

Visualização de Funções na Ferramenta *OpenDX*

Marilyn Menecucci Ibañez,

Universidade Federal de Itajubá

PIBIC/INPE-CNPq Laboratório Associado de Computação e Matemática Aplicada, CTE, INPE,

Av. dos Astronautas, 1758, 12227-010 - Jardim da Granja, São José dos Campos - SP

E-mail: marilyn_unifei@yahoo.com.br,

Margarete Oliveira Domingues, **Stephan Stephany,**

Laboratório Associado de Computação e Matemática Aplicada, CTE, INPE,

Av. dos Astronautas, 1758, 12227-010 - Jardim da Granja, São José dos Campos - SP

E-mail: margarete@lac.inpe.br, stephan@lac.inpe.br,

Odim Mendes Jnior

Divisão de Geofísica Espacial, CEA, INPE,

Av. dos Astronautas, 1758, 12227-010 - Jardim da Granja, São José dos Campos - SP

E-mail: odim@dge.inpe.br,

Resumo

As ferramentas de visualização científica são bastante aplicadas no estudo de funções matemáticas. Este artigo apresenta a utilização da ferramenta *OpenDX* no estudo de funções multidimensionais.

Palavras-chave

Visualização científica, data explorer, opendx

Introdução

A visualização científica é bastante utilizada nas diversas áreas da ciência. A sua aplicação nas áreas de engenharia, física e matemática para a visualização de funções multidimensionais retrata uma das suas utilizações frequentes.

O *OpenDX* é um programa gratuito de código aberto, semelhante e derivado do Data Explorer da IBM [3] e constituído por um conjunto de ferramentas de manipulação, processamento, visualização e animação gráfica. Além disso, ele possui um grande potencial de transformação de dados e visualizações de funções que é utilizado em diversas áreas científicas como Ciências Espaciais, Meteorologia, Oceanografia, entre outras.

O programa contém diversos ambientes de programação que permitem ao usuário ter uma grande flexibilidade no momento de criar uma visualização. Esses ambientes se dividem em um editor de programas visuais (VPE), que é o ambiente gráfico do programa; e um módulo constituído por um script, que permite ao usuário ter o controle do programa de visualização por meio de um prompt. O script do programa possibilita que o usuário estenda as funcionalidades do programa adicionando módulos

por meio de bibliotecas, assim como, criando seus próprios módulos [2].

Este trabalho focaliza o problema da visualização de funções bidimensionais e sua utilização como auxílio a análise de comportamentos e comparações entre essas funções. O caso de estudo é a análise de dispersão de ondas em um modelo de água rasa e em suas respectivas verses discretas nas grades de Arakawa e Lamb [7]. Para essa tarefa utiliza-se a ferramenta de visualização científica de *OpenDX* em ambiente de programação.

Metodologia

O *OpenDX* apresenta um menu inicial, como mostra a Figura 1, que permite importar dados, editar um arquivo .net já existente, criar um arquivo .net por meio do VPE, visualizar um exemplo do banco de dados do *OpenDX* ou visualizar o tutorial do programa. Para acionar o ambiente gráfico - o VPE, clica-se na opção *New Visual Program*.

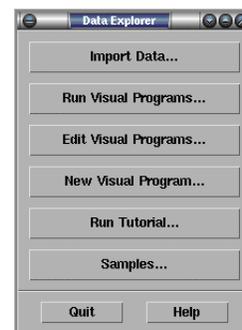


Figura 1: Janela do *Data Explorer*

A janela do VPE bem como suas funções são apresentadas na Figura 2.

