



MINISTÉRIO DA CIÊNCIA E TECNOLOGIA

INSTITUTO NACIONAL DE PESQUISAS ESPACIAIS

sid.inpe.br/mtc-m21d/2024/07.23.11.04-NTC

TUTORIAL DO PARAVIEW PARA SIMULAÇÕES MAGNETO-HIDRODINÂMICAS

Leonardo Sattler Cassará

Margarete Oliveira Domingues

Odim Mendes Júnior

Varlei Everton Menconi

Nota Técnico-Científica

URL do documento original:

<<http://urlib.net/8JMKD3MGP3W34T/4BMP9KH>>

INPE

São José dos Campos

2024

PUBLICADO POR:

Instituto Nacional de Pesquisas Espaciais - INPE
Coordenação de Ensino, Pesquisa e Extensão (COEPE)
Divisão de Biblioteca (DIBIB)
CEP 12.227-010
São José dos Campos - SP - Brasil
Tel.:(012) 3208-6923/7348
E-mail: pubtc@inpe.br

CONSELHO DE EDITORAÇÃO E PRESERVAÇÃO DA PRODUÇÃO INTELLECTUAL DO INPE - CEPPII (PORTARIA Nº 176/2018/SEI-INPE):**Presidente:**

Dra. Marley Cavalcante de Lima Moscati - Coordenação-Geral de Ciências da Terra (CGCT)

Membros:

Dra. Ieda Del Arco Sanches - Conselho de Pós-Graduação (CPG)
Dr. Evandro Marconi Rocco - Coordenação-Geral de Engenharia, Tecnologia e Ciência Espaciais (CGCE)
Dr. Rafael Duarte Coelho dos Santos - Coordenação-Geral de Infraestrutura e Pesquisas Aplicadas (CGIP)
Simone Angélica Del Ducca Barbedo - Divisão de Biblioteca (DIBIB)

BIBLIOTECA DIGITAL:

Dr. Gerald Jean Francis Banon
Clayton Martins Pereira - Divisão de Biblioteca (DIBIB)

REVISÃO E NORMALIZAÇÃO DOCUMENTÁRIA:

Simone Angélica Del Ducca Barbedo - Divisão de Biblioteca (DIBIB)
André Luis Dias Fernandes - Divisão de Biblioteca (DIBIB)

EDITORAÇÃO ELETRÔNICA:

Ivone Martins - Divisão de Biblioteca (DIBIB)
André Luis Dias Fernandes - Divisão de Biblioteca (DIBIB)



MINISTÉRIO DA CIÊNCIA E TECNOLOGIA

INSTITUTO NACIONAL DE PESQUISAS ESPACIAIS

sid.inpe.br/mtc-m21d/2024/07.23.11.04-NTC

TUTORIAL DO PARAVIEW PARA SIMULAÇÕES MAGNETO-HIDRODINÂMICAS

Leonardo Sattler Cassará

Margarete Oliveira Domingues

Odim Mendes Júnior

Varlei Everton Menconi

Nota Técnico-Científica

URL do documento original:

<<http://urlib.net/8JMKD3MGP3W34T/4BMP9KH>>

INPE

São José dos Campos

2024



Esta obra foi licenciada sob uma [Licença Creative Commons Atribuição-NãoComercial 3.0 Não Adaptada](#).

This work is licensed under a [Creative Commons Attribution-NonCommercial 3.0 Unported License](#).

RESUMO

Este trabalho é um tutorial do **ParaView**, um ambiente para análise e visualização de simulações computacionais. São apresentados os primeiros passos dos usuários, fornecendo informações de instalação e uma introdução à interface gráfica do usuário. Em seguida, são fornecidas instruções para processamento de dados com dicas e exemplos práticos. As ilustrações da aplicação são obtidas usando dados `.vtk` gerados com o framework AMROC (Adaptive Mesh Refinement in Object-Oriented C++), que é particularmente útil nas investigações de magnetohidrodinâmica espacial de uma equipe de pesquisa do INPE. Também são explorados scripts de macro em Python, relevantes para a execução automática de **ParaView** diretamente da linha de comando. A estrutura dos arquivos VTK, formato comumente utilizado para representação de dados científicos, é abordada, e também são discutidas boas práticas em relação às definições de cor e opacidade para renderização desses dados.

Palavras-chave: Visualização científica. Simulação numérica. HPC. Magnetohidrodinâmica. Ciências Espaciais.

TUTORIAL ON PARAVIEW FOR MAGNETOHYDRODYNAMICS SIMULATIONS

ABSTRACT

This work is a tutorial of **ParaView**, a framework for analysis and visualization of computational simulations. The users' first steps are presented, providing installation information and an introduction to the graphical user interface. Then, instructions for processing data are provided with tips and practical examples. The application illustrations are obtained by using `.vtk` data generated with the AMROC framework (Adaptive Mesh Refinement in Object-oriented C++), which is particularly useful in the space magnetohydrodynamics investigations of an INPE research team. Macro scripts in Python are also explored, relevant for automatic execution of **ParaView** directly from the command line. The structure of VTK files, a commonly used format for representing scientific data, is addressed, and good practices regarding colour and opacity definitions for rendering these data are also discussed.

Keywords: Scientific visualization. Numerical simulation. HPC. Magnetohydrodynamics. Space Science.

LISTA DE FIGURAS

	<u>Pág.</u>
3.1 Interface de usuário do ParaView	7
3.2 Entrada de dados .vtk no ParaView	8
3.3 Visualização básica de uma simulação.	9
4.1 Construindo vetor com o filtro <i>Calculator</i>	11
4.2 Utilização do filtro <i>Stream Tracer</i> com <i>Line</i>	13
4.3 Utilização do filtro <i>Stream Tracer</i> com <i>Point Cloud</i>	15
4.4 Visualizando vetores como setas com o filtro <i>Glyph</i>	16
4.5 Exemplo de utilização do filtro <i>Clip</i>	21
4.6 Exemplo de utilização do filtro <i>Slice</i>	22
4.7 Gerando gráfico com o filtro <i>Plot Over Line</i>	23
4.8 Visualizando vetores 2D com o filtro <i>Evenly Spaced Streamlines 2D</i>	24
5.1 Salvando dados numéricos	25
5.2 Salvando imagem.	26
5.3 Salvando um vídeo.	26
5.4 janelas de uma macro	28
6.1 Janela de edição das funções de transferência do ParaView	34
6.2 Visualização do campo de divergência magnética $\nabla \cdot \mathbf{B}$	35
6.3 Editando fonte e cor dos eixos do domínio e a cor de fundo da imagem.	37
6.4 Editando as propriedades de texto da barra de cor.	38
6.5 Editando as funções de transferência de opacidade e cor.	39
7.1 Estrutura básica de um arquivo <i>legacy .vtk</i>	41

LISTA DE ABREVIATURAS E SIGLAS

ASCII	–	American Standard Code for Information Interchange
ALCF	–	Argonne Leadership Computing Facility < alcf.anl.gov >
DKRZ	–	German Climate Computing Center < dkrz.de >
HDF	–	Hierarchical Data Format
IDRIS	–	Institut du développement et des ressources en informatique scientifique < www.idris.fr >
OSTI	–	U.S. Department of Energy Office of Scientific and Technical Information < www.osti.gov >
SANDIA	–	Sandia National Laboratories < www.sandia.gov/ >
SOU	–	University of Southampton < cmg.soton.ac.uk >
TASCC	–	Texas Advanced Super Computing Center < www.tacc.utexas.edu >
VTK	–	Virtualization Toolkit
XML	–	eXtensible Markup Language

SUMÁRIO

	<u>Pág.</u>
1 INTRODUÇÃO	2
2 INSTALAÇÃO DA ÚLTIMA VERSÃO DISPONÍVEL	4
3 PASSOS INICIAIS DO USUÁRIO	6
3.1 Interface de usuário	6
3.2 Entrada de dados	7
3.3 Visualização básica	8
4 VISUALIZAÇÃO DE CAMPOS VETORIAIS	10
4.1 Construindo vetores: filtro <i>Calculator</i>	10
4.2 Linhas de campo: filtro <i>Stream Tracer</i>	12
4.2.1 Usando <i>Line</i>	12
4.2.2 Usando <i>Point Cloud</i>	13
4.3 Vetores como setas: filtro <i>Glyph</i>	14
4.4 Outros filtros	17
4.4.1 Filtro <i>Clip</i>	17
4.4.2 Filtro <i>Slice</i>	17
4.4.3 Filtro <i>Plot Over Line</i>	18
4.4.4 Filtro <i>Evenly Spaced Streamlines 2D</i>	18
5 SALVANDO E AUTOMATIZANDO	25
5.1 Dados numéricos: <i>Save Data</i>	25
5.2 Imagens: <i>Save Screenshot</i>	26
5.3 Vídeos: <i>Save Animation</i>	26
5.4 Estado do programa: <i>Save State</i>	27
5.5 Macros e scripts Python	27
5.5.1 Gerando macros: <i>Start/Stop Trace</i>	27
5.5.2 Rodando scripts Python	28
6 MAPEAMENTO DE CORES	30
7 FORMATO DE DADOS .VTK	40

REFERÊNCIAS BIBLIOGRÁFICAS	43
APÊNDICE A - LINKS ÚTEIS	45
APÊNDICE B - ANACONDA (PYTHON)	46

1 INTRODUÇÃO

Um significativo desafio para as simulações científicas de fenômenos naturais é dispor de recursos para as adequadas visualizações, sobretudo na análises de sistemas dinâmicos e multidimensionais que abrangem diversas escalas temporais e espaciais, e que geram como resultado grandes bases de dados estruturados.

Atendendo essas necessidades, a plataforma **ParaView**¹ constitui-se como um ambiente computacional robusto para análises e visualizações de dados multidimensionais (AHRENS et al., 2005). Multiplataforma e de código aberto, essa ferramenta teve seu desenvolvimento iniciado em 2000, e hoje é amplamente utilizada na área de modelagem numérica em diversos centros mundiais de computação científica de alto desempenho. Neste contexto, auxiliando a sua disseminação e facilitação de uso, esta nota técnico-científica contribui tanto para a formação de recursos humanos como no treinamento para desenvolvimento das atividades do Laboratório Numérico-Computacional Experimental de Plasmas Espaciais (LANCE). As pesquisas desenvolvidas contribuem com o atendimento da missão do Instituto Nacional de Pesquisas Espaciais, enquadrando-se especificamente essas nos projetos de *Pesquisa multiescala adaptativa numérico-computacional para aplicações no estudo de plasmas espaciais*, processo SEI 01340.003199/2021-72, da COPDT/CGPI, e de *Desenvolvimento de pesquisa teórica com modelagem magneto-hidrodinâmica para plasmas espaciais*, processo SEI 01340.003098/2021-00, da DIHPA/CGCE.

Para simulações magneto-hidrodinâmicas (MHD), o LANCE faz uso principalmente do framework AMROC, Adaptive Mesh Refinement in Object-oriented C++ (DEITERDING, 2003; DEITERDING, 2011). Nele, desenvolve-se um módulo MHD multidimensional em parceria com a Universidade de Southampton, conforme descrito nos projetos Domingues (2019), Domingues e Deiterding (2016), Domingues (2016) e Domingues e Deiterding (2016). O ambiente de computação de alto desempenho está atendido pelo recurso HPC-CEA, projetado e implantado especialmente para modelagens de plasmas espaciais (<<http://urlib.net/8JMKD3MGP3W34R/3UP4BT8>>), localizado nas instalações do programa de Estudos e Monitoramento Brasileiro do Clima Espacial (Embrace), situado no piso superior do prédio CEA-II. Como parte do conjunto de atividades científicas, o grupo de pesquisadores e outros recursos humanos envolvidos produzem simulações computacionais que exigem análises e visualizações específicas para entendimento dos fenômenos espaciais de interesse (para exemplo, consultar os trabalhos desenvolvidos por Lopes et al. (2018)

¹<<https://www.paraview.org/about/>>

e [Lopes \(2019\)](#)), em situação que demanda os melhores recursos de computação científica. Desta forma, caracterizando o seu objetivo, esta nota apresenta um tutorial prático e sucinto com o propósito de introduzir o **ParaView** para pós-processamento de tais simulações, propiciando a usuários interessados uma visão com foco mais prático. No entanto, aproveitando, indo além de tal propósito específico, este texto contém informações relevantes para a programação de análises e visualizações úteis a aplicações gerais de física computacional.

O documento está organizado da forma descrita a seguir. O Capítulo 2 apresenta a instalação do **ParaView**. O Capítulo 3 introduz a interface gráfica e ensina como fazer uma leitura e visualização básica de dados de uma simulação. O Capítulo 4 descreve o passo a passo de como modificar e visualizar esses dados para análise física. O Capítulo 5 apresenta algumas opções de salvamento e automatização do **ParaView**. Oportunamente, o Capítulo 6 discute boas práticas na escolha da paleta de cores e detalha o funcionamento das funções de transferência de opacidade e cor. Como uma última parte, para auxiliar o caráter prático do texto, o Capítulo 7 descreve brevemente o formato de arquivo para visualização científica utilizado, o *.vtk* (comumente lido *ponto vtk*). Complementarmente, o Apêndice A oferece links úteis pertinentes ao uso do **ParaView**, e o Apêndice B orienta na instalação do ambiente de programação Python, relevante para a utilização do **ParaView**.

