



MINISTÉRIO DA CIÊNCIA E TECNOLOGIA
INSTITUTO NACIONAL DE PESQUISAS ESPACIAIS

INPE-10067-PRE/5612

**VISUALIZAÇÃO CIENTÍFICA DOS RESULTADOS DE MÉTODOS
HÍBRIDOS DE WAVELES + DIFERENÇAS FINITAS**

Roberto Blaz*
Marilyn Menecucci Ibañez**
Margarete Oliveira Domínguez

*UNIB – bolsista PIBIC/CNPq
** UMC – bolsista PIBIC/CNPq

Seminário de Iniciação Científica do INPE (SICINPE).

VISUALIZAÇÃO CIENTÍFICA DOS RESULTADOS DE MÉTODOS HÍBRIDOS DE WAVELES + DIFERENÇAS FINITAS

Roberto Blaz (UNIB, Bolsista PIBIC/CNPq)
Marilyn Menecucci Ibañez (UMC, Bolsista PIBIC/CNPq)
Dra. Margarete Oliveira Domingues (CPTEC/INPE)

RESUMO

Em Hidrodinâmica são comuns as situações em que escoamentos apresentam estruturas com variações bruscas. Para uma representação adequada dessas estruturas são necessárias grades muito refinadas sendo que em muitos casos, essas estruturas possuem uma localização espacial reduzida em comparação com o domínio do escoamento. Por essa razão, há um interesse por métodos numéricos adaptativos que usem subgrades mais refinadas apenas nessas regiões e subgrades menos refinadas nas regiões em que o escoamento seja mais suave. Este trabalho tem o objetivo de visualizar dados de um desses modelos de adaptabilidade espacial, de forma a minimizar os esforços computacionais de reconstrução dessas subgrades em sua grade regular mais refinada. Esses dados são setorizados em blocos de dados contendo sua posição relativa a grade mais refinada e seu nível de refinamento. Para atender a metodologia de trabalho, o pacote de visualização científica OpenDX foi escolhido, constituindo em um recurso potente de visualização multidimensional, com código aberto e de distribuição gratuita. Na implementação da metodologia aprendeu-se a manipular, processar, transformar, criar processos de visualização e animar dados sob condições multiplataforma. Com esses recursos, preparou-se uma forma de acessar a estrutura de dados em um dos formatos próprios do OpenDX, dessa forma os diversos blocos de dados puderam ser reposicionados na malha original de acordo com a sua posição e refinamento. Após esse procedimento foi desenvolvido um programa de visualização desses dados no ambiente gráfico do OpenDX. Como resultado deste trabalho foi possível visualizar o escoamento de interesse sem a reconstrução da malha mais refinada e sem perda de informação visual. Foram realizadas visualizações de dados de modelos numéricos de previsão de tempo e de dados observacionais de descargas elétricas. Foi possível também realizar a documentação de entrada de dados e iniciar a documentação do ambiente programação de visual . A conclusão desta etapa é que o domínio do uso dessa ferramenta mostra ser muito útil para visualizações de dados de modelos numéricos e dados observacionais, tão usuais nesse ambiente científico.

Aluno do Curso de Ciência da Computação, UNIB. E-mail: blazz@bol.com.br

Aluno do Curso de Engenharia da Computação, UNIB. E-mail: marilyn_mba@yahoo.com.br

Pesquisadora do Centro de Previsão de Tempo e Estudos Climáticos E-mail: margaret@cptec.inpe.br



MINISTÉRIO DA CIÊNCIA E TECNOLOGIA
INSTITUTO NACIONAL DE PESQUISAS ESPACIAIS

Visualização Científico dos Resultados de Métodos Híbridos de Wavelets + Diferenças Finitas

**RELATÓRIO FINAL DE PROJETO DE INICIAÇÃO CIENTÍFICA
(PIBIC/CNPq/INPE)**

**Marilyn Menecucci Ibañez(Bolsista PIBIC/CNPq)
E-mail:marilyn_mba@yahoo.com.br**

**Dra. Margarete Oliveira Domingues (CPTEC/INPE, Orientadora)
E-mail: margaret@cptec.inpe.br**

Maio de 2003

Relatório Atividades

PIBIC/INPE

Bolsista: Marilyn Menecucci Ibañez

Orientadora: Margarete Oliveira Domingues

Instituição Financiadora: CNPq

Instituição Coordenadora: INPE

Período: abr/2003 a jul/2003

INTRODUÇÃO

Este relatório refere-se às atividades desenvolvidas no projeto "Visualização Científica dos Resultados de Métodos Híbridos de Wavelets + Diferenças Finitas" do PIBIC-CNPq/INPE, no Laboratório Associado de Meteorologia e Oceanografia (LMO) do Centro de Previsão de Tempo e Estudos Climáticos (CPTEC) do INPE.

A visualização científica é de grande importância para a apresentação dos resultados de modelos numéricos e dados meteorológicos e oceanográficos, especialmente em representações bidimensional e tridimensional.

O objetivo científico do trabalho proposto é estender as facilidades do poderoso pacote gratuito de visualização científica OPENDX a resultados numéricos de modelos e dados meteorológicos. Isso resulta no benefício de rápida interpretação de resultados, possibilita a aplicação a outros fenômenos pertinentes; e facilita a utilização dessas informações pela comunidade.