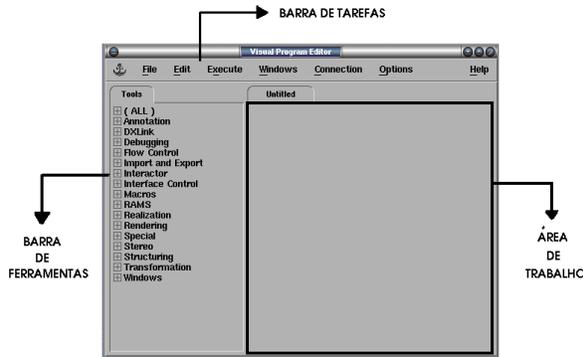


Figura 2: Exemplo da janela do *Visual Program Editor*

Para visualizar as funções bidimensionais no *OpenDX* é necessário construir uma estrutura de um programa de visualização utilizando as ferramentas disponíveis no VPE. Na construção da malha das funções utiliza-se a ferramenta *Construct* que permite construir objetos utilizando coordenadas do eixo cartesiano. Esta ferramenta, como todas do *OpenDX*, possui uma tela de configuração como mostra a Figura 3 que precisa ser configurada para se utilizar de forma adequada a ferramenta.

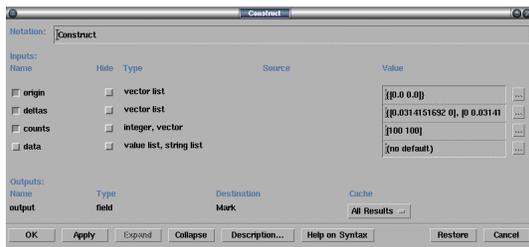


Figura 3: Tela de configuração da ferramenta *Construct*.

Na tela de configuração da ferramenta *Construct*, para se especificar a origem dos eixos de coordenadas cartesianas como o ponto (0,0) utiliza-se a opção *origin* e para se especificar a variação dos eixos como π utiliza-se a opção *delta*. Na opção *count* especifica-se a quantidade de pontos em cada eixo, neste caso, 100 pontos. Nas outras opções utiliza-se os valores *default* do *OpenDX*

Na continuidade da construção da estrutura do programa de visualização utiliza-se a ferramenta *Mark* que permite selecionar qual componente de um campo deve ser selecionada, neste caso *position*, por meio da opção *name* da sua tela de configuração. A

ferramenta utilizada *Compute2* permite especificar a função que se deseja visualizar, como por exemplo

$$(\text{sqrt}(1 + L * (\sin(a.x) + \sin(a.y))))$$

e seus parâmetros de entrada por meio das opções *name*, nome do parâmetro, e *input*, dado de entrada. O dado de entrada pode ser digitado diretamente na opção *input* ou ser uma outra ferramenta conectada, como a ferramenta *Scalar* que apresentada como saída um número inteiro, que pode ser especificado em um Painel de Controle como mostra a Figura 4.

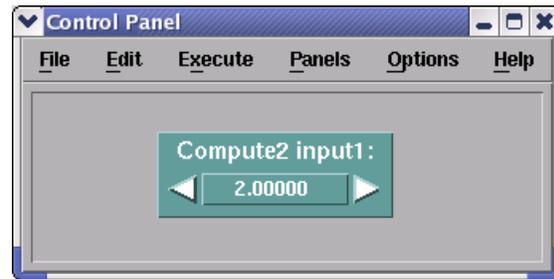


Figura 4: Tela do Painel de Controle da ferramenta *Scalar*.

Para a criação da superfície que representa a função utiliza-se a ferramenta *RubberSheet* que deforma uma variável de superfície com base na quantidade de valores de dados em cada ponto da superfície gerando uma visualização 3D. Na coloração da superfície utiliza-se a ferramenta *AutoColor*.

As funções também são visualizadas como contornos que representam uma estimativa de 10% de erro, utilizando no lugar da ferramenta *RubberSheet* as ferramentas *Compute*, que calcula a porcentagem de erro, e *Isosurface* que cria as linhas de contorno. Também se utiliza o conjunto de ferramentas *Construct*, *Mark* e *Compute2* em que nesta última tem se como função de entrada $\text{sqrt}(1+L*L*(a.x*a.x+a.y*a.y))$. Este conjunto de ferramentas é um dado de entrada da ferramenta *Compute2* explicada anteriormente, que divide a equação para auxiliar no cálculo do erro. Na coloração do contorno utiliza-se a ferramenta *Color* que difere da ferramenta *AutoColor* por permitir selecionar a cor.