2 INSTALAÇÃO DA ÚLTIMA VERSÃO DISPONÍVEL

No ambiente computacional GNU Linux Ubuntu¹, o ParaView pode ser facilmente instalado com o comando `apt`:

```
sudo apt install paraview
```

Contudo a versão instalada via `apt` comumente está defasada com relação à última versão estável disponível no website oficial do ParaView (<<https://www.paraview.org/download/>>). Para instalar o ParaView obtido diretamente do website, siga as instruções abaixo. Caso já possua uma versão antiga instalada, remova-a primeiro:

```
sudo apt remove paraview
```

O diretório `/opt` é um local adequado para a instalação do ParaView. Inicie fazendo o Download do arquivo `tar.gz` da última versão estável. Após o download, abra um terminal no local do arquivo baixado (normalmente no diretório Downloads). Para criar uma pasta em `/opt` e instalar o ParaView, escreva o seguinte comando:

```
sudo mkdir /opt/paraview
```

Descompacte o arquivo `tar.gz` dentro de `/opt/paraview` (operação que pode demorar um pouco):

```
sudo tar xzf ParaView*.tar.gz -C /opt/paraview --strip-components=1
```

Agora adicione o ParaView no `$PATH`. Primeiro abra seu arquivo `~/.bashrc`

```
sudo gedit ~/.bashrc
```

Em seguida cole a seguinte linha no fim do arquivo

```
export PATH=$PATH:/opt/paraview/bin/
```

Salve e feche o `~/.bashrc`. Está feito. Para usar o ParaView, basta abrir um novo terminal e proceder com o seguinte comando:

```
paraview &
```

Adicionar o `&` permite execução em modo background, i.e. em modo oculto e com o terminal livre. Em alguns casos, a instalação não adiciona um ícone do ParaView no painel de programas. Se esse for o caso, crie um arquivo `paraview.desktop`:

```
sudo gedit /usr/share/applications/paraview.desktop
```

¹<<https://ubuntu.com/>>

Cole as seguintes linhas, salve e feche.

```
[Desktop Entry]
Version=1.0
Name=ParaView

Exec=paraview
Terminal=false
Icon=/opt/paraview/share/icons/hicolor/96x96/apps/paraview.png
Type=Application

[NewWindow Shortcut Group]
Name=New Window
Exec=paraview
TargetEnvironment=Unity
```

3 PASSOS INICIAIS DO USUÁRIO

Para efeitos de aprendizagem, é importante nomear as diferentes partes do programa exploradas nos próximos capítulos. A seção a seguir introduz o **ParaView** desta forma e, nas seguintes, estabelecem-se as ações básicas para visualizar saídas do **AMROC**.

3.1 Interface de usuário

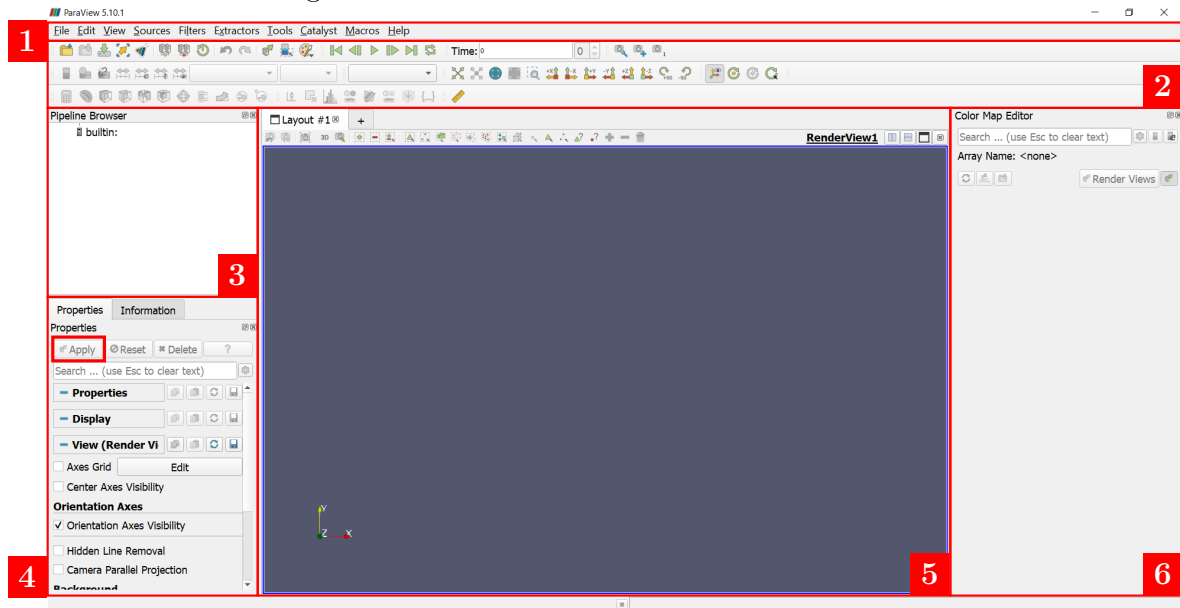
Ao abrir o **ParaView**, o usuário vê uma tela como a da Figura 3.1, em que as principais partes do programa estão enumeradas. São elas:

- 1 Barra de menu**, com as opções mais gerais, como *File*, *Edit*, etc. Em particular, a opção *Filters* acessa todos os algoritmos de transformação de dados, o cerne do **ParaView**
- 2 Barra de ferramentas**, com diversos botões de edição e controle. As setas verdes do lado esquerdo de *Time* mudam o tempo da simulação, e os diferentes botões em formato de setas e eixos abaixo destes alteram o posicionamento da visualização
- 3 Pipeline browser**, que abriga os arquivos importados e os resultados das edições em sequência de cima para baixo
- 4 Painel de propriedades**, local das principais ações de edição.

DICA: O BOTÃO *Apply*, DESTACADO NA FIGURA 3.3, PRECISA SER CLICADO PARA APLICAR AS AÇÕES DOS PROCESSAMENTOS. ALTERNATIVAMENTE, PODE-SE DEFINIR QUE TODAS AS AÇÕES SEJAM AUTOMATICAMENTE ATIVADAS. ESSA OPÇÃO AUTOMÁTICA PODE TORNAR OS PROCESSAMENTOS E AS VISUALIZAÇÕES LENTAS E, NO CASO DE GRANDES BASES DE DADOS, RECOMENDA-SE CLICAR EM *Apply* SOMENTE AO FINAL DOS PROCESSAMENTOS

- 5 Layout**, que contém a visualização. Os botões no topo à esquerda servem para posicionar a câmera e selecionar dados, e os do topo à direita servem para fechar e criar novas janelas de visualização (dividindo na horizontal ou vertical), que podem servir ao mesmo arquivo ou outra importação
- 6 Janela de ferramentas**, que serve como uma extensão da **barra de ferramentas**. Seu aparecimento é controlado na opção *View* da **barra de menu** ou por meio das opções na **barra de ferramentas** que a ativam. Um exemplo importante é opção da barra de cores, vista mais adiante.

Figura 3.1 - Interface de usuário do ParaView.



3.2 Entrada de dados

Arquivos podem ser importados no ParaView por, pelo menos, três maneiras:

- na **barra de menu**, clique em *File* e depois em *Open*, **ou**
- na **barra de ferramentas**, clique no primeiro item com formato de pasta no topo à esquerda, **ou**
- no **Pipeline browser**, clique com o botão direito do mouse e depois em *Open*

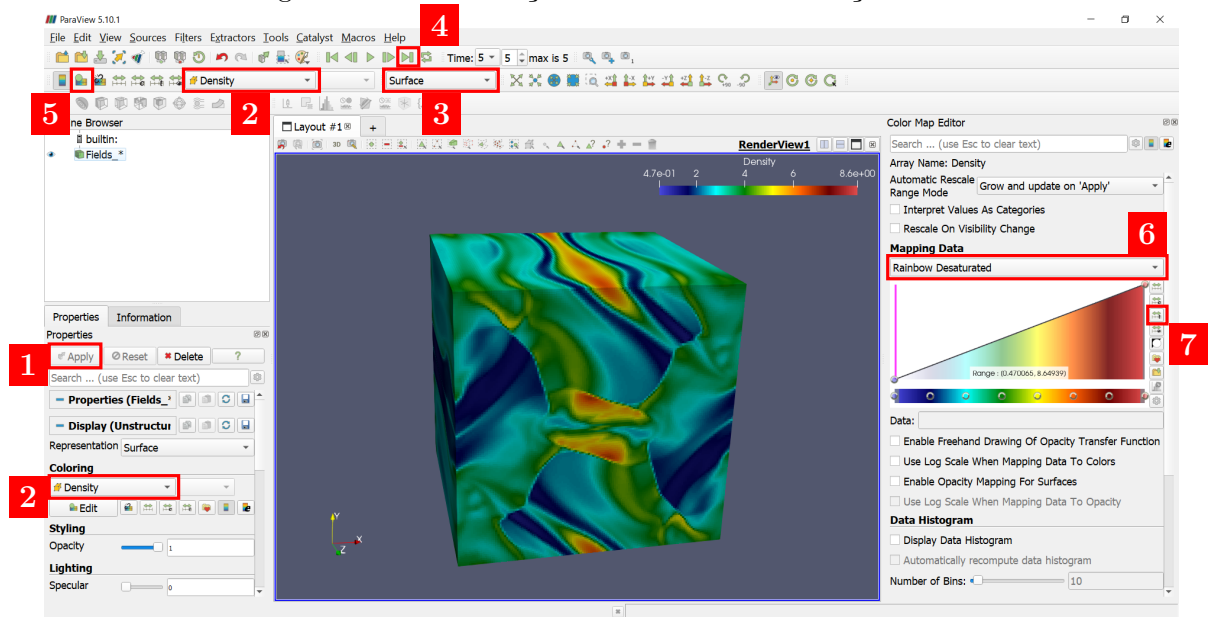
Em seguida, uma janela como a da Figura 3.2 deve surgir. No AMROC, os arquivos de saída possuem formato HDF (Hierarchical Data Format), que são convertidos em VTK (Virtualization Toolkit) para visualização no ParaView. Em geral, as saídas de uma simulação aparecem com “...” no nome e podem ser importadas em lote. Alternativamente, a série de arquivos pode ser expandida para importação apenas de um tempo específico. Essas séries no AMROC são usualmente denominadas Field...vtk, e na importação da Figura 3.2 a saída final é o arquivo Fields_595.vtk. Clique *Ok* selecionando a série para importar toda a simulação.

DICA: SALVE LOCAIS FAVORITOS SELECIONANDO UM DIRETÓRIO E CLICANDO EM “+”.

Após a importação dos dados conforme a seção anterior, uma visualização básica segue os passos indicados na Figura 3.3:

- 8

Figura 3.3 - Visualização básica de uma simulação.



DICA: DENTRO DO **layout**, UTILIZE O BOTÃO DE ROLAGEM DO MOUSE PARA DAR ZOOM IN E OUT DA FIGURA; SE A VISUALIZAÇÃO É 3D, CONFORME INDICADO NO QUARTO BOTÃO DA ESQUERDA PARA DIREITA, CLIQUE E ARRASTE PARA GIRAR O DOMÍNIO EM TORNO DO SEU CENTRO; NO CASO 2D, O DOMÍNIO FIXA SUA PERSPECTIVA E NÃO PODE SER GIRADO, SOMENTE ARRASTADO. A BARRA DE COR TAMBÉM PODE SER ARRASTADA.

4 VISUALIZAÇÃO DE CAMPOS VETORIAIS

No capítulo anterior, o exemplo de visualização utilizou a variável densidade, que é um campo escalar. Além da densidade, outras grandezas escalares que o AMROC-MHD oferece como saída incluem as componentes do campo de velocidade e as componentes do campo magnético. Essas componentes geram campos vetoriais, cujas visualizações são tratadas nas seções a seguir.

4.1 Construindo vetores: filtro *Calculator*

Como o domínio deste exemplo é tridimensional, a simulação fornece três componentes como as saídas de cada vetor, que, no caso do campo magnético, são "Y_01_-_Bx", "Y_02_-_By", "Y_03_-_Bz", que precisam ser multiplicadas pelos seus respectivos versores *iHat*, *jHat*, *kHat* e então somadas para obter o vetor. As Figuras 4.1(a)–(b) ilustram os passos para a construção do campo magnético com o filtro *Calculator*, acessado por meio da opção *Filters* na **barra de menu**:

- 1 clique no filtro *Calculator* via *Filters>Common* na **barra de menu**
- 2 utilize a calculadora no **painel de propriedades** para escrever a equação

$$\text{"Y_01__Bx"} * iHat + \text{"Y_02__By"} * jHat + \text{"Y_03__Bz"} * kHat$$

- 3 renomeie o resultado para B tanto no **painel de propriedades** quanto no **pipeline browser** (opcional; mas importante para organização)

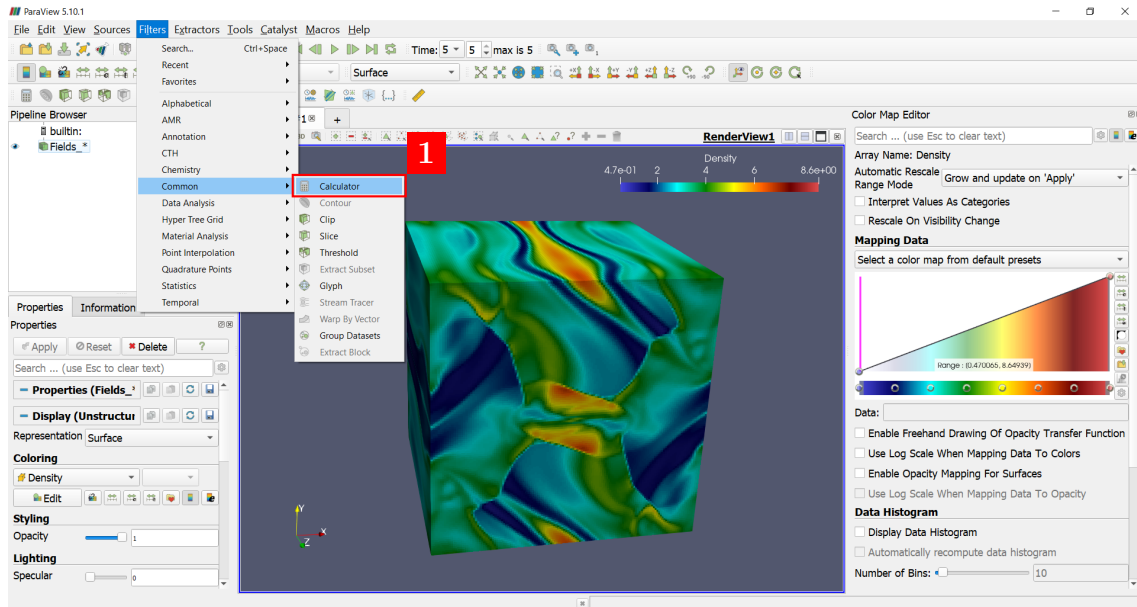
Após clicar em *Apply*, o resultado de B permanece no **pipeline browser** abaixo de *Fields_**, e sua visualização surge no **layout** mostrando a variável previamente selecionada, densidade. Para visualizar a magnitude do campo magnético, outra grandeza escalar agora disponível, use o botão 2 da Figura 3.3 e escolha a variável B. Alterne entre cada visualização ativando/desativando o ícone de olho na frente do nome dos dados no **pipeline browser** e alterne entre a seleção de cada com um clique sobre seu nome.