FASE ATUAL DO PROJETO

As atividades desenvolvidas nesses dois meses de etapa inicial foram de introdução ao problema a ser implementado e de complementação da formação acadêmica do bolsista. Essas etapas estão de acordo com o plano de trabalho proposto. Além da parte de complementação acadêmica na parte computacional do estudante, a forma de redação e apresentação de relatórios técnicos e o início da implementação de programas visuais no OPENDX também foram desenvolvidos. Maiores detalhes são apresentados no relatório técnico anexo.

Para facilitar o entendimento dos processos de entrada e visualização de dados no OPENDX, procedimentos com complexidades, o bolsista dedicou-se a esse tópico, neste período de bolsa. Como resultado desse trabalho e para difundir as facilidades de programação visual no OPENDX por outros usuários, o bolsista está redigindo um relatório técnico sobre esse assunto. Esse relatório é apresentado anexo, em versão preliminar.

Com a familiaridade iniciada com o OPENDX nesta etapa, o bolsista já está desenvolvendo programas no ambiente de programação visual, ainda que de forma modesta.

COMPLEMENTAÇÃO ACADÊMICA DO BOLSISTA

Foram estudados aspectos computacionais gerais e específicos para o desenvolvimento do trabalho proposto. A seguir são apresentadas em linhas gerais as etapas já realizadas.

A. Aspectos Computacionais Gerais

Foram estudados tópicos básicos para execução do projeto

- 1) Introdução ao sistema operacional GNU/LINUX e as implicações de licenças GNU;
- 2) Noções de Programação em GNU/LINUX : compiladores GNU/gcc, utilitário GNU/make, depurador GNU/gdb, ferramentas de controle de versão como o rcs, e editores de texto mais utilizados como o kwrite.
- 3) Noções de outras ferramentas de visualização científicas e de tratamento de imagens, em especial do GNU/gnuplot;
- 4) Introdução ao \LaTeX e suas facilidades de editoração digital;

B. Aspectos Computacionais Específicos

- 1) Introdução a entrada de dados no OPENDX ;
- 2) Introdução ao ambiente de programação visual do OPENDX ;
- 3) Introdução a linguagem de programação visual do OPENDX e suas facilidades;
- 4) Introdução a linguagem de programação C

Outras Atividades Acadêmicas

O bolsista também participou do "curso de introdução a ferramentas fisico-matemáticas de livre distribuição" ministrado no setor de treinamento INPE para alunos da Pós Graduação em Meteorologia e Ciências Espaciais em março de 2003 (certificado anexo).

COMENTÁRIOS FINAIS

Nesta fase do projeto as atividades de complementação de formação acadêmica do bolsista foram as atividades principais desenvolvidas. Ocorreram também avanços nas etapas de entendimento da visualização de dados no OPENDX .

Planos para o 2ª parte do Projeto

Nesta fase dar-se-á ênfase aos aspectos teóricos do projeto:

- a) Rudimentos de eletricidade atmosférica;
- b) Como são gerados os dados que serão visualizados, como se enquadra seu trabalho de Iniciação Científica no projeto de visualização de dados, e qual a sua importância nesse contexto;

- c) Forma de apresentação escrita de relatórios e artigos (cont.);
- d) Formas de apresentação oral, uso de ferramentas disponíveis;
- e) Introdução a outros problemas de visualização científica de interesse em Meteorologia.

Será estudado em detalhes a forma de entrada de dados de malhas regulares por blocos e implementados programas visuais para apresentação desses dados, como a representação evolutiva, visualização simultânea da malha e dos resultados dos modelos. Geração de saídas estáticas e dinâmicas desses resultados.

Cronograma:

- Formação teórica do Bolsista: Mar/2002–Abr/2002
- Treinamento: Ago—Out/2003
- Desenvolvimento dos programas: Nov/2003—Fev/2004
- Documentação dos Resultados: Mar-Mai/2004
- Preparação dos resultados para publicação: Mai—Jul/2004.

São José do Campos, 30 de maio de 2003.



MINISTÉRIO DA CIÊNCIA E TECNOLOGIA
INSTITUTO NACIONAL DE PESQUISAS ESPACIAIS

INPE

VISUALIZAÇÃO DE DADOS NO OPENDX

Marilyn Menecucci Ibañez — bolsista PBIC/INPE
Dra. Margarete Oliveira Domingues — orientadora

Versão Preliminar
Distribuição restrita

INPE
São José dos Campos
2002

CAPÍTULO 1

Introdução

O OPENDX é um programa gratuito de código aberto, semelhante ao Data Explorer da IBM e constituído por um conjunto de ferramentas de manipulação, processamento, visualização e animação gráfica. Além disto, ele possui um grande potencial de transformação de dados que é utilizado em diversas áreas científicas como Meteorologia, Ciências Espaciais, Oceanografia, entre outras. Para se obter maiores informações sobre o programa pode-se consultar a página <<http://www.opendx.org>>.

Este visualizador gráfico, que foi desenvolvido pelo IBM Visualization Systems, contém um sistema de ajuda interativo que fornece uma vasta quantidade de informações sobre suas funções e um tutorial que dá acesso ao usuário a uma variedade de exemplos de montagem de visualizações. O OPENDX também permite ao usuário estender sua funcionalidades adicionando seus próprios módulos por meio de bibliotecas, assim como, criar seus próprios módulos (Thompson et al., 2001).