Como se estava representando gráficos de funções havia a necessidade de construir eixos de coordenadas cartesianas, que podem ser construídos utilizando-se a ferramenta *AutoAxes*, que cria os eixos por meio de um dado de entrada. Também utiliza como entrada a ferramenta *AutoCamera* que permite representar a direção do eixo. Na coloração dos eixos utiliza-se a ferramenta *Color*.

Para unir-se a visualização das funções com os eixos utiliza-se a ferramenta *Collect*. Já as ferramentas *Translate* e *Rotate* são utilizadas para movimentar e rotacionar a imagem, por meio de dados de

eixos de coordenadas cartesianas. E por fim, a ferramenta *Image* é utilizada para gerar a visualização resultante.

Aplicação e Resultados

Um modelo de água rasa bidimensional no plano f pode ser descrito pelo seguinte sistema de equações

$$\frac{\partial u}{\partial t} - fv + g\frac{\partial h}{\partial x} = 0 \quad (1)$$

$$\frac{\partial v}{\partial t} + fu + g\frac{\partial h}{\partial y} = 0 \quad (2)$$

$$\frac{\partial h}{\partial t} + \mathcal{H} \left[\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} \right] = 0 \quad (3)$$

em que $u = u(x, y, t)$ é a componente zonal do vento, $v = v(x, y, t)$ é a componente meridional do vento, $h = h(x, y, t)$ é o deslocamento da coluna do fluido em questão, \mathcal{H} é a altura média da coluna do fluido, g é a aceleração da gravidade e f é o parâmetro de Coriolis, conforme descrito em [8]. Neste modelo \mathcal{H} , g e f são considerados constantes.

Assumindo-se soluções do tipo onda plana

$$u = U(t)e^{i(\xi x + \eta y + \omega t)}, \quad v = V(t)e^{i(\xi x + \eta y + \omega t)},$$

$$h = H(t)e^{i(kx + \ell y + \omega t)},$$

nas Equações 1, 2 e 3 obtêm-se a relação de dispersão não trivial desse modelo:

$$\left(\frac{\omega}{f}\right)^2 = 1 + \left(\frac{\lambda}{d}\right)^2 [(kd)^2 + (\ell d)^2], \quad (4)$$

em que $\lambda = \frac{\sqrt{gH}}{f}$.

Considerando as grades de [7] esquematizadas na Figura 5, as relações de dispersão desse modelo são expressas a seguir:

Grade A:

$$\left(\frac{w}{f}\right)_A^2 = 1 + \left(\frac{\lambda}{d}\right)^2 \left[\sin\left(\frac{kd}{2}\right) + \sin\left(\frac{\ell d}{2}\right) \right]$$

Grade B:

$$\left(\frac{w}{f}\right)_B^2 = 1 + 2\left(\frac{\lambda}{d}\right)^2 [1 - \cos(kd) \cos(\ell d)]$$

Grade C:

$$\left(\frac{w}{f}\right)_C^2 = \cos^2\left(\frac{kd}{2}\right) \cos^2\left(\frac{\ell d}{2}\right) + 4\left(\frac{\lambda}{d}\right)^2 \left[\sin^2\left(\frac{kd}{2}\right) \sin^2\left(\frac{\ell d}{2}\right) \right],$$