DICA: VISUALIZAR E SELECIONAR O DADO SÃO AÇÕES DIFERENTES. O PRIMEIRO ESCOLHE QUEM APARECE NO **layout**; ENQUANTO O SEGUNDO ESCOLHE O DADO A SER MODIFICADO PELAS SUAS AÇÕES NO **PARAVIEW**.

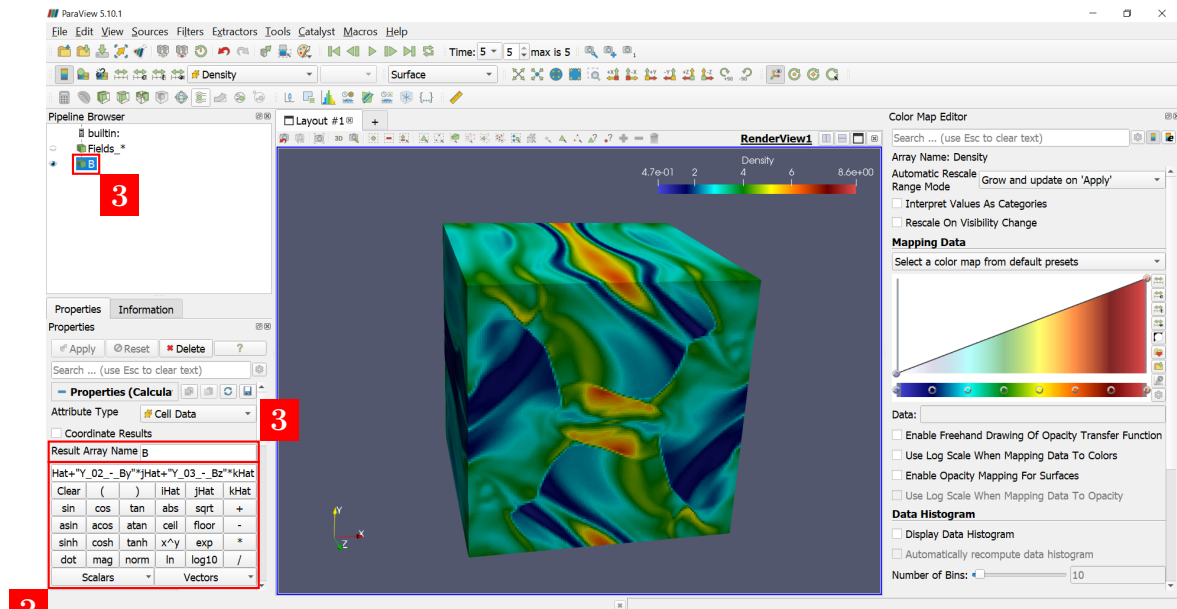
DICA: OUTROS ESCALARES TAMBÉM PODEM SER CRIADOS USANDO O FILTRO *Calculator*;
A ENERGIA MAGNÉTICA, POR EXEMPLO, USA A SEGUINTE EQUAÇÃO:

$$0.5 * \text{sqrt}("Y_01__Bx"^2 + "Y_02__By"^2 + "Y_03__Bz"^2).$$

Figura 4.1 - Construindo vetor com o filtro *Calculator*.



(a)



(b)

4.2 Linhas de campo: filtro *Stream Tracer*

Construído o vetor, seu campo vetorial pode ser visualizado com o filtro *Stream Tracer*. Ele traça a visualização dos vetores através de uma fonte, que pode ser uma linha (*Line*) ou um ponto (*Point Cloud*). Se uma linha, o filtro calcula os vetores que a cruzam; se um ponto, ele determina uma esfera centrada num ponto e calcula os vetores que passam por ela.

4.2.1 Usando *Line*

A fonte padrão do *Stream Tracer* é uma linha, que é a fonte mais apropriada para os dados do AMROC. Com destaque, as Figuras 4.2(a)–(c) mostram as etapas para a visualização do campo magnético com este filtro usando a linha como fonte:

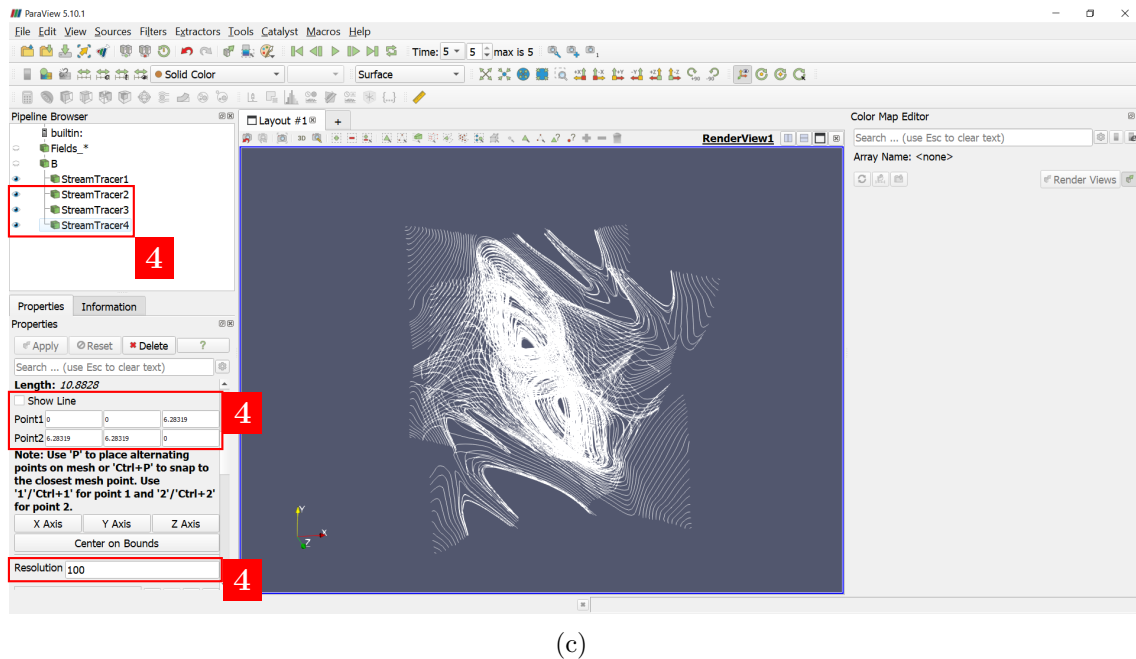
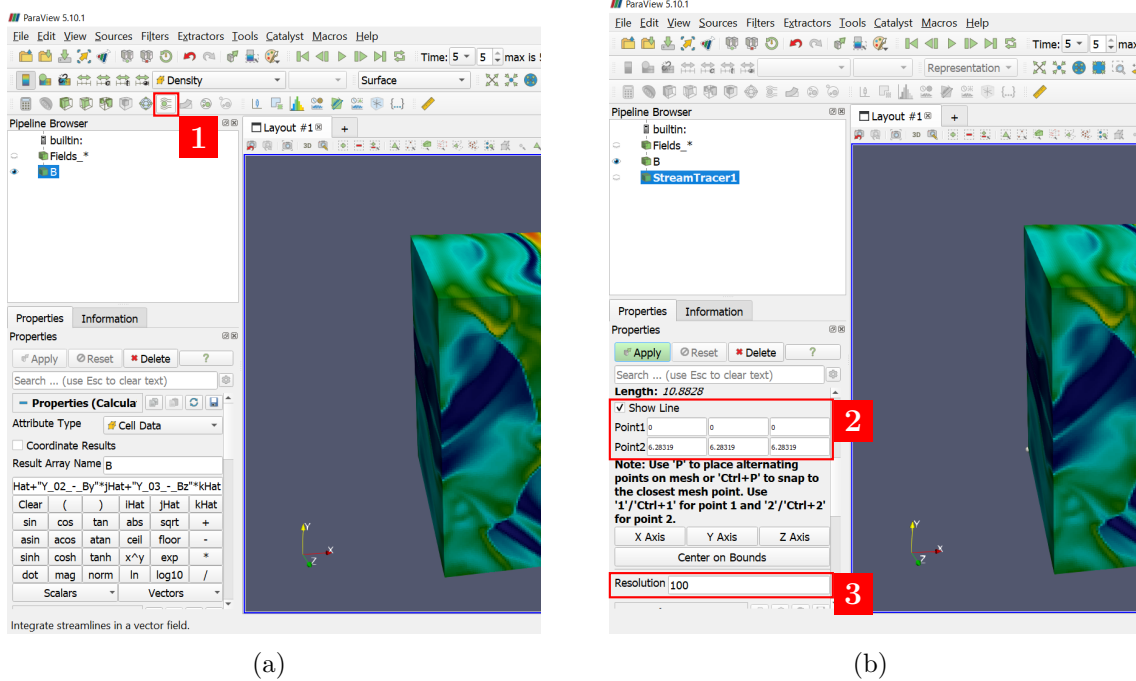
- 1 selecione o filtro *Stream Tracer* pela **barra de menu** indo em *Filters>Alphabetical*, ou clique no botão da Figura 4.2(a) na **barra de ferramentas**
- 2 neste quadro, *Point1* se refere à origem e *Point2* à coordenada final da linha-fonte, respectivamente $(0,0,0)$ e $(2\pi, 2\pi, 2\pi)$. Ela passa por uma diagonal interna, e essa posição deve ser mantida. Marque/desmarque a caixa *Show Line* para exibir/esconder a linha-fonte
- 3 mude o número máximo padrão de linhas de campo a serem visualizadas em *Resolution* de 1000 para 100, caso contrário a renderização do campo levará muito tempo; o botão *Apply* já pode ser clicado para visualizar o resultado, mas é recomendado seguir e aplicar tudo de uma vez ao final
- 4 selecione B novamente no **pipeline browser**; repita todos os passos anteriores posicionando a linha-fonte nas três diagonais internas restantes, usando para isso as seguintes coordenadas:

StreamTracer2: *Point1* = $(2\pi, 0, 0)$, *Point2* = $(0, 2\pi, 2\pi)$

StreamTracer3: *Point1* = $(0, 2\pi, 0)$, *Point2* = $(2\pi, 0, 2\pi)$

StreamTracer4: *Point1* = $(0, 2\pi, 2\pi)$, *Point2* = $(2\pi, 2\pi, 0)$

Figura 4.2 - Utilização do filtro *Stream Tracer* com *Line*.