O objetivo deste relatório é utilizar as ferramentas de visualização de dados do OPENDX. Neste trabalho os dados de entrada encontram-se na extensão .general e utiliza-se para a apresentação dos dados o ambiente operacional GNU/LINUX.

Este relatório foi organizado com a intenção de apresentar o ambiente de programação visual do OPENDX e resumir as informações apresentadas em (IBM, sd.b,s,s; Thompson et al., 2001). No Capítulo 2, o leitor é apresentado ao ambiente OPENDX, estruturas e funções, e no Capítulo 3 apresenta-se exemplos de visualizações de dados.

2.1.1 Comandos da barra de ferramentas

File: Apresenta funções que permitem criar um novo programa, salvar, carregar macros, carregar descrição de macros, imprimir e sair do programa.

Edit: Apresenta funções que permitem configurar o programa, encontrar ferramentas, tabelas de entrada/saída, ferramentas de seleção e não seleção, deletar, copiar, apagar, colar, adicionar anotações, inserir um programa, criar macros, nomear macros, manipular uma página e adicionar comentários.

Execute: Por meio deste menu é possível executar o programa, executar somente as alterações do programa, finalizar a execução e

Windows: Apresenta as funções que possibilitam criar um novo painel de controle e abrir um painel de controle selecionado, todos os painéis de controle, o painel de controle pelo nome, macros selecionados, janela de mensagem, um editor de Colo

Connection: Apresenta funções que permitem iniciar um servidor, desconectar de um servidor, reiniciar um servidor e agrupar execuções de tarefas.

Options: Apresenta funções que permitem apresentar uma barra de ferramentas, impedir sobreposições de imagens, acessar o Control Painel, agrupar Control Painel e criar grades.

Help: Contém um sistema de ajuda interativo, que permite ao usuário selecionar qualquer item e obter informações sobre ele, procurar informações sobre funções em ordem alfabética e obter informações sobre o OPENDX.

2.1.2 Área de Programação

Outra facilidade apresentada é a área de programação visual, que é utilizada como área de trabalho para a construção do programa visual, com é mostrado na Figura 3.3. Inicialmente esta área é denominada Untitled, entretanto, o usuário pode renomeá-la e também inserir outras áreas de programação, para facilitar a organização dos programas.

2.2 Macros Disponíveis no Módulo Padrão

No campo **Categories** encontra-se as funções que possibilitam a estruturação dos dados. Neste campo encontra-se a opção **ALL** que lista todas as funções de construção da estrutura de dados em ordem alfabética, sendo que estas funções também estão separadas por

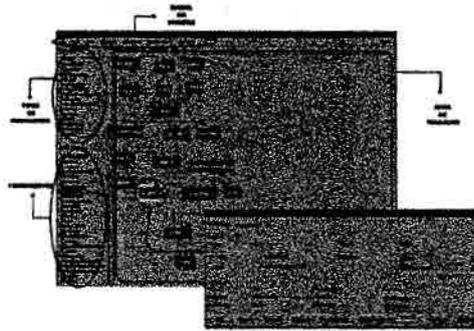


FIGURA 2.2 – Exemplo da janela do Visual Program Editor

categorias. A seguir são explicadas cada categoria com suas respectivas funções.

Annotation: Apresenta vários tipos de informações para uma visualização:

AutoAxes: Cria uma caixa de eixos para um dado específico, usando uma camera específica. A camera é requerida para que o *AutoAxes* possa determinar a posição dos eixos das caixas e dos rótulos.

AutoGlyph: Produz uma figura representacional, tal como uma seta para todos os valores de dados em um arquivo de entrada.

BarChar: Cria uma barra de caracteres.

Caption: Cria um título para uma imagem. A posição do título é especificada em pixel ou coordenadas. As legendas sempre ficam alinhadas com a janela.

ColorBar: Cria uma paleta de cores para ser exibida na janela. Aceita um mapa de cores como entrada.

Format: Cria uma variável para valores numéricos. Por exemplo, você pode criar a string "Isosurface valor = 23.4", em que 23.4 é a saída uma função *Interactor*.

Glyph: Produz uma figura representacional, como uma seta, para todo valor de dado no arquivo de entrada. Este módulo produz um controle mais preciso do tamanho da figura representacional do que o *AutoGlyph*.

Legend: Produz uma legenda que associa uma variável com uma cor.

Parse: Separa uma variável em seus componentes (ex.: a variável "data = 3.7" em "data", "=", e o valor do ponto flutuante "3.7").