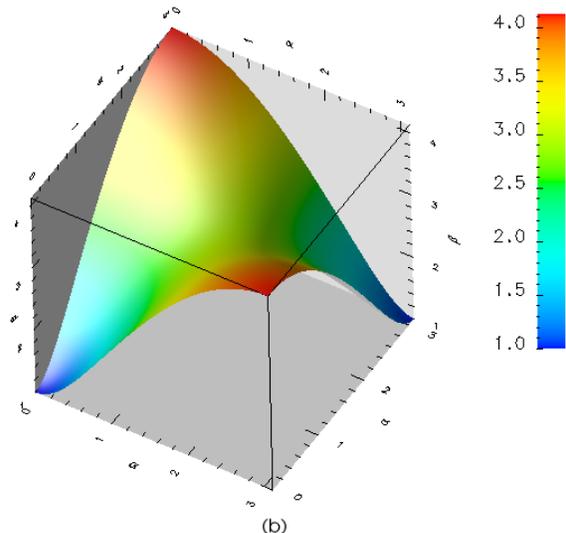
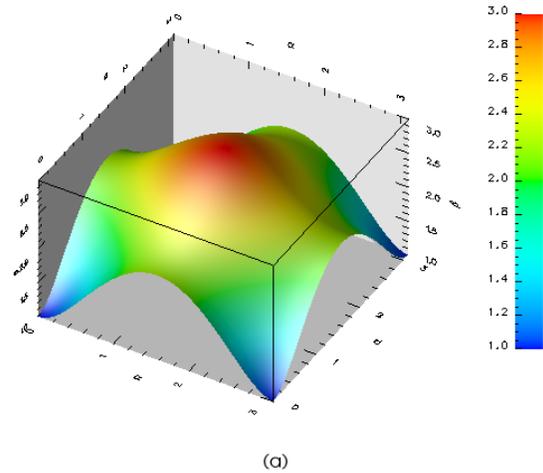
Grade D:

$$\left(\frac{w}{f}\right)_D^2 = \cos^2\left(\frac{kd}{2}\right) \cos^2\left(\frac{\ell d}{2}\right) + \left(\frac{\lambda}{d}\right)^2 \left[\cos^2\left(\frac{kd}{2}\right) \sin^2(\ell d) + \cos^2\left(\frac{\ell d}{2}\right) \sin^2(kd) \right]$$

Grade E:

$$\left(\frac{w}{f}\right)^2 = 1 + 2\left(\frac{\lambda}{d}\right)^2 \left(\sin^2\left(\frac{kd}{\sqrt{2}}\right) + \sin^2\left(\frac{\ell d}{\sqrt{2}}\right) \right)$$

Para cada uma das relações de dispersão descritas é construída uma estrutura de um programa de visualização. A Figura 6 mostra a estrutura completa do programa de visualização e as Figuras 5 (a), (b), (c), (d) e (e) as visualizações resultantes.