4.2.2 Usando *Point Cloud*

Como os dados do AMROC são baseados em volumes-finitos, o uso de *Point Cloud* precisa da conversão dos dados de células para pontos no espaço. Isso é feito com o filtro *Cell Data to Point Data*, aqui as Figuras 4.3(a)–(b) ilustram todas as etapas:

- 1 tire a visualização dos filtros anteriores e selecione o dado B novamente; utilizando o atalho para filtros Ctrl + barra de espaço, escreva *Cell Data to Point Data* e tecle enter; finalize clicando em *Apply*
- 2 aplique o filtro *Stream Tracer* e mude a fonte para *Point Cloud*
- 3 mantenha o centro da esfera posicionado no meio do domínio com *Center* igual a (π, π, π) , e mude o valor do raio em *Radius* para π ; a caixa *Show Sphere* exhibe/esconde a esfera-fonte
- 4 aumente de 100 para 200 em *Number of Points* para visualizar um número razoável de linhas e aplique

4.3 Vetores como setas: filtro *Glyph*

Por vezes a visualização das linhas do campo não é suficiente, sendo necessário também o sentido dos vetores. O filtro *Glyph* trata os vetores como setas, aqui as Figuras 4.4(a)–(c) exemplificam sua utilização com o campo de velocidade:

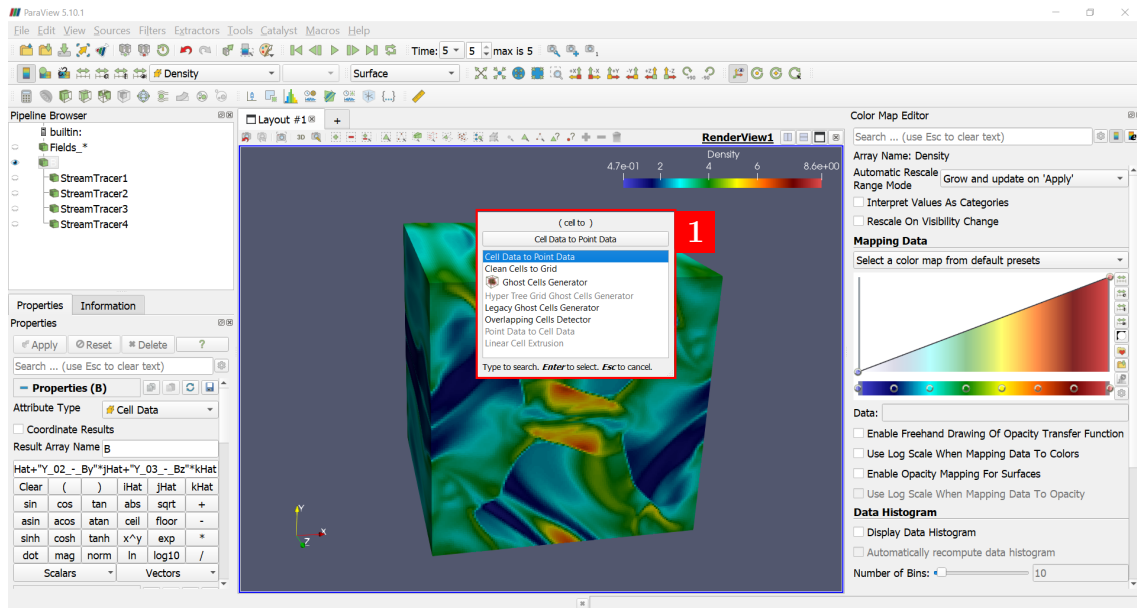
- 1 selecionado o dado *Fields_**, utilize o filtro *Calculator* para escrever a equação

$$\text{Velocity_u} * \mathbf{iHat} + \text{Velocity_v} * \mathbf{jHat} + \text{Velocity_w} * \mathbf{kHat},$$

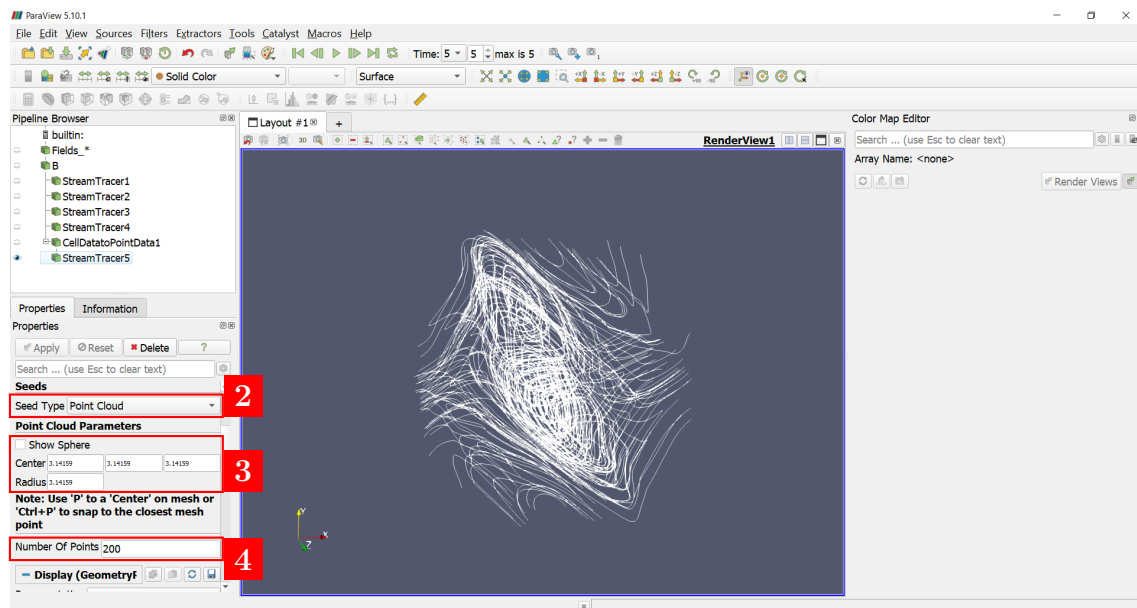
renomeie o resultado para V e aplique

- 2 selecione o filtro *Glyph* diretamente na **barra de ferramentas**
- 3 no **painel de propriedades**, escale o tamanho dos vetores pela magnitude do vetor V criado
- 4 mude o *Maximum Number of Samples* para 2000 e *Seed* para 5000 de modo a visualizar um número razoável de vetores e, em seguida, aplique

Figura 4.3 - Utilização do filtro *Stream Tracer* com *Point Cloud*.



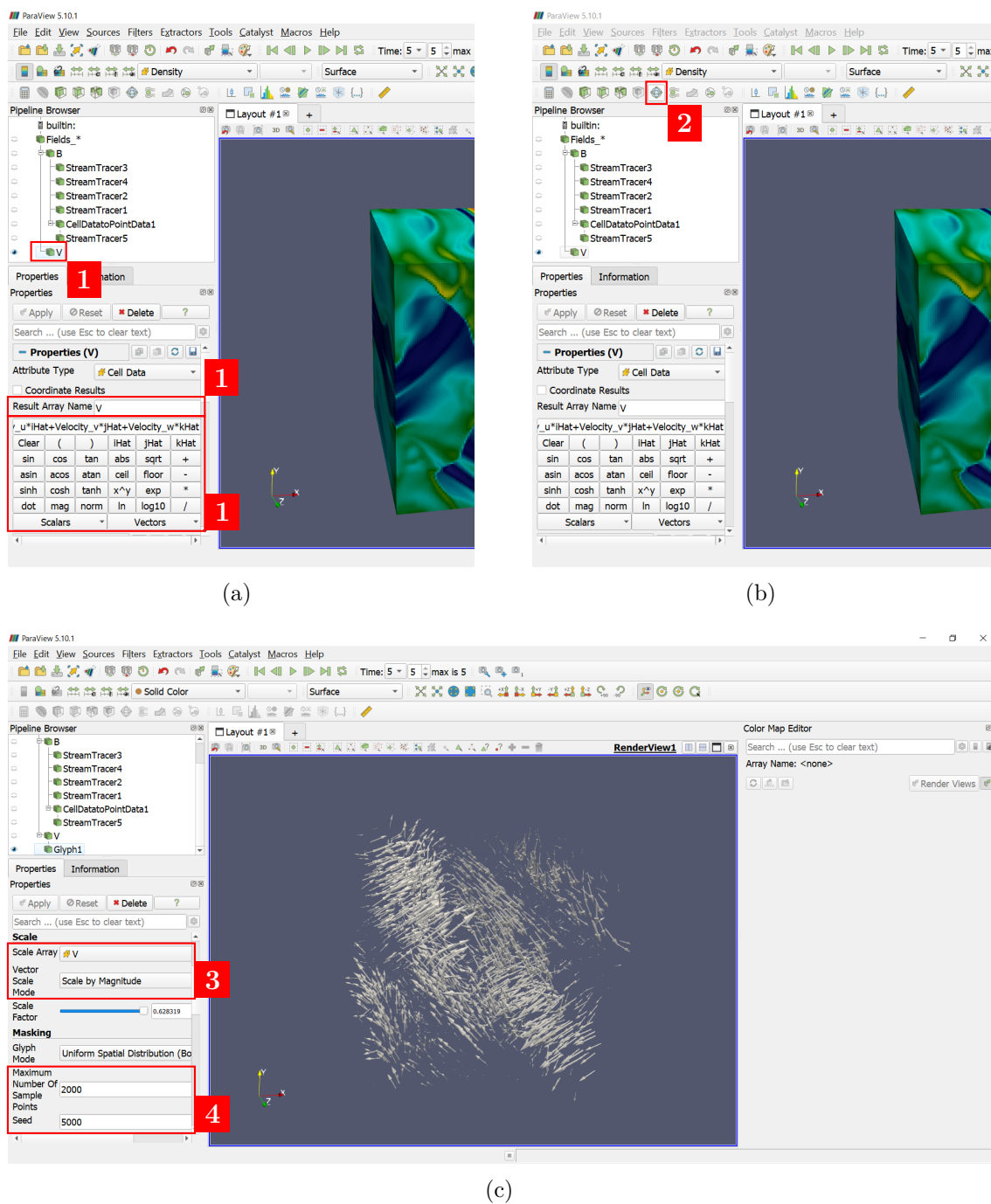
(a)



(b)

DICA: TESTE DIFERENTES POSICIONAMENTOS DE *Line* PARA CAPTAR AS ESTRUTURAS QUE DESEJA DO CAMPO; SE SEU OBJETIVO É VISUALIZAR A DISTRIBUIÇÃO DO CAMPO EM TODO O DOMÍNIO, SE ATENTE PARA DISTRIBUÍ-LAS SIMETRICAMENTE PARA EVITAR VIÉS DE DIREÇÃO, OU CONSIDERE USAR O *Point Cloud* PARA GARANTIR UMA VISUALIZAÇÃO MAIS ISOTRÓPICA. VEJA TAMBÉM O FILTRO 2D DA SEÇÃO 4.4.4.

Figura 4.4 - Visualizando vetores como setas com o filtro *Glyph*.



DICA: PARÂMETROS DO **painel de propriedades** como *Resolution*, *Number of Points*, *Seed* E ETC. CONTROLAM A QUANTIDADE DE VETORES RENDERIZADOS; TESTE DIFERENTES VALORES ATÉ OBTER A VISUALIZAÇÃO DESEJADA, MAS COMECE COM NÚMEROS BAIXOS (POR EXEMPLO, METADE DO VALOR PADRÃO) POIS O COMPUTADOR PODE TRAVAR DURANTE O PROCESSAMENTO.

4.4 Outros filtros

As seções anteriores introduziram os filtros *Calculator*, *Stream Tracer*, *Cell Data to Point Data* e *Glyph*. Até a versão 5.10.1, o **ParaView** dispõe de mais de 200 filtros, todos acessados pelas opções *Filters > Alphabetical* na **barra de menu** ou pelo atalho Ctrl + barra de espaço. Filtros cujo nome aparecem apagados não são aplicáveis ao dado então selecionado no **pipeline browser**. Por exemplo, *Stream Tracer* e *Glyph* só são aplicáveis aos dados vetoriais B e V e não a *Fields_**. Neste universo de filtros, existem alguns que são bem simples e muito importantes para a análise dos resultados do AMROC-MHD, e por isso são apresentados nas seções seguintes.

4.4.1 Filtro *Clip*

O filtro *Clip* secciona o domínio a partir de planos de corte. Ele é útil para retirar partes dos domínios 2D e 3D, e pode ser usado como nas Figuras 4.5(a)–(c):

- 1** selecione o dado *Fields_** e clique no filtro *Clip* na **barra de ferramentas**;
- 2** neste quadro o plano é posicionado: *Origin* marca o centro do plano, no caso em (π, π, π) , e *Normal* a direção da normal, no caso apontando para a direção $-x$ com a definição $(-1, 0, 0)$; mantenha esses valores para o primeiro plano
- 3** repita os passos anteriores, agora direcionando a normal para a direção $+y$

$$\text{Clip2: } \textit{Normal} = (0, 1, 0)$$

e aplique; se preferir, desmarque a opção *Show Plane* para esconder o plano de corte e visualizar somente o perfil de densidade do plasma, agora parcialmente visível no interior da simulação

4.4.2 Filtro *Slice*

Em algumas situações, a visualização precisa ser feita na parte interior do domínio 3D; e não na superfície. Nesses casos, o filtro *Slice* corta o domínio em planos de visualização, conforme mostrado nas Figuras 4.6(a)–(c):

- 1** selecione o dado *Fields_** e clique no filtro *Slice* na **barra de ferramentas**

- 2 defina o posicionamento do plano em *Origin* e a direção do corte em *Normal*; mantenha o primeiro em (π, π, π) , o centro do domínio; já o segundo, que define a normal do plano, mude para a direção $+x$ com a definição $(1, 0, 0)$
- 3 repita os passos anteriores, agora direcionando a normal para $+y$ e $+z$ respectivamente com as seguintes definições:

Slice2: *Normal* = $(0, 1, 0)$,

Slice3: *Normal* = $(0, 0, 1)$

- 4 após aplicar o último filtro *Slice*, selecione e visualize o dado *Glyph1* e mude a propriedade *Coloring* no **painel de propriedades** para seguir os valores da densidade; essa visualização simultânea mostra a distribuição e dinâmica do plasma dentro do vortex

4.4.3 Filtro *Plot Over Line*

Para acessar mais quantitativamente as variações espaciais, o filtro *Plot Over Line* gera gráficos dos dados ao longo de uma reta que cruza o domínio. Nas Figuras 4.7(a)–(c), a variável densidade é plotada ao longo do eixo y (entre 0 e 2π) numa reta passando pelos pontos fixos $x = \pi$ e $z = 2\pi$:

- 1 selecione o dado *Fields_** e clique no filtro *Plot Over Line* na **barra de ferramentas**
- 2 defina a origem e o final da reta respectivamente com *Point1* em $(\pi, 0, 2\pi)$ e *Point2* em $(\pi, 2\pi, 2\pi)$
- 3 no **painel de propriedades**, vá na propriedade *Series Parameters*, desmarque todas as variáveis e em seguida marque somente a densidade antes de aplicar o filtro; ao terminar, feche a **janela de ferramentas** para expandir o **layout** e visualizar apenas o resultado, que detalha a variação espacial da densidade globalmente (em todo o domínio) à esquerda e localmente (somente sobre a reta) à direita

4.4.4 Filtro *Evenly Spaced Streamlines 2D*

Além do *Stream Tracer*, um outro filtro que permite traçar as linhas de um campo vetorial é o *Evenly Spaced Streamlines 2D*. Ele foi desenvolvido para resolver o pro-

blema de má distribuição e adensamento de linhas que podem ocorrer devido ao posicionamento das fontes *Line* e *Point Cloud* (das Seções 4.2.1 e 4.2.2). Ele permite total controle da densidade e conectividade das linhas com cobertura ampla e homogênea. Detalhes do seu algoritmo estão em (JOBARD; LEFER, 1997) e (LIU et al., 2006).