- Plot:** Cria uma saída bidimensional para uma linha. Você pode especificar: rótulos para eixos; limites de eixos, eixo y no lado esquerdo e direito da impressão; e eixos lineares e logarítmicos.
- Ribbon:** Produz uma faixa para especificar a largura de uma linha. Se uma componente normal está presente na linha (Exemplo: se a opção `curl` foi usada para criar uma curva ou listra), a espiral da faixa apontará para a direção da normal.
- Text:** Produz um texto que aparece no espaço ocupado pela imagem. Isto é, um título diferente, este módulo especifica o tamanho e a posição do texto em alguma coordenada como o dado.
- Tube:** Produz um tubo para especificar o diâmetro e o número de faixas de uma linha. Se uma componente normal está a presente em uma linha (Exemplo: se a opção `curl` foi usada para criar uma linha ou faixa), a espiral do tubo apontará para a direção da normal.
- DXLink:** Estas ferramentas são usadas para controlar o *Data Explorer* separado de um programa.
- DXLInput:** Habilita uma aplicação remota do *DXLink* para ajuste de valores de parâmetros em um programa de visualização.
- DXLInputNamed:** Habilita uma aplicação remota do *DXLink* para ajustar de valores de parâmetros em um programa de visualização, mas também concede o nome da variável para ser ajustada via um condutor dentro da ferramenta *DXLInputNamed*.
- DxLOutput:** Envia um valor para uma aplicação remota.
- Debugging:** Estas ferramentas facilitam as análises da execução do programa.
- Describe:** Apresenta informações sobre qualquer objeto do *Data Explorer* na janela de mensagens ("*Message windows*").
- Echo:** Repete uma mensagem. Pode também apresentar exemplo de *Array Objects* (ex.: saída dos módulos *Statistics*). Na interface do usuário esta saída aparece na janela de mensagens.
- Message:** Exibe uma mensagem para o usuário, ou na janela de mensagens ou em uma caixa de diálogo.
- Print:** Imprime informações sobre um objeto (ex.: um arquivo ou uma imagem). Você pode especificar o nível de detalhes. Em uma interface de usuário a saída de dados aparece em uma janela de mensagem.

System: Permite executar arbitrariamente comandos dos sistemas.

Trace: Habilita o sinal do tempo gasto em um módulo, o sinal de memória, ou o uso de alertas de erros ou módulos de textos.

Usage: Exibe a quantidade de memória corrente usada pelo *Data Explorer*.

Verify: Verifica um objeto para *consistency* internas

VisualObject: Cria uma classificação de objetos representando uma hierarquia de objetos.

Flow Control: Estas ferramentas controlam o fluxo de execução em um programa visual.

Done: Especifica se uma execução de um loop deveria terminar.

Execute: Permite ao usuário mudar o estado de execução de um programa visual sem usar o menu *Execute* na interface de usuário.

First: Indica se a atual repetição de um loop é a primeira.

ForEachMember: Inicia um loop para qualquer membro de um grupo ou item em uma formação.

ForEachN: Repete inteiramente uma entrada específica de inteiros.

GetGlobal: Recupera um objeto perdido. Mantém uma ordem entre as execuções.

GetLocal: Recupera um objeto perdido.

Route: Encaminha um objeto através de um caminho especificado.

SetGlobal: Localiza um objeto perdido. Mantém uma ordem entre as execuções.

SetLocal: Localiza um objeto perdido.

Switch: Comuta a saída entre uma lista de entradas.

Import and Export: As duas primeiras ferramentas listadas, junto com *ReadImage* e *WriteImage*, estão relacionadas com o fluxo de entrada e saída de dados em um programa visual. As outras processam os dados imediatamente após eles terem sido importados.

Export: Exporta objetos criados no *Data Explorer* para um arquivo externo (no formato de arquivo do *Data Explorer*).

- Import:** Conduz os dados de entrada do *Data Explorer* para um arquivo especificado. Se o arquivo contém mais do que uma variável ou contém múltiplas estruturas de dados, porções dos dados podem ser especificadas para importação. Os formatos suportados são originários dos formatos do *Data Explorer*, CDF, netCDF, HDF, e o formato .general.
- ImportSpreadsheet:** Conduz os dados de entrada do *Data Explorer* para o formato spreadsheet, ou tabula, os arquivos de dados.
- Include:** Inclui (ou exclui) pontos baseados em valores de dados (ex.: remove todos os pontos com valores maiores do que 9.3). Esta função pode também ser usada para remover dados considerados inválidos.
- Partiton:** Subdivide os dados aproveitando-se do paralelismo com o *Data Explorer SMP*. Você pode controlar o nível da subdivisão.
- ReadImage:** Lê uma imagem de um arquivo externo.
- Reduce:** Reduz a resolução de uma entrada de dados, filtrando e selecionando a entrada de baixa resolução.
- Refine:** Aumenta o número de exemplos em um dado. Este módulo interpola dados ou cores em uma nova posição de valores de dados e cores na posição original. Também pode converter conexões para quadrantes ou faces para triângulos, e de cubos para tetraedros.
- Slab:** Toma uma posição inferior de um dado com uma conexão regular (cubo, quadrante e linhas). Especifica-se que eixo toma a posição inferior, em que lugar o *slab* deve iniciar, e quanto denso o *slab* deve ser. O módulo não faz interpolação.
- Slice:** Toma uma posição inferior de um dado com conexões regulares, como o módulo *Slab*. Entretanto, diferente do módulo *Slab*. A função cria uma saída com uma dimensionalidade a menos do que a dimensionalidade da entrada de dados. Por exemplo, importa-se dados como uma grade de quatro dimensões, com a quarta dimensão representando tempo. Usa-se então o módulo *Slice* para criar faixas de três dimensões para os dados de entrada.
- Stack:** Agrupa uma série de arquivos de n-dimensões para um arquivo de (n+1)-dimensões. Este módulo pode também ser usado para aumentar a dimensionalidade de um arquivo.
- Transpose:** Transpõe a posição de um arquivo. Por exemplo, pode trocar os eixos x e y.

WriteImage: Escreve uma imagem para um arquivo externo em um formato específico. A mesma funcionalidade é provida pela opção *SaveImage* do menu *File* da janela *Image*.

Interactor: Estas ferramentas permitem controle de interatividade na entrada de outros módulos em um programa de visualização. Elas podem ser usadas separadas com a interface do usuário.