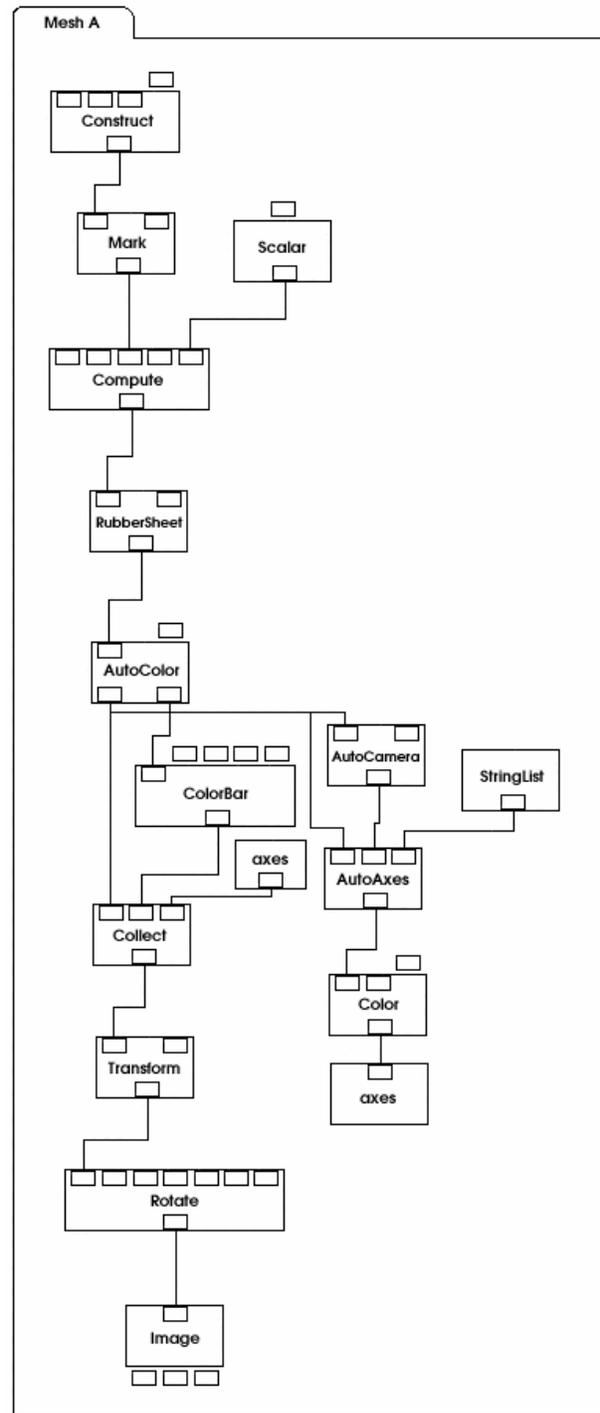
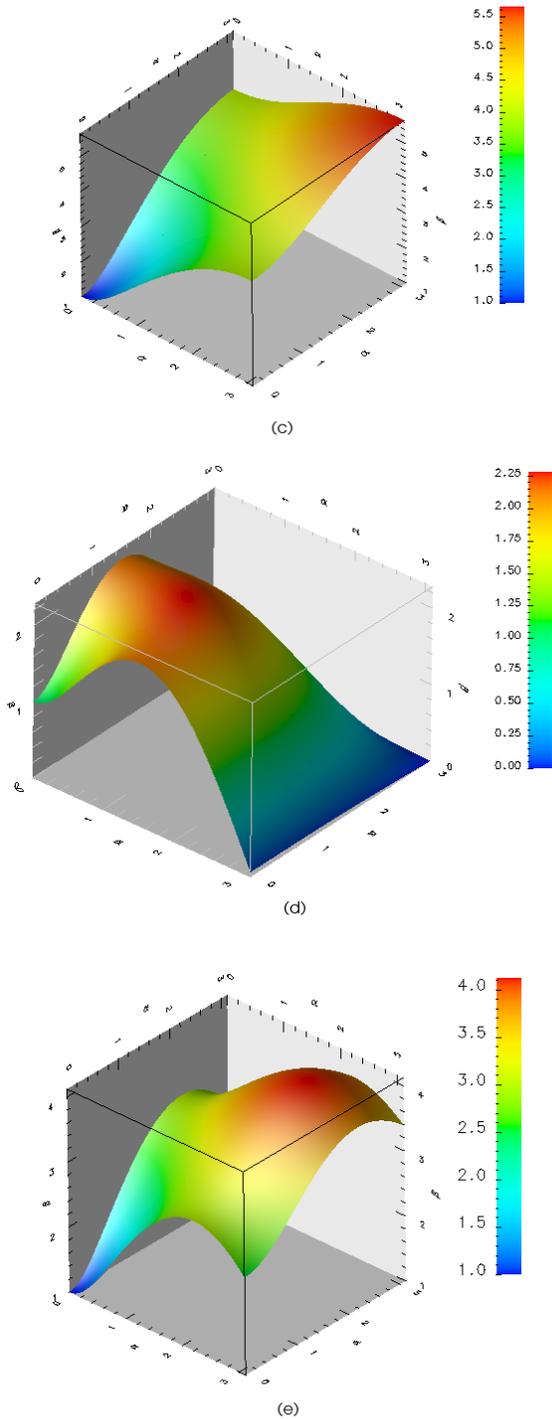


Figura 5: Visualização resultante do termo (ω/f) para: (a) Grade A (b) Grade B (c) Grade C (d) Grade D (e) Grade E.

Figura 6: Exemplo da estrutura do programa de visualização que representa a superfície de uma função.



Na construção dos contornos que representam 10% de erro de discretização, também contruiu-se para cada função uma estrutura de programa de visualização e uniu-se cada uma dessas estruturas, por meio de ferramentas de transmissão (*Transmitter*) e de recepção (*Receiver*), para gerar uma visualização comparativa das funções. A Figura 7 (a) apresenta a estrutura do programa de visualização da função exemplifica na Grade A e a 7 (b) a união das estruturas de cada função. A Figura 8 (a) mostra a visualização resultante da função da Grade A e a 8 (b) mostra a visualização resultante da união de todas as funções.