Apesar de poderoso, sua aplicação tem limitações por funcionar apenas em duas dimensões (ainda não existe para 3D) e por ser computacionalmente custoso, de modo a forçar o encerramento do **ParaView** dependendo do seu uso. Por isso é importante testá-lo em uma seção dedicada, conforme feito na Figura 4.8(a)–(c). Delete os filtros e dados do **pipeline browser**, ou feche e abra uma nova seção do **ParaView** (para salvar o estado atual da seção antes de encerrá-la, veja a Seção 5.4 do Capítulo 5). Em seguida, carregue somente o arquivo referente ao último instante de tempo da simulação Orszag-Tang Vortex 3D e prossiga com os seguintes passos:

- 1 selecione o filtro *Slice*
- 2 clique em *Z Normal* para apontar a normal na direção z e aplique
- 3 selecione a calculadora
- 4 como o corte do domínio foi realizado sobre o eixo x – y , calcule o vetor campo magnético usando somente as coordenadas destes eixos:

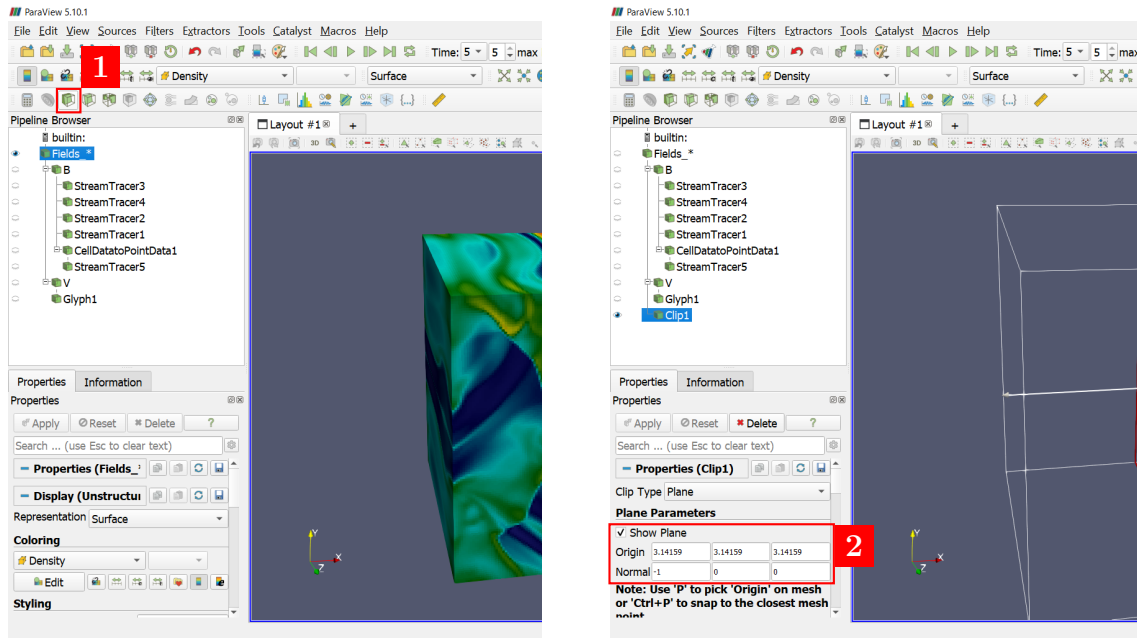
$$\text{"Y_01_}_Bx"*i\text{Hat} + \text{"Y_02_}_By"*j\text{Hat},$$

renomeie o resultado e o dado para B_{2D} e aplique

- 5 usando o atalho Ctrl + barra de espaço, procure pelo filtro *Evenly Spaced Streamlines 2D* e tecle Enter. Os parâmetros deste filtro funcionam da seguinte maneira: a partir de uma posição inicial do domínio (*Start Position*), diversas fontes são automaticamente criadas respeitando-se uma distância de separação (*Separating Distance*). Uma linha é construída a partir da fonte até a borda do domínio ou até chegar a uma distância $d = \text{Separating Distance} * \text{Separating Distance Ratio}$ de outra. Duas linhas são consideradas conectadas se elas possuem dois pontos numa distância $d < \text{Closed Loop Maximum Distance}$. Essa integração é baseada no método de Runge–Kutta, estando disponíveis os tipos (*Integrator Type*) de segunda e quarta ordem. Mude somente o parâmetro *Closed Loop Maximum Distance* de 0.2 para 0.1 e aplique

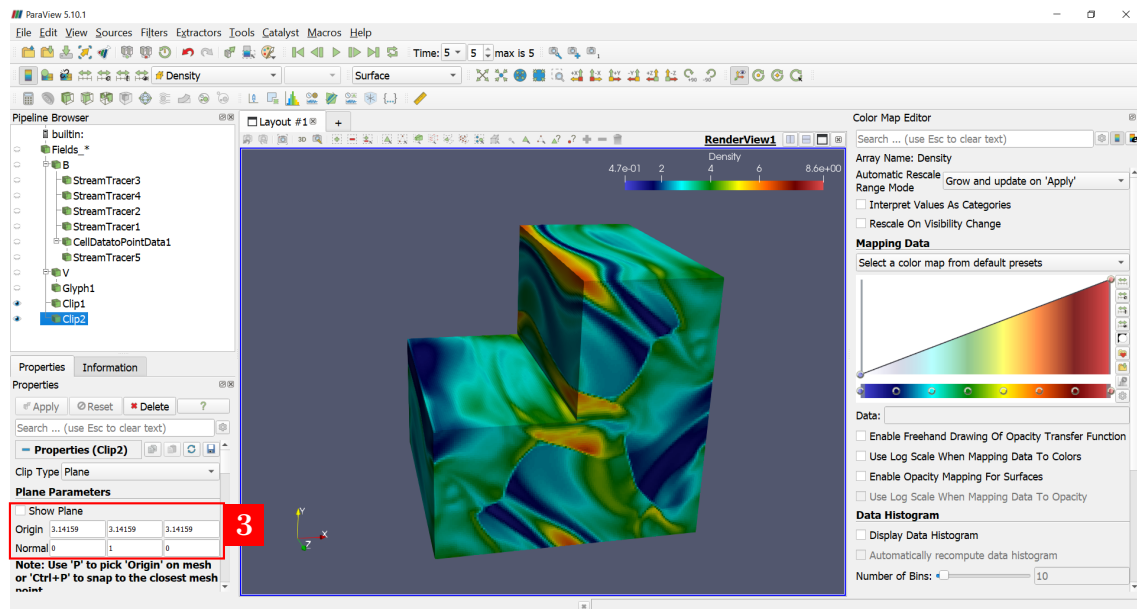
- 6** selecione para visualização o campo magnético 2D
- 7** especifique a coordenada x
- 8** mude a renderização de *Surface* para *Wireframe*
- 9** selecione o mapa de cor *Cool to Warm* para distinguir valores positivos e negativos, ou seja, a direção do campo (se para a direita ou esquerda)
- 10** aumente a espessura da renderização do *Wireframe* de 1 para 3 de modo a facilitar a visualização

Figura 4.5 - Exemplo de utilização do filtro *Clip*.



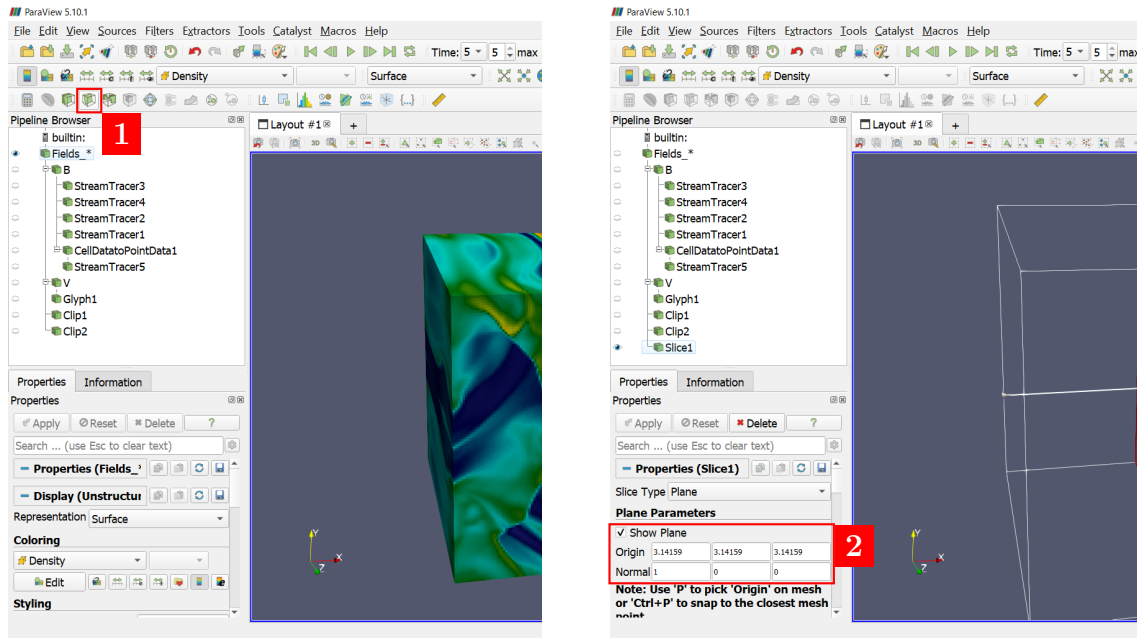
(a)

(b)



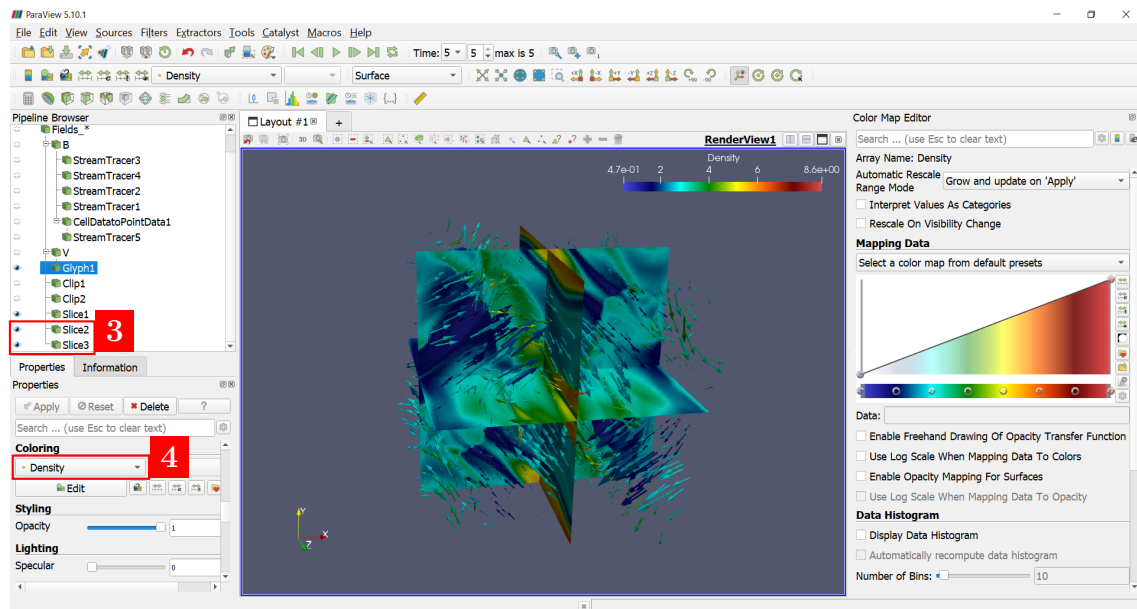
(c)

Figura 4.6 - Exemplo de utilização do filtro *Slice*.