FileSelector: Apresenta uma caixa de diálogo para seleção de arquivos de padrão *Motif*. A saída é um nome de arquivo.

Integer: Apresenta um espessador, um cursor, um apresentador ou um texto interativo. A saída é um número inteiro.

IntegerList: Apresenta uma lista ou um texto interativo. A saída é uma lista de inteiros.

Reset: Apresenta um botão "toggle". A saída é um valor da primeira execução, e um valor diferente nas próximas execuções até que o "toggle" seja selecionado novamente.

Scalar: Apresenta um espessador, um cursor, um apresentador ou um texto interativo. A saída é um número real.

ScalarList: Apresenta um espessador, um cursor, um apresentador ou um texto interativo. A saída é uma lista de valores escalares.

Selector: Apresenta um menu de opções, uma entrada de botão de rádio, ou um botão "toggle". As saídas são um valor e um caracter, representando uma escolha para uma seleção.

SelectorList: Apresenta uma lista de seleção. A saída é uma lista de valores e uma lista de caracteres, representando uma escolha de uma ou mais de várias.

String: Apresenta um texto interativo. A saída é um texto.

StringList: Apresenta uma lista de textos. A saída é uma lista de caracteres.

Toggle: Apresenta um botão "toggle". A saída é um dos dois valores (entrada ou saída).

Value: Apresenta um texto interativo. A saída é um valor (escalar, vetor, "tensor", ou matriz).

ValueList: Apresenta uma lista ou um texto interativo. A saída é uma lista de valores.

Vector: Apresenta um espessador ou um texto interativo. A saída é um vetor.

VectorLit: Apresenta uma lista ou um texto interativo. A saída é uma lista de vetores.

Interface Control: Estas ferramentas são usadas para controlar as ferramentas do *Data Explorer* dentro de um programa de visualização.

ManageColormapEditor: Permite abrir ou fechar editores de cores dentro de um programa de visualização.

ManageControlPanel: Permite abrir ou fechar painéis de controle dentro de um programa de visualização.

ManageImageWindow: Permite abrir ou fechar janelas com imagens dentro de um programa de visualização.

ManageSequencer: Determina se o painel de controle *Sequence* é exibido ou não.

Realization: Estas ferramentas criam estruturas para tradução e exibição (por exemplo: faixas, conexões triangulares, superfícies e caixas).

AutoGrid: Divide uma variável em faixas baseado nos valores de divisões de dados.

Connect: Cria conexões triangulares para uma variável para posições espalhadas.

Construct: Cria uma variável arbitrária. Você pode especificar a origem, a variação, a soma em cada dimensão e o dado. Pode-se usar a função *Construct* para criar uma variável contendo uma componente de posição com valores de posições dados (ex.: a saída da ferramenta *ProbeList*).

Enumerate: Gera uma lista numérica.

Grid: Produz a entrada de dados em uma grade. Você pode construir retângulos, elipses, linhas, blocos, etc. Especifica-se o tamanho do objeto e o número de pontos na grade.

Isolate: Escolhe os elementos de conexões de uma variável para que ela possa ser individualmente vista.

- Isosurface:** Cria superfícies ou linhas para valores constantes. Para dados volumétricos, cria superfícies; para dados de superfícies cria um contorno. Use o módulo *ClipPlane* para exibir o interior de uma isosuperfície.
- MapToPlane:** Esboça uma variável de 3-dimensão em um plano arbitrário. Especifica-se um ponto e modelo para definir o plano.
- Regrid:** Esboça uma entrada de pontos espalhados em uma grade específica.
- RubberSheet:** Deforma uma variável de superfície (composta de triângulos, quadrantes ou linhas) para um agrupamento de valores de dados para cada ponto. Usa-se o módulo *Normals* ou *FaceNormals* para adicionar sombreado antes da tradução.
- Sample:** Produz uma entrada de pontos em uma variável arbitrária. Por exemplo, você pode produzir a entrada de aproximadamente 11 exemplos em uma isosuperfície.
- ShowBoundary:** Cria uma classificação para um objeto que é o limite de uma variável volumétrica.
- ShowBox:** Cria uma entrada de uma linha "renderable" que representa a caixa limite de uma variável.
- ShowConnections:** Cria uma entrada de classificação para linhas que representa a conexão de elementos em uma variável.
- ShowPositions:** Cria uma entrada de uma classificação para pontos que representa a posição da variável.
- Streakline:** Processa uma linha que traça o caminho de uma partícula através de uma variação de campo vetorial. A entrada de dados no módulo é uma série de campos vetoriais, ou um campo simples que é um membro de uma série de variáveis vetoriais. Controla-se os pontos iniciais de uma linha, e pode opcionalmente providenciar um campo curvo para produzir uma curva em qualquer faixa ou tubo construído para a linha.
- Streamline:** Processa uma linha que traça o caminho de uma partícula por meio de um campo vetorial constante. A entrada de dados para um módulo é um campo vetorial. É possível controlar os pontos iniciais de uma linha curva, e pode opcionalmente providenciar um campo curvo para produzir uma curva em qualquer faixa ou tubo construído para a curva.

Rendering: Estas ferramentas criam uma imagem exibida para uma classificação de objeto ou modifica as características visuais do(s) objeto(s) iniciados. Por exemplo, os módulos *Normals* e *FaceNormals* podem ser usados para adicionar sombra a uma estrutura geométrica criada com *RubberSheet* (categoria *Realization*), enquanto *Light* e *AmbientLight* podem ser usados para mudar a iluminação padrão.