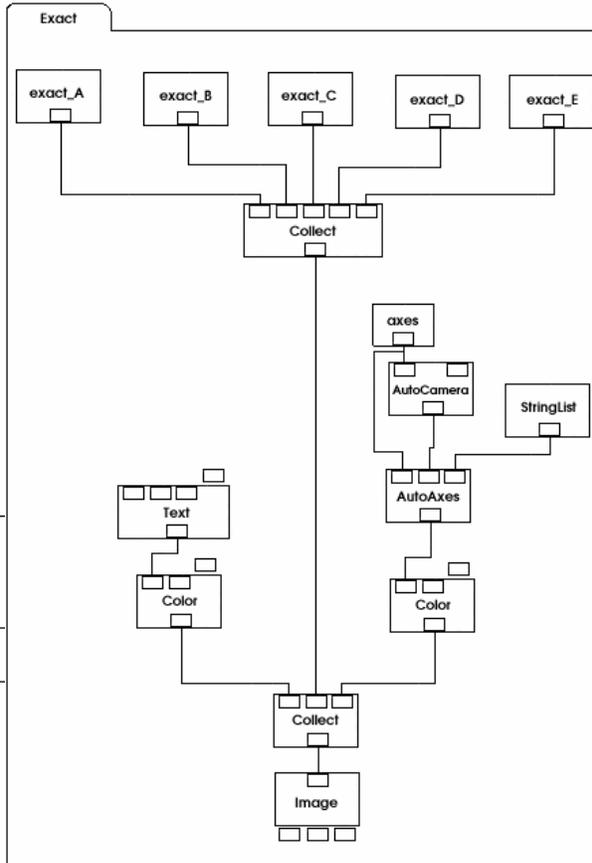
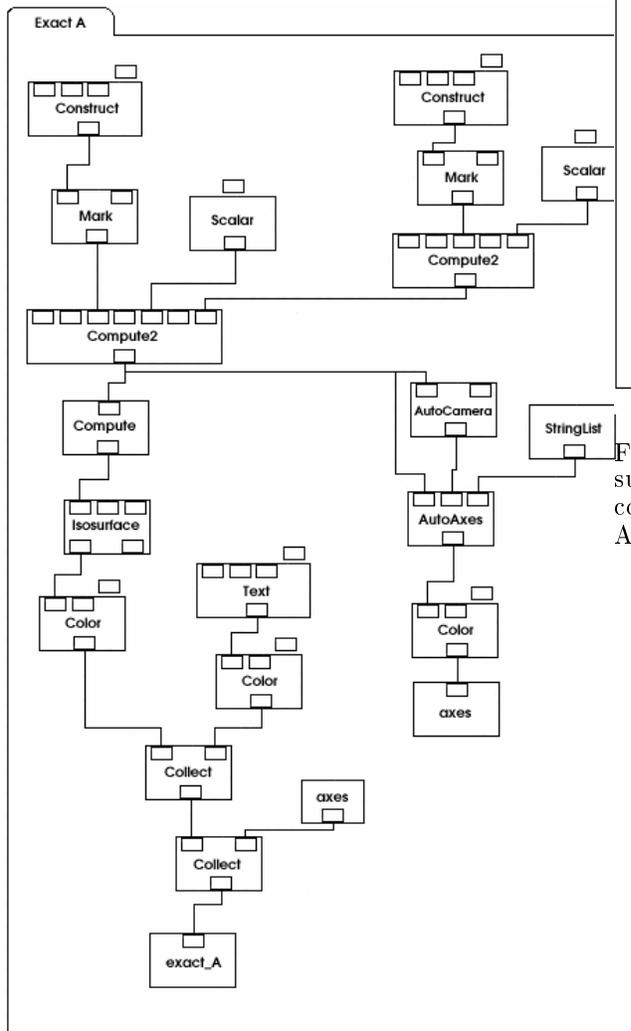
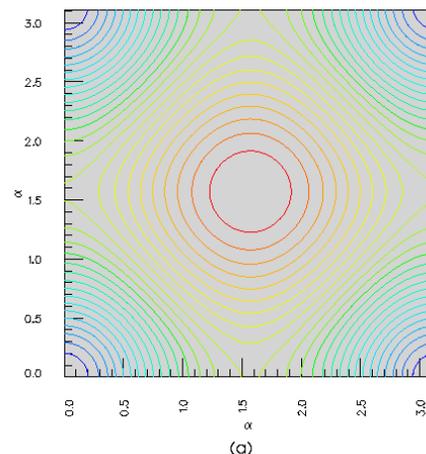