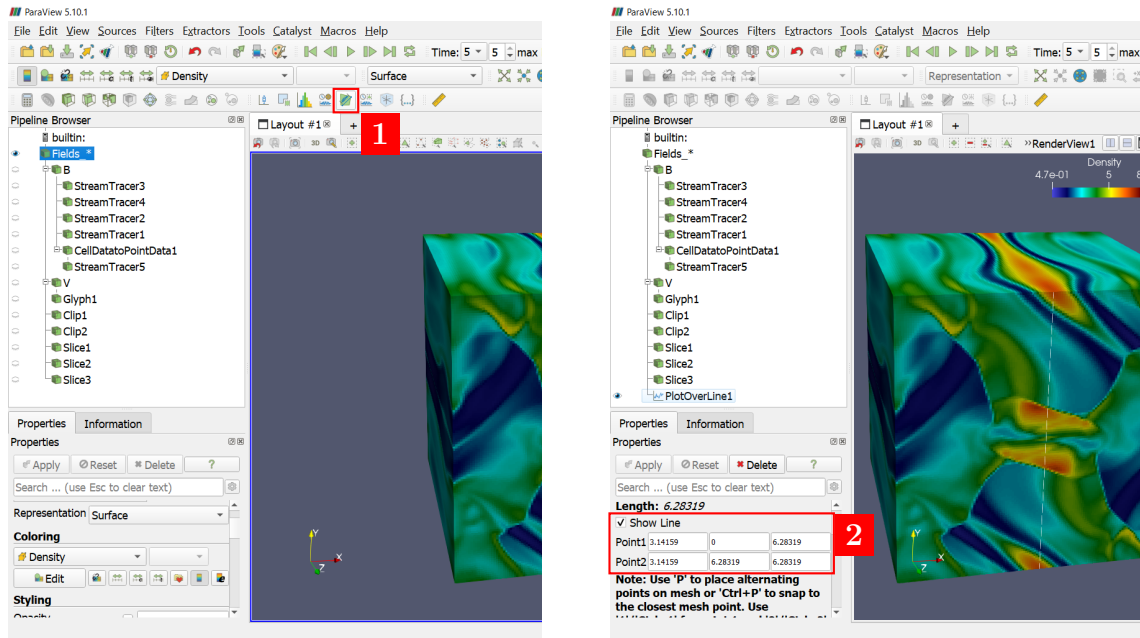
(a)

(b)



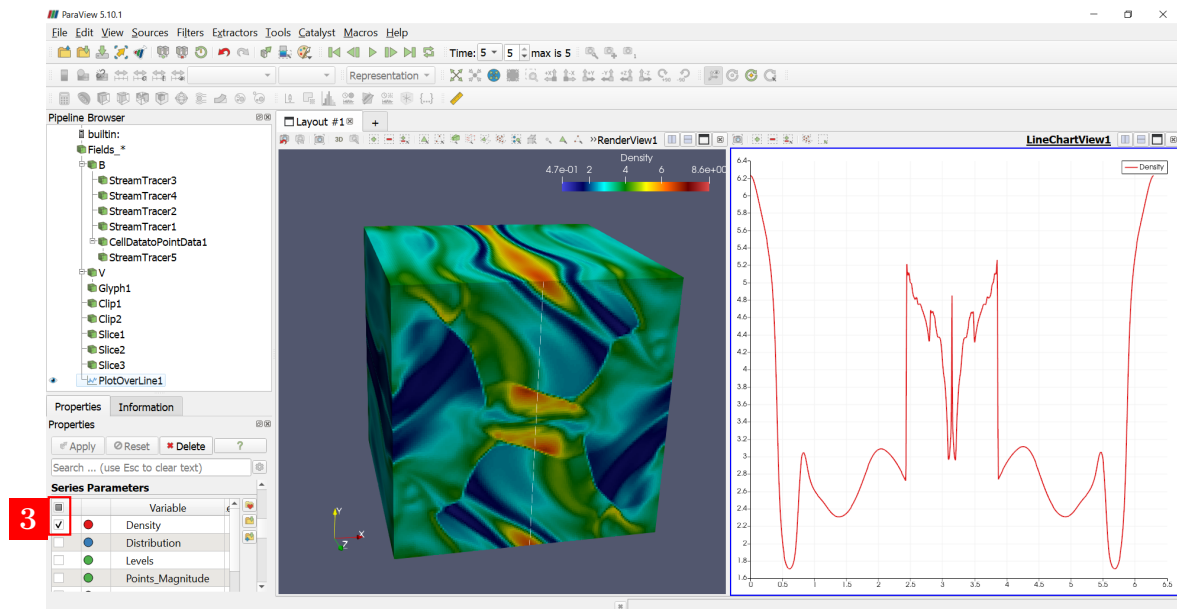
(c)

Figura 4.7 - Gerando gráfico com o filtro *Plot Over Line*.



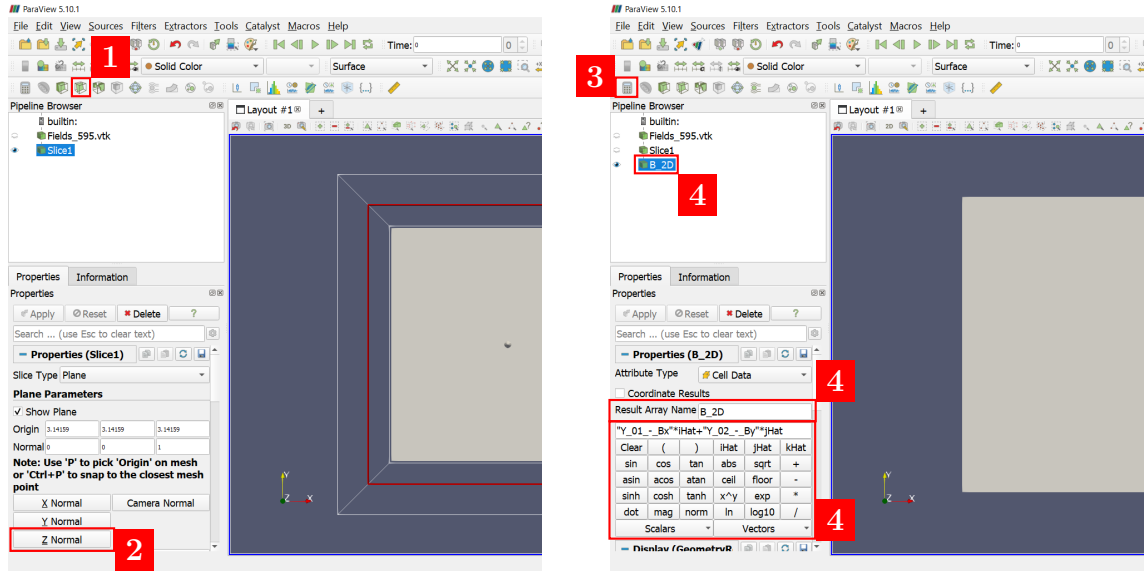
(a)

(b)



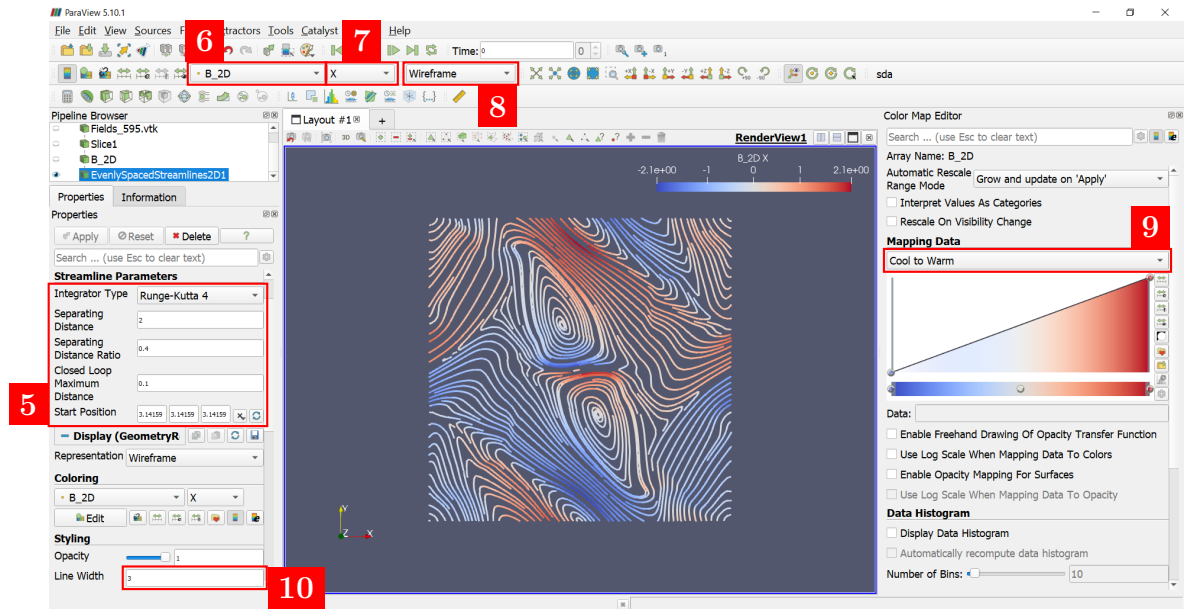
(c)

Figura 4.8 - Visualizando vetores 2D com o filtro *Evenly Spaced Streamlines 2D*.



(a)

(b)



(c)

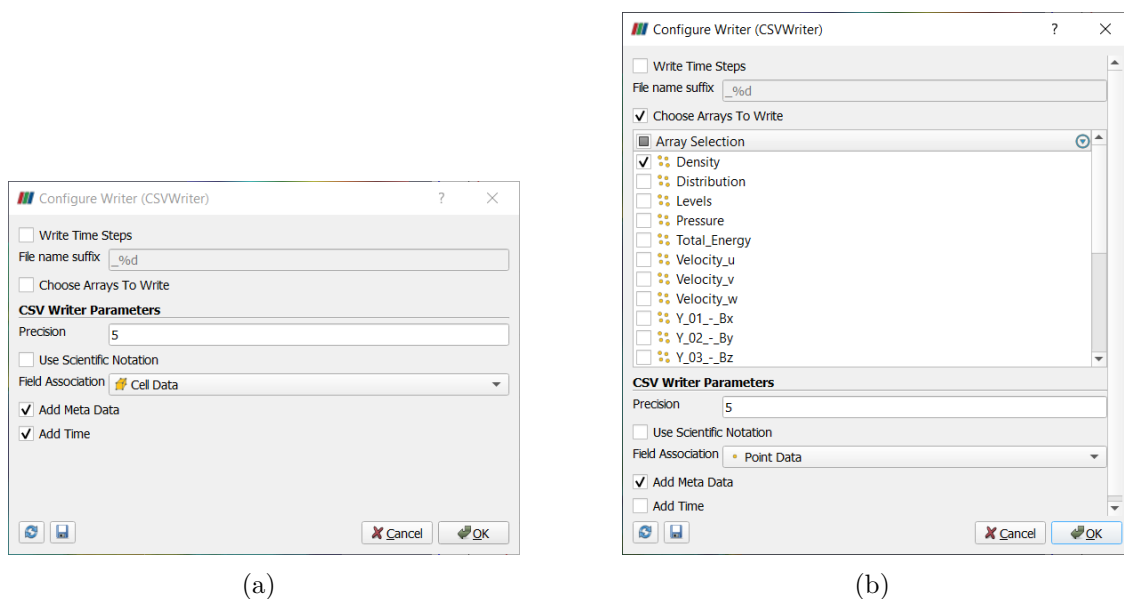
5 SALVANDO E AUTOMATIZANDO

Dados podem ser salvos de diferentes maneiras no **ParaView**. Desde valores numéricos na forma de tabelas ou planilhas até imagens e vídeos. Além disso, o estado do programa pode ser salvo para prosseguir após um encerramento e macros podem ser criadas para registrar cada mudança de estado e automatizar o trabalho. Essas ações são apresentadas a seguir.

5.1 Dados numéricos: *Save Data*

Os valores numéricos dos dados **.vtk** não são próprios para leitura direta. No caso do **AMROC**, eles contêm informações básicas no topo seguidas dos valores em binário (para detalhes, ver Capítulo 7). Para acessá-los da maneira ideal, exporte qualquer dado dentro do **ParaView** na forma de tabelas em **ascii** com a opção *Save Data*. Usando como exemplo os resultados da Figura 4.7(c), podem ser salvos tanto os valores do domínio inteiro selecionando o dado **Fields_*** no **pipeline browser** quanto os valores do plot selecionando o dado **PlotOverLine1**. Selecione um, vá na **barra de menu** e clique em **File>Save Data**. Uma janela para confirmar o formato (por padrão **.csv**) e o local aparece e, em seguida, uma janela de opções específicas. A Figura 5.1 mostra essa janela de opções para os dados **Fields_*** (a) e **PlotOverLine1** (b). No primeiro caso, a caixa *Add Time* foi marcada para adicionar uma coluna para o tempo e, no segundo, a caixa *Choose Arrays To Write* foi marcada para expandir as opções e salvar somente os valores da densidade na tabela.

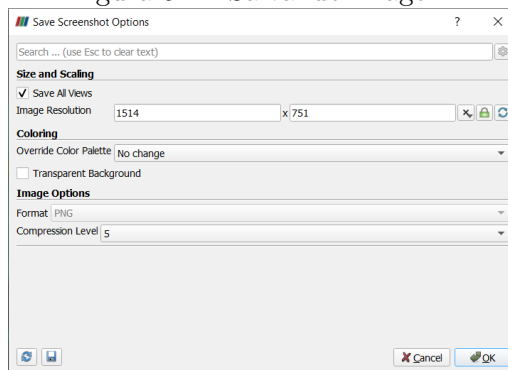
Figura 5.1 - Salvando dados numéricos do domínio (a) e do plot sobre uma linha (b).



5.2 Imagens: *Save Screenshot*

As visualizações presentes no **layout** podem ser salvas selecionando (clitando sobre) uma janela e em *File>Save Screenshot*. Os formatos incluem .png (padrão), .jpg, .tif e outros. Usando o mesmo exemplo anterior, a Figura 5.2 mostra as opções específicas para imagem, com a caixa *Save All Views* marcada para considerar todas as janelas do **layout** (do domínio e da curva de densidade lado a lado). Em *Override Color Palette* define-se a cor do fundo (no caso branco), enquanto a caixa *Transparent Background* o torna transparente.