AmbientLight: Produz um ambiente iluminado. Você pode especificar a cor da luminosidade. Ambientes iluminados iluminam a superfície igualmente, indiferente da direção, assim um objeto iluminado só com a luz ambiente não está sombreado. Note que não é necessário especificar uma luminosidade para o ambiente, porque um pequeno agrupamento de luminosidade ambiente está construído no modelo de objeto. Entretanto, uma vez especificada uma iluminação ignora-se a iluminação padrão.

Arrange: Cria uma imagem simples para uma coleção de imagens. É possível especificar como muitas imagens devem ser colocadas em uma fila.

AutoCamera: Este módulo produz um camera apropriada como entrada para os módulos *Render* ou *Display*. Especificando uma direção “look from” muda-se o padrão da camera. A largura da imagem em unidades de modelo (projeção ortográfica) ou o campo de visualização (projeção perspectiva), ou o tamanho da imagem e a proporção do aspecto também podem ser mudados.

Camera: Produz uma camera como entrada para os módulos *Render* ou *Display*. Você pode especificar o ponto “look-to”, o ponto “look-from”, a largura da imagem em unidades correspondentes as dos dados, tamanho da imagem e a proporção do aspecto. A função *Camera* difere da *AutoCamera* na especificação do “look-form” como ponto em vez de direção.

ClipBox: Recorta um objeto usando uma caixa. Você pode especificar a explicitidade da caixa, com dois ângulos definindo uma caixa ou pode ser a caixa limite de um objeto. O módulo coloca somente parte do objeto dentro da caixa. As ferramentas *Render*, *Display* ou *Image* efetivamente fazem o corte.

ClipPlane: Recorta um objeto por um plano. O usuário especifica o plano com dois pontos e uma normal. O lado do plano em que está a pro-

jeção da normal é o lado que será cortado (isto é, não mostrado). As ferramentas *Render*, *Display* ou *Image* efetivamente fazem o corte.

Display: Coloca e/ou exhibe uma imagem na tela. Se a camera não é estipulada, o módulo *Display* aguarda a primeira entrada para ser uma imagem (ex.: a saída do *Render* ou *Arrange*).

FaceNormals: Processa uma normal em uma superfície. Quando você usa o módulo *FaceNormals*, cada face da superfície é sombreada.

Image: Esta ferramenta coloca e exhibe uma imagem na tela. Funciona como a *AutoCamera* e a *Display* combinados. Usando a ferramenta *Image* para colocar uma imagem habilitada em muitas direções interativas que não estão sendo utilizadas quando se usa *AutoCamera* e *Display*. Estas opções são utilizadas na opção *View Control* do menu *Options* na janela *Image*. A ferramenta *Image* pode ser utilizada somente em interfaces gráficas.

Light: Produz uma luminosidade em um ponto distante. Você pode especificar a direção e cor da luminosidade. Note que não é necessário especificar uma luminosidade, porque existe um padrão luminoso construído em um modelo de objeto; entretanto, uma vez especificada a luminosidade ignora-se a luminosidade padrão.

Normals: Processa pontos ou faces normais como sombras em uma superfície. Por exemplo, você pode usar este módulo para produzir sombras em deformações e limites. Entretanto, sombras em uma isosuperfície é agradável você usar a opção *gradient shading* no módulo *isosurface*.

Overlay: Sobre põe duas imagens. O resultado é uma nova imagem que pode ser exibido usando o módulo *Display*. A nova imagem é uma soma pixel a pixel de duas imagens, onde "1-blend" atenua os pixels da imagem base, e "blend" atenua os pixels da imagem sobreposta, (blend é uma valor entre 0 e 1). Você também pode fazer "chromakeying" para especificar "blend" como uma cor RGB ou como um caracter especificando uma cor.

Render: Interpreta um objeto e cria uma imagem. O objeto pode se qualquer combinação de volumes, superfícies, linhas, e pontos e é possível recortar o objeto. É necessário fornecer uma camera para o módulo *Render* para especificar direção visualizada.

Reorient: Rotaciona ou inverte uma imagem ou um grupo de imagem.

Rotate: Rotaciona um objeto em volta de um eixo especificado. As ferramentas *Render*, *Display* ou *Image* efetivamente fazem a rotação.

Scale: Muda a dimensão de um objeto ao longo dos eixos x , y , e z . A escala é efetivamente feita nas ferramentas *Render*, *Display* ou *Image*.

ScaleScreen: Coloca escalas em todo o objeto da tela (tipicamente legendas e barras coloridas) para um agrupamento especificado.

Shade: Permite a você especificar parâmetros de sombra para objetos tal como especularidade.

Transform: Move, rotaciona e redimensiona um objeto. As ferramentas *Render*, *Display* ou *Image* efetivamente fazem a transferência.

Translate: Move um objeto ao longo dos eixos x , y , e z . As ferramentas *Render*, *Display* ou *Image* efetivamente fazem a transferência.

UpdateCamera: Faz alterações específicas para a camera de entrada.

Special: As ferramentas desta categoria podem ser usadas para uma variedade de propósitos.

Colormap: Apresenta uma ferramenta interativa para criação de mapas coloridos.

Input: Define uma entrada para uma macro.

