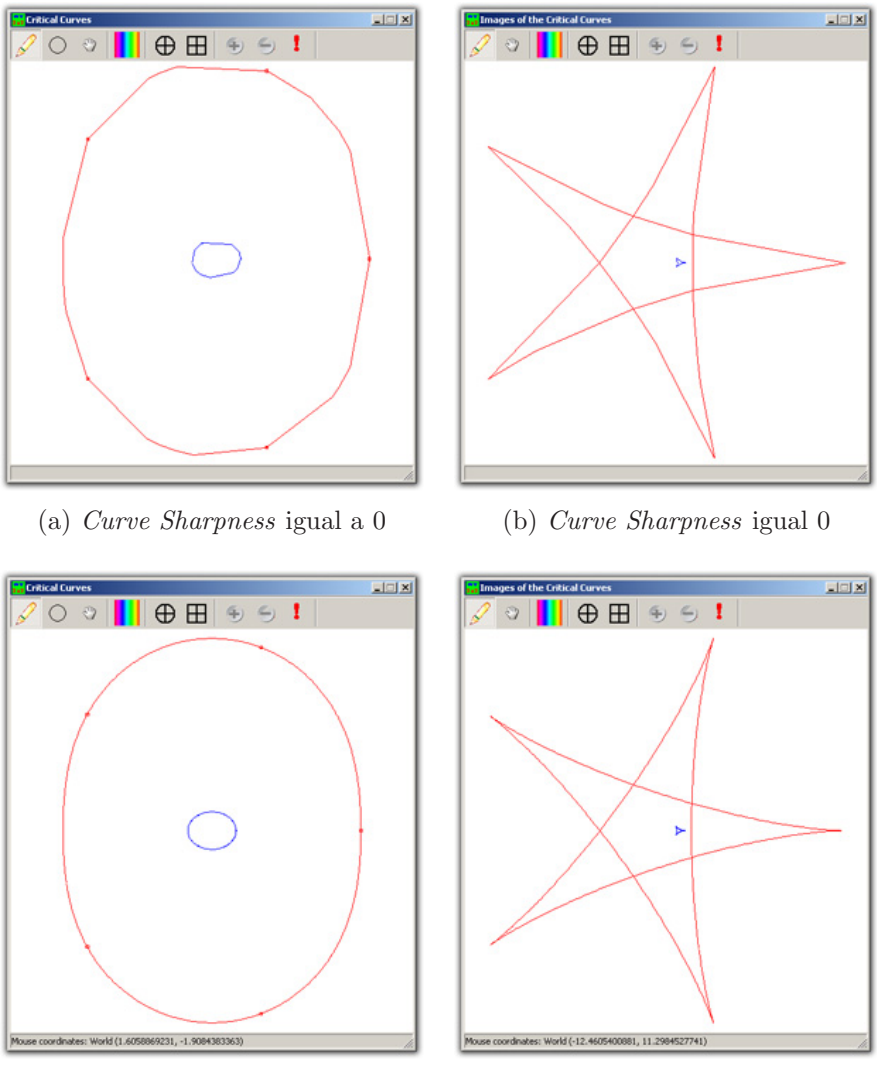
A guia *Grid Settings* contém os parâmetros que determinam o tamanho e a densidade da malha quadrangular centrada na origem utilizada na busca inicial por pontos críticos. Essa malha é uniforme, com espaçamento *Grid step* paras as NGRID colunas e linhas mais internas e expande-se geometricamente a partir daí, com razão *Grid ratio*.

1- NGRID: Determina o número de colunas e linhas da malha. A malha possui $2 \cdot \text{NGRID}$ colunas e $2 \cdot \text{NGRID}$ linhas.

2- Grid step: Espaçamento entre as NGRID linhas e colunas da malha próximo à origem.

3- Grid ratio: Razão da progressão geométrica que determina o espaçamento entre as NGRID linhas e colunas mais afastadas da origem.

Por fim, a guia *GUI Settings* possui o parâmetro *Curve frame width* que determina os tamanhos das janelas do domínio o do contradomínio.



(a) *Curve Sharpness* igual a 0 (b) *Curve Sharpness* igual 0

(c) *Curve Sharpness* igual a 0.95 (d) *Curve Sharpness* igual a 0.95

Figura 4.9: Exemplos de configurações do parâmetro *Curve Sharpness*