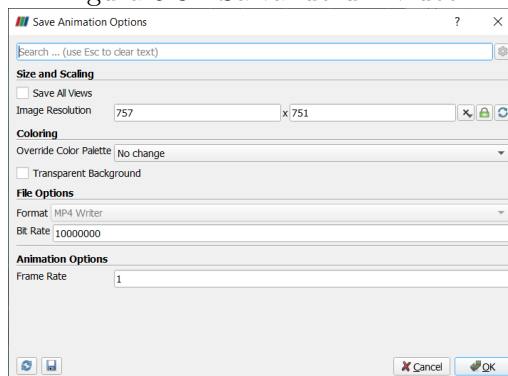
Figura 5.2 - Salvando imagem.



5.3 Vídeos: *Save Animation*

Vídeos da simulação também podem ser salvos com a opção *File>Save Animation*. Pode-se salvar uma sequência nos mesmos formatos de imagem e usar um software de empilhamento, ou salvar por exemplo em .mp4 ou .avi para obter um filme pronto. A Figura 5.3 traz as opções de salvamento para o formato .mp4. Assim como para imagem, a caixa *Save All Views* permite gerar o filme com todas as visualizações do **layout**. As opções de cor e transparência do fundo também são as mesmas.

Figura 5.3 - Salvando um vídeo.



5.4 Estado do programa: *Save State*

O estado do **ParaView** pode ser salvo como um checkpoint de trabalho para continuar as modificações mais tarde ou em outro computador (desde que os mesmos dados estejam acessíveis). Para isso, use a opção *File>Save State*. Para carregar o estado salvo, vá em *File>Load State*.

Pode-se salvar o estado no formato `.pvsm`, baseado na linguagem XML, ou em `.py` como um script Python. Esse script pode ser carregado para dentro do programa como o `.pvsm` ou executado externamente em ambiente Python, ou seja, sem a necessidade da interface gráfica. Além disso, scripts Python são úteis para automatização de tarefas via macros, introduzidas na próxima seção.

5.5 Macros e scripts Python

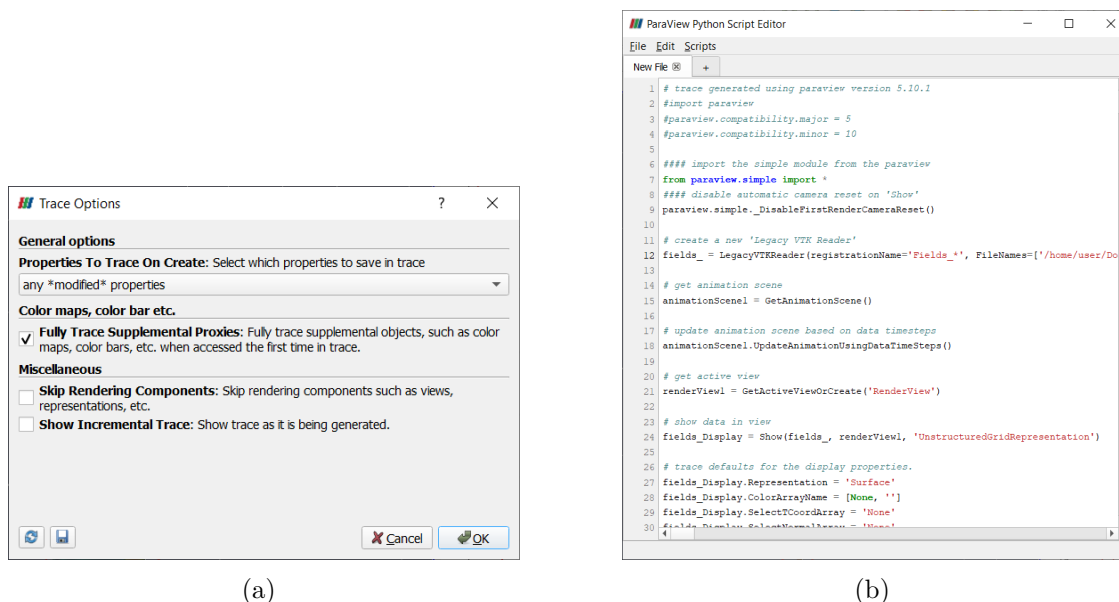
Imagine a seguinte situação: tem-se uma simulação de domínio definido (2D ou 3D e de extensão dada), mas alguns parâmetros da solução precisam ser testados. Após cada simulação de teste os novos resultados precisam ser importados e trabalhados no **ParaView**. Nesses casos é interessante ter um pipeline de filtros e salvamentos pronto para essas simulações, que pode ser aplicado sempre que novos resultados forem gerados sem sequer precisar abrir o **ParaView**, localmente ou em um cluster. Isso é possível com macros e uma distribuição Python conforme a seguir.

5.5.1 Gerando macros: *Start/Stop Trace*

Para criar uma macro completa, que inclua desde a importação do `.vtk` até a exportação de resultados, a primeira coisa que deve fazer ao abrir o **ParaView** é iniciar uma gravação clicando em *Tools>Start Trace* na **barra de menu**. A janela da Figura 5.4(a) aparece, e a opção padrão *any *modified* properties* é a ideal para gravar somente as ações mais importantes. Logo abaixo, a caixa *Fully Trace Supplemental Proxies* precisa estar marcada para incluir as propriedades de cor. Além dessas opções, *Skip Rendering Components* exclui algumas ações de visualização e *Show Incremental Trace* exhibe a construção do script a cada ação ao invés dele pronto no final. Deixe ambas desmarcadas. Após clicar em *Ok*, cada ação é gravada como um comando Python automaticamente e em segundo plano. Finalize a macro voltando à mesma opção na **barra de menu**, dessa vez como *Tools>Stop Trace*, e clique.

O resultado é um script Python como o da Figura 5.4(b), que pode ser salvo como uma macro para ser chamada na opção *Macros* da **barra de menu**. Alternativamente, pode-se salvá-lo em qualquer lugar e executá-lo sem a necessidade de interface

Figura 5.4 - Janela de início da macro (a) e script Python após finalizar a macro (b).



gráfica, bastando então uma distribuição Python. Porém, a maioria das distribuições presentes nos computadores não é compatível com o **ParaView**. A solução para executar estes scripts é discutida na próxima seção.

5.5.2 Rodando scripts Python

O **ParaView** vem com sua própria distribuição Python, chamada **pvpython**, que é automaticamente instalada. É possível executar um interpretador dentro da interface de usuário indo na **barra de menu**, *View>Python Shell*. Uma janela surge abaixo do **layout**, e pode-se clicar em *Run Script* para procurar e executar o arquivo .py da Figura 5.4(b). No entanto, esse procedimento é similar a salvá-lo e executá-lo como uma macro conforme a instrução do parágrafo anterior.

Para contornar completamente o uso da interface gráfica, supondo que o script foi salvo com o nome **macroparaview.py** ao lado dos arquivos .vtk utilizados, basta abrir o terminal neste diretório e executar a seguinte linha:

```
pvpython macroparaview.py
```

Note que para executar o **pvpython** desta forma de qualquer lugar da sua máquina o **ParaView** deve estar adicionado ao seu **\$PATH** conforme instruído no Capítulo 2. Porém essa macro foi criada com caminhos específicos para os arquivos de entrada (.vtk) e saída (tabelas e figuras) e, por isso, seus comandos só funcionam se tais

hierarquias forem mantidas.

Para executar o script de maneira mais independente, basta trocar todas as strings específicas do local por uma variável que mude automaticamente. Isso pode ser feito em Python pondo as seguintes linhas no topo do script `macroparaview.py`:

```
import os
diretorio=os.getcwd()
```

O método `getcwd()` retorna o diretório atual, que é guardado na variável `diretorio` usada para substituir as strings específicas do local. Para alterar os caminhos de entrada, procure pela variável `FileNames` dentro do seu script. Nas linhas desta variável, se há uma string como `'/home/user/Documents/Fields_0.vtk'`, troque-a por `diretorio+'/Fields_0.vtk'`. Para os caminhos de saída, procure pelos comandos iniciados com `Save`, como por exemplo `SaveScreenshot`. Se uma string dentro deste comando é `'/home/user/Desktop/screenshot.png'`, troque-a por `diretorio+'/screenshot.png'`. Com essas modificações, o script pode ser usado independente do seu local de criação. Se resultados de uma simulação semelhante forem gerados em outro local, basta copiar o script para lá e executá-lo.

6 MAPEAMENTO DE CORES

Este capítulo trata de uma das etapas mais importante do processamento de um dado, que é a definição das cores e legendas utilizadas na sua visualização. Escolhas erradas podem ocultar características importantes ou mesmo tornar a imagem final imprópria para determinadas apresentações. Além de conhecer o **ParaView**, uma visualização apropriada do resultado das simulações depende de três requisitos (FARGE, 2000):

1. um bom conhecimento do problema físico e matemático sendo resolvido numericamente (ou seja, o que está sendo apresentado) e das questões relevantes a serem levantadas na visualização de um campo (ou seja, o porque está sendo apresentado)
2. entendimento básico do funcionamento do olho humano e de como percebemos formas e cores, de modo a evitar ilusões e outros artifícios indesejados (por exemplo, a cor amarela fica indistinguível da branca ou da marrom dependendo da cor de fundo da apresentação, se muito clara ou escura)
3. consciência de aspectos culturais como simbolismos e associações, importante para reduzir o risco de interpretações errôneas por conta de significados implícitos (por exemplo, é senso comum em muitos países associar a cor vermelha a conceitos como quente, forte ou muito, e a cor azul a frio, fraco ou pouco)

Na renderização de uma imagem, a representação de um valor numérico em cor é feita por meio de um mapa de cores. Esse mapeamento também inclui opacidade e, portanto, recebe o nome de *função de transferência*, um termo mais geral para a transformação (mapeamento) de dados em cor. No **ParaView**, as funções de transferência de cor e opacidade são editadas separadamente. Essas e demais opções de renderização da imagem são acessadas na **janela de ferramentas** utilizando o *View>Color Map Editor*. Este editor está apresentado na Figura 6.1 e suas principais opções são:

- 1** botão para edição das propriedades de legenda da exibição, que incluem cor, fonte e tamanho dos textos, título para a barra de cor, sua espessura e largura, precisão dos seus valores numéricos, etc.
- 2** opção para interpretar os valores do dado em categorias, interessante para o caso deste conter apenas valores discretos (por exemplo, um campo que

varia entre os inteiros de 1 a 10); o mapa de cor assume as cores das categorias criadas após marcar esta caixa

3 selecionador de mapas de cores; ele exibe a lista padrão (*Default*) de mapas do **ParaView**; mas muitos outros estão disponíveis. Os mapas estão separados em categorias baseadas nas suas funções, entre elas: *Linear*, contendo mapas de uma só matiz que mudam a luminosidade e saturação da cor incrementalmente, normalmente destinados a dados que apresentam ordenação; *Diverging*, que muda a luminosidade e saturação de duas cores diferentes que se encontram no meio em uma cor não saturada, e indicados para campos que precisam de clara distinção entre valores negativos e positivos; *Isoluminant*, com mapas que mantém uma luminância constante variando apenas a matiz, recomendados para a renderização de estruturas 3D; e *Rainbow*, cujos mapas se baseiam nas cores da luz nos diferentes comprimentos de onda, e apesar de muito comuns, são problemáticos em situações de impressão preto e branco, má iluminação de fundo, ou para audiências com daltonismo. Uma vez escolhido o mapa, suas funções de transferência de opacidade e cor podem ser editadas nas áreas marcadas logo abaixo, que são:

3.1 editor da função de transferência de opacidade; pode-se clicar em qualquer parte deste painel para selecionar um ponto de controle associado a um valor do dado, e arraste para cima ou para baixo para alterar seu valor de opacidade (entre 0 e 1); clicar numa área vazia cria um ponto de controle novo, e teclar em Delete deleta um ponto selecionado; os pontos das extremidades do triângulo, de coordenadas ($\text{densidade}_{\min}, 0$) e ($\text{densidade}_{\max}, 1$), não podem ser deletados. A edição de opacidade é desativada por padrão para a renderização de superfícies (ver opções do item 5 abaixo)

3.2 editor da função de transferência de cor; dê dois cliques num ponto de controle para mudar a cor associada ao valor selecionado, e arraste um ponto lateralmente para mudar os intervalos do mapa de cor; criar e deletar pontos de controle funcionam tal qual para a opacidade, mas este está restrito apenas à movimentação da esquerda para a direita, do ponto mapeando o menor valor do dado (densidade_{\min}) até ao de maior valor (densidade_{\max}), que não podem ser deletados nem movidos

3.3 caixa para edição do ponto de controle; ao invés de arrastar um ponto nos painéis de opacidade ou cor, selecione-o e edite seu valor numérico

diretamente nesta caixa

- 4** os 9 botões à direita das funções de transferência fazem as seguintes ações (de cima para baixo): (1) redefine o valor mínimo e máximo da visualização para os valores presentes no dado selecionado no **pipeline browser**, reescalando assim todo o mapa de cores; (2) redefine o máximo e mínimo para os valores entrados pelo usuário (abre uma janela para escrita); (3) os redefine com base no máximo e mínimo de toda a simulação; (4) os redefine com base no tempo sendo exibido; (5) inverte a função de transferência de cor; (6) abre a janela de seleção de mapas de cor e suas categorias; (7) salva a função de transferência sendo personalizada, que pode depois ser encontrada dentro das categorias *User* e *Default*; (8) computa um histograma para os dados (ver caixa de opções 6 abaixo) ; (9) permite edição manual da função de transferência (expande uma tabela com os pontos de controle e suas cores para edição)
- 5** as 4 caixas abaixo das funções de transferência, quando marcadas, ativam as seguintes propriedades: (1) permitem editar a opacidade desenhando diretamente no painel ao invés de arrastando pontos de controle; (2) ativa a escala logarítmica no mapeamento de cor (requer valores não-nulos e positivos, caso contrário retornará erro); (3) ativa a função de transferência de opacidade para superfícies, ativada por padrão apenas para renderização de volumes (ver opção 3 da Figura 3.3); (4) ativa a escala logarítmica no mapeamento de opacidade
- 6** a primeira caixa neste conjunto de opções ativa o histograma do dado (a ser exibido na área da edição de opacidade, acima da barra de cor), que quando ativado pode ter seu número de barras (10 por padrão) alterado mais abaixo
- 7** muda o espaço de cor, ou seja, a maneira de interpolação dos dados; sua escolha não altera a cor de um ponto de controle, mas pode alterar drasticamente as cores entre eles
- 8** define a cor e opacidade usados para representar valores inválidos nos dados
- 9** define o número de cores disponíveis para o mapeamento (256 por padrão)

Como exemplo, a Figura 6.2 ilustra a visualização do campo de divergência magnética ou, em linguagem físico-matemática, $\nabla \cdot \mathbf{B}$. Esse campo escalar soma as derivadas

do campo magnético nas três direções espaciais, sendo positivo (negativo) quando mais linhas do campo magnético saem de (chegam em) algum ponto no espaço do que o oposto. Essa soma deve ser zero (mesma quantidade de linhas saindo e chegando) para respeitar a chamada Lei de Gauss do magnetismo, que explica a inexistência de monopolos magnéticos. Devido a erros inerentes à aproximação numérica, ferramentas como o **AMROC-MHD** precisam forçar a Lei de Gauss às equações do modelo com alguma técnica que evita acúmulo de $\nabla \cdot \mathbf{B}$ e/ou reduz sua magnitude. Sendo assim, visualizar a divergência ajuda a investigar a eficiência da técnica empregada e a consistência física do resultado.