Output: Define uma saída para uma macro.

Pick: Permite ao usuário selecionar objetos em uma imagem usando o mouse.

Probe: Permite ao usuário selecionar um ponto xyz em uma imagem usando o mouse.

ProbeList: Permite ao usuário selecionar múltiplos pontos xyz em uma imagem usando o mouse.

Receiver: Aceita os dados de saída de um transmissor para conexões "sem fio".

Sequencer: Permite a animação de um programa de visualização.

Transmitter: Produz um objeto em conexões "sem fio".

Structuring: Estas ferramentas manipulam as estruturas de dados do *Data Explorer*. Sua função inclui a criação de hierarquias, seleção de elementos em uma hierarquia, permitindo operações em outros componentes de dados, manipulação

de variável ou grupo de componente, e determina que parte do programa visual está sendo executada.

Append: Adiciona objetos como membros de um grupo existente.

Attribute: Extrai um atributo para um objeto.

ChangeGroupMember: Insere, renomeia ou apaga um membro de um grupo existente.

ChangeGroupType: Muda o tipo do grupo.

Collect: Coleta objetos em um grupo. Por exemplo, você pode usar o *collect* para coletar uma curva, um isosuperfície e uma luminosidade. Você pode então passar coleta para a ferramenta *Image*.

CollectMultiGrid: Coleta objetos em um *MultiGrid* (um grupo que será tratado como um objeto de dados simples).

CollectNamed: Coleta objetos em um grupo (como no *Collect*) mas permite que qualquer objeto do grupo receba um nome.

CollectSeries: Coleta objetos em uma série. É possível fornecer para qualquer elemento da série uma posição na série.

CopyContainer: Copia o objeto do bloco mais elevado.

Extract: Extrai um componente de um campo.(ex.:as cores do componente).

Inquire: Retorna informações sobre os objetos de entrada (ex.: tipo de dados, número de elementos).

List: Encadeia vários itens em uma lista simples.

Mark: Marca um componente específico em um campo como um componente de dado. Muitos módulos operam somente no componente de dados. Assim o módulo *Mark* permite aos módulos operarem em outros componentes de dados (ex.: posição ou cor). Se um componente de dado já existe, ele é salvo em um componente de dados. Retaurasse o módulo usando *Unmark*.

Options: Associa atributos com um objeto. Por exemplo, opções de traçado tal como tipo de marcador podem ser especificadas pela adição de atributos em uma linha.

Remove: Remove um componente especificado de um campo.

Rename: Renomeia um componente em um campo. Por exemplo, você pode renomeiar o componente de cor para “cor front” para obter somente cores “front-facing”. Você deverá ter consciência que o nome do componente tem significado especial para os módulos usados.

Replace: Substitui um componente em um campo por um componente de outro campo, ou de uma matriz.

Select:] Seleciona membros de fora de um grupo ou elementos de uma lista. Duas isosuperfícies, uma curva e um plano mapeado. É possível usar, então, a entrada *selector* para o módulo e escolher quais delas é a saída do módulo.

Unmark: Desfaz a ação do módulo *Mark*. Move o componente de dado para depois do componente que você especificar e o componente dado salvo é restaurado para o componente de dado.

Transformation: Estas ferramentas geralmente modificam ou adicionam componentes de campos de entrada sem mudar posições subjacentes e conexões. Por exemplo, cria um componente de cor baseado em um valor de dado de um campo e o *Compute* faz uma operação matemática em um componente de dado de um campo.

Autocolor: Automaticamente colore um campo de dados. Pelo padrão, o mínimo valor de dados é o azul, o máximo valor de dados é o vermelho e valores que estão entre azul e vermelho são ciano, vermelho verde e amarelo. Controla-se qual parâmetro de dados a cor abrange e qual porção da cor deve-se usar. Controla-se também a opacidade do objeto e intensidade da cor. A *Autocolor* também escolhe cores e volumes apropriados para volumes.

AutoGrayScale: Colore automaticamente um campo de dados usando uma escala cinza.

Categorize: Classifica dados, criando um componente de dado inteiro ao longo de uma tabela em que referencia inteiros.

CategoryStatistics: Representa várias estatísticas como mínimo, máximo, número de intens, etc., em dados categóricos.

Color: Permite a você mais controle sobre a coloração de um objeto do que é possível com o *AutoColor*. Especifica-se uma cor representando-a por um carácter (ex.: “spring green”), uma cor RGB ou um mapa de cores (*Colormap Editor*, por exemplo) para se aplicado em um campo

de dado. Pode-se utilizar o módulo cor para fazer objetos translúcidos (ex.: isosuperfícies).

Compute: Representa ponto por ponto uma expressão aritmética em um campo ou em vários campos. Por exemplo, ela pode adicionar um componente de dados em um campo para tangenciar um componente de dados em outro campo. Você também pode usar o módulo *Compute* para selecionar componentes de um campo vetorial; construir um componente vetorial para compôr um campo de entrada escalar; ou representar uma operação condicional.

Compute2: Difere do *Compute* em permitir que uma expressão passe pelo meio de uma tabela de entrada de dados (ex.: de um Seletor interativo).

Convert: Converte cor, saturação, e espaço coloridos e vermelho, verde e espaço azul. Você pode converter também um vetor simple, uma lista de vetores ou um mapa de cores.

DFT: Realiza uma transformação “Fourier” discreta em um campo de 2- ou 3-dimensões.