Figura 7: Exemplo da estrutura do programa de visualização de uma função que representa o contorno com 10% de erro: (a) Programa da função da Grade A (b) União de todas as funções.



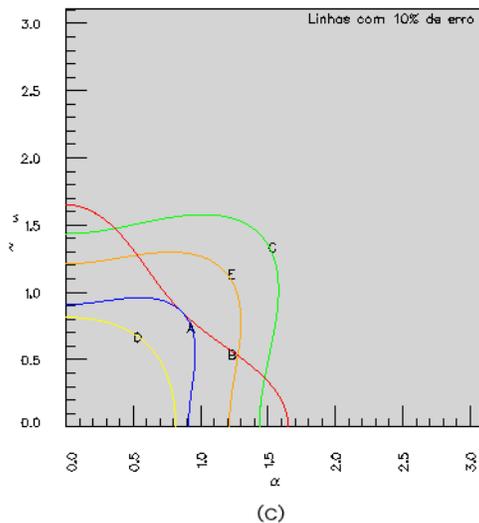
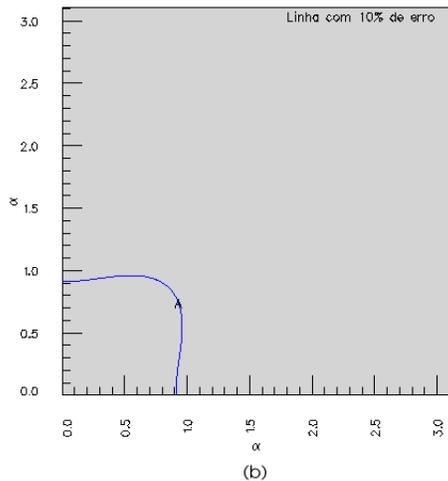


Figura 8: Visualização resultante: (a) Função da Grade A (b) Função da Grade A representando 10% de erro (c) Conjunto de todas funções representando 10% de erro.

Referências

- [1] Blaz, R.; Domingues, M. O.; Mendes, O. Jr.; Introdução à entrada de Dados no *OpenDX*: Formatos *dx*, *.general* e *.grb*. Brasil: INPE, 2003 53p.
- [2] Thompson, D.; Braun, J.; Ford, R. *OpenDX Path to Visualization*. Estados Unidos: Visualization and Imagery Solution, Inc, 2001. 207p. <<http://www.vizsolutions.com>>.
- [3] IBM Visualization data explorer. Programmer's reference. SC38-0486-03. IBM, Estados Unidos, maio 1997a. <<http://www.opendx.org>>.
- [4] IBM Visualization data explorer. quickstar Guide. SC34-3262-02. IBM, Estados Unidos, setembro 1997b. <<http://www.opendx.org>>.
- [5] Página oficial do *OpenDX* <<http://www.opendx.org>>.
- [6] Página do Data Explorer <<http://www.research.ibm.com/DataExplorer>>.
- [7] Arakawa A., Lamb, V. Computational design of the Basic Dynamical Processes of UCLA General Circulation Model. In: *Methods of Computational Physics*. vol. 17, Academic Press, New York (1977), p. 173.
- [8] Dukowicz, J. K. Mesh Effects for Rossby waves. *Journal of Computational Physics*, 119,188-194(1995).