Direction: Converte azimutais, elevações e distâncias para coordenadas Cartesianas (x, y e z), que são utilizadas para especificar a direção e para especificar a *ClipPlane* ou a direção da normal do *MaptoPlane*.

DivCurl: Processa a divergência e a curva de um campo vetorial.

Equalize: Aplica um histograma em um campo.

FFT: Realiza uma rápida transformação em campo de 2- ou 3-dimensões.

Filter: Filtra um campo. Você pode especificar uma descrição de nome em um filtro (ex.: “laplacian”) ou explicitamente uma matriz de filtro.

Gradient: Processa a inclinação de um campo escalar.

Histogram: Processa um histograma e a média de um campo de entrada de dados. Pode-se então usar a *Plot* para visualizar o resultado como um gráfico de 2-dimensões.

Lookup: Usa um objeto para procurar o valor de outro objeto em um campo.

Map: Mapea campos em um outro. Por exemplo, é possível mapear um campo de dado em uma isosuperfície ou em um tubo que tenha sido formatado por uma linha.

Measure: Realiza medidas em um objeto (ex.: superfície, área ou volume).

Morph: Aplica um operador morfológico binário, tal como desgaste ou dilatação.

Post: Muda a dependência de dados (e outro componentes) entre posições e conexões.

QuantizeImage: Converte uma imagem para um formato que contém muitas cores como pixels para uma imagem quantizada com um mapa de cores com um número especificado de cores ente 8 e 256.

Sort: Classifica uma lista ou campo baseado em valores do componente de dados.

Statistics: Processa a estatística de um campo: o ponto médio, o desvio padrão, a variância, mínimos e máximos. Estas estatísticas são entradas para outros módulos, ou impressão usando os módulos *Echo* ou *Print*.

Windows: Estas ferramentas criam ou supervisionam janelas de imagens.

ReadImageWindow: Lê uma imagem (pixels) posterior de uma janela *Image* ou *Display*, para as imagens interpretadas por software e hardware.

SuperviseState: Observa a ocorrência de ações no mouse e teclado em uma janela *Display*, baseado em uma solicitação definida.

SuperviseWindow: Cria uma janela que será monitorada pelo "SuperviseState".

CAPÍTULO 3

Exemplos de Visualizações

3.1 Montando uma Visualização

Na janela *Visual Program Editor* selecione a opção *FileSelector*, que se encontra no menu *Categories* na opção *Interactor*. Após a função selecionada clique na área *Untitled*, na qual irá aparecer a caixa da função. Esta caixa permite selecionar o caminho do dados que são visualizados por meio da janela

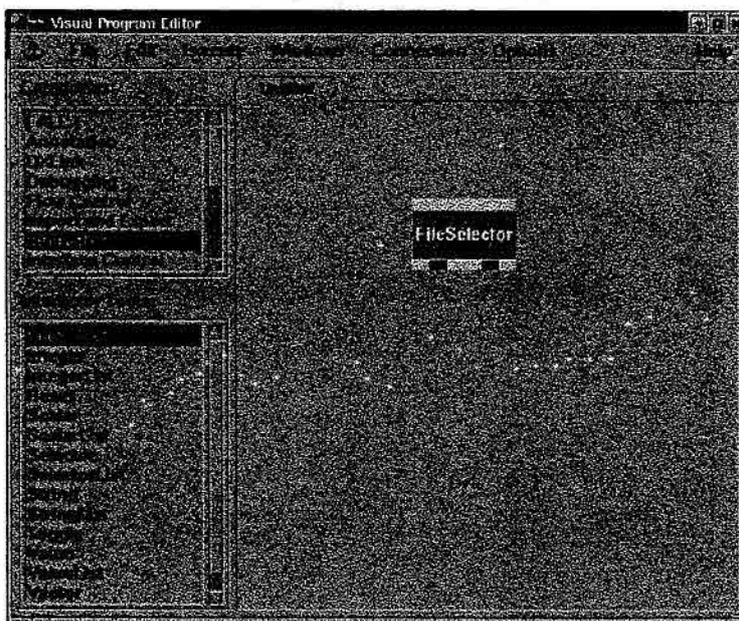


FIGURA 3.1 – Exemplo da janela do *Visual Program Editor*

Os próximos itens são selecionados de forma similar ao *FileSelector*, mudando apenas a categoria em que se encontram: *Import*, categoria *Import and Export*. Todas as funções utilizadas foram explicadas no capítulo deste relatório.

REFERÊNCIAS BIBLIOGRÁFICAS

Blaz, R.; Domingues, M. O.; Mendes, O. J. **Introdução à entrada de dados no OpenDX: Formatos , .general e .grb.** Instituto Nacional de Pesquisas Espaciais, 2003. 11

IBM. **Visualization data explorer. programmer 's reference.** SC38-0486-03. IBM, Estados Unidos, sd.a. <<http://www.opendx.org>>. 9

——. **Visualization data explorer. quickstart guide.** SC34-3262-02. IBM, Estados Unidos, sd.b. <<http://www.opendx.org>>. 9

——. **Visualization data explorer. user's reference.** SC38-0486-03. IBM, Estados Unidos, sd.c. <<http://www.opendx.org>>. 9

Thompson, D.; Braun, J.; Ford, R. **Opendx path to visualization.** Visualization and Imagery Solution, 2001. <<http://www.vizsolutions.com>>. 9