Figura 6.1 - Janela de edição das funções de transferência do ParaView.

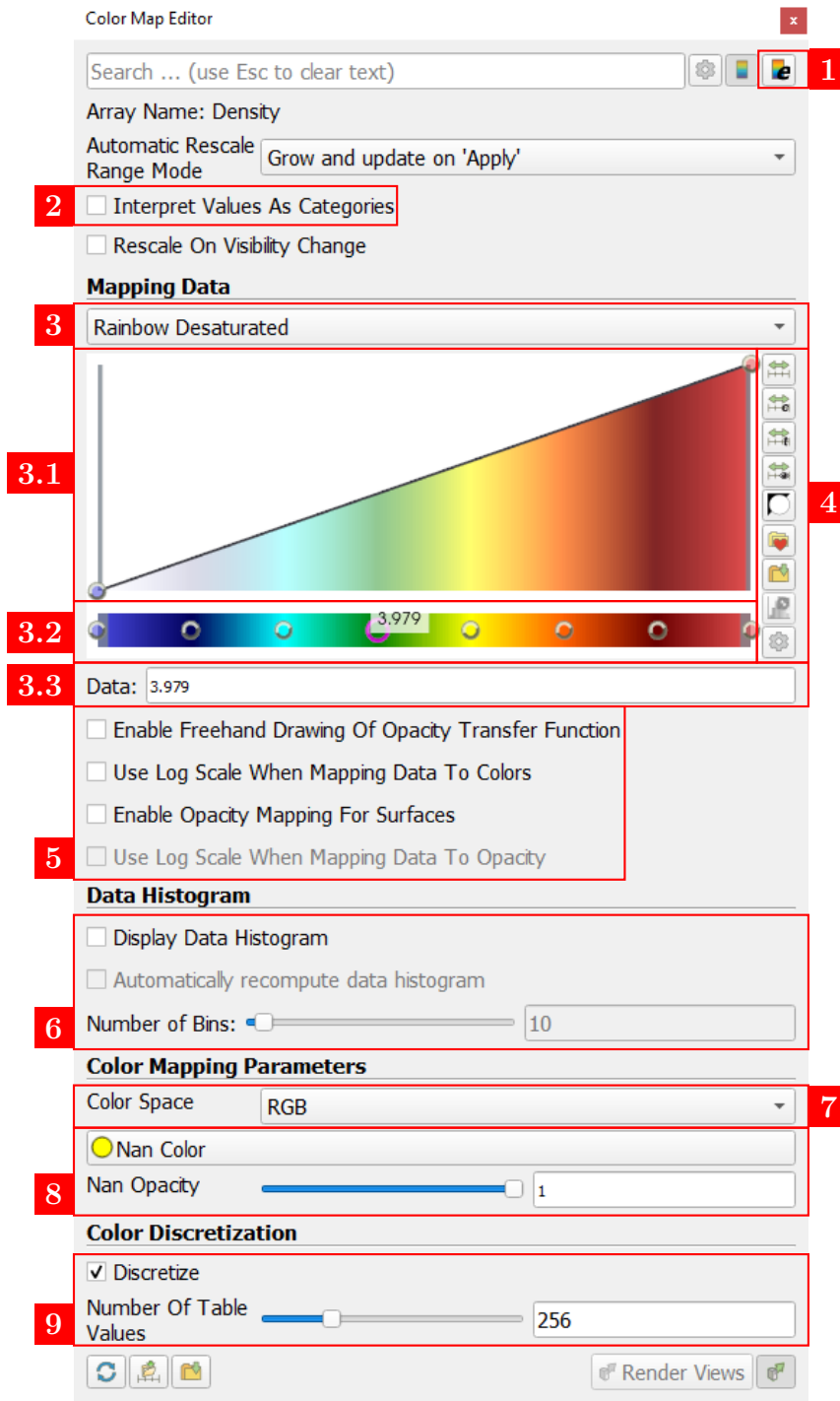
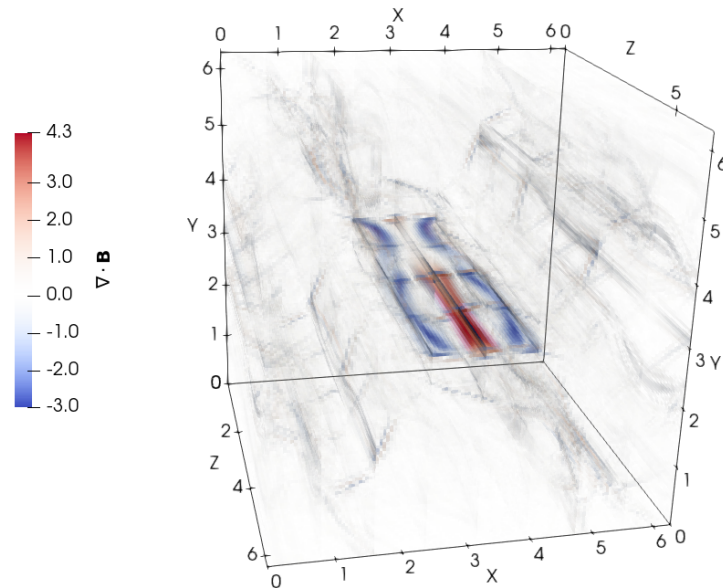


Figura 6.2 - Visualização do campo de divergência magnética $\nabla \cdot \mathbf{B}$.



Fonte: Reprodução do autor.

Os passos nas Figuras 6.3, 6.4 e 6.5 mostram como gerar a visualização do campo de divergência magnética da Figura 6.2. Na primeira se edita a cor do fundo e os eixos do domínio, na segunda as legendas da barra de cor, e por último as funções de transferência. Iniciando com a Figura 6.3:

- 1** mude a visualização para o campo de divergência
- 2** ative a visualização dos eixos
- 3** clique no botão para abrir a janela que edita a aparência dos eixos
- 4** clique no botão de engrenagem para expandir as opções de edição e altere-as conforme a figura
- 5** ative a paleta de cores para o fundo
- 6** mude a cor de fundo para branco

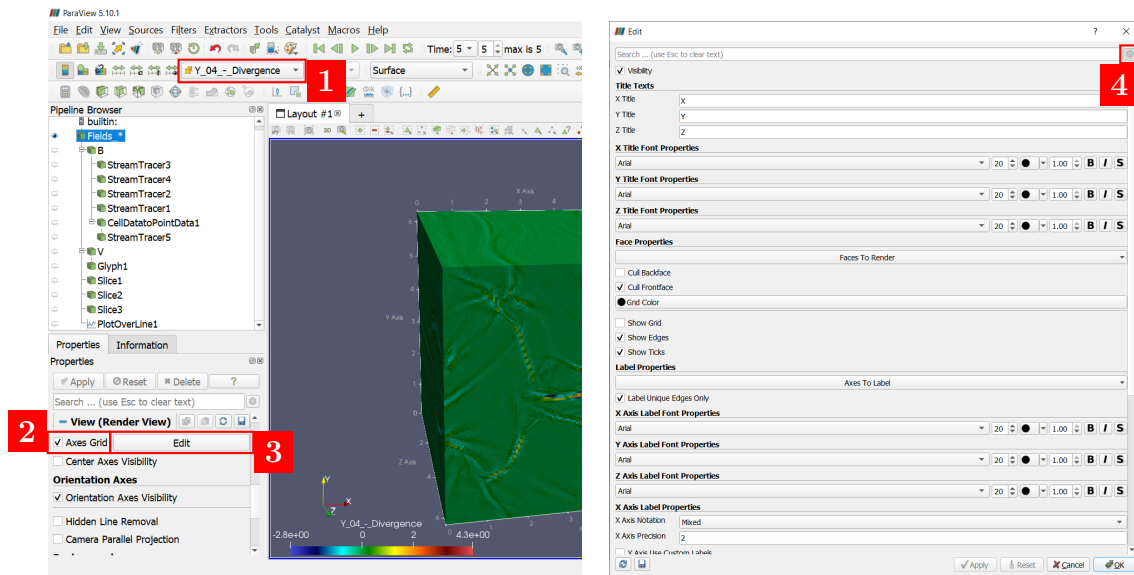
Para alterar as legendas, siga os passos destacados na Figura 6.4:

- 1** clique no botão para abrir a janela de edição de legendas
- 2** clique na engrenagem para expandir as opções desta janela e altere-as conforme a figura
- 3** desative a visualização do ícone que indica a orientação dos eixos
- 4** clique sobre a barra de cor e arraste-a até a posição indicada

Para finalizar a renderização, altere as funções de transferência como indicado na Figura 6.5:

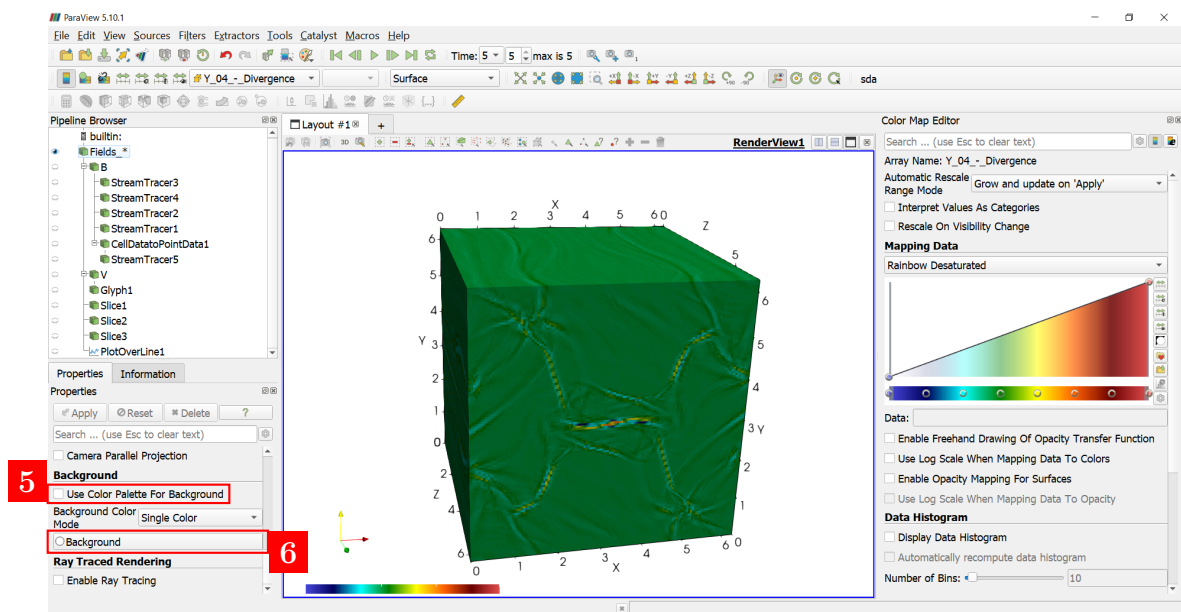
- 1** ative a visualização do histograma dos dados
- 2** ative a edição de opacidade
- 3** altere a barra de cor para *Cool to Warm*
- 4** na área da função de transferência de cor, clique sobre o ponto de controle do meio para selecioná-lo
- 5** na caixa abaixo, mude seu valor de mapeamento para zero
- 6** agora na área da função de transferência de opacidade, clique e arraste o primeiro ponto (à esquerda) até o topo para mapeá-lo na opacidade máxima (igual a 1)
- 7** clique numa região qualquer desta área para criar um novo ponto de controle de opacidade, e arraste-o até a base para mapeá-lo na opacidade mínima (igual a 0)
- 8** na caixa abaixo, finalize mudando o valor de mapeamento deste novo ponto para zero; salve a imagem em *File>Save Screenshot*

Figura 6.3 - Editando fonte e cor dos eixos do domínio e a cor de fundo da imagem.



(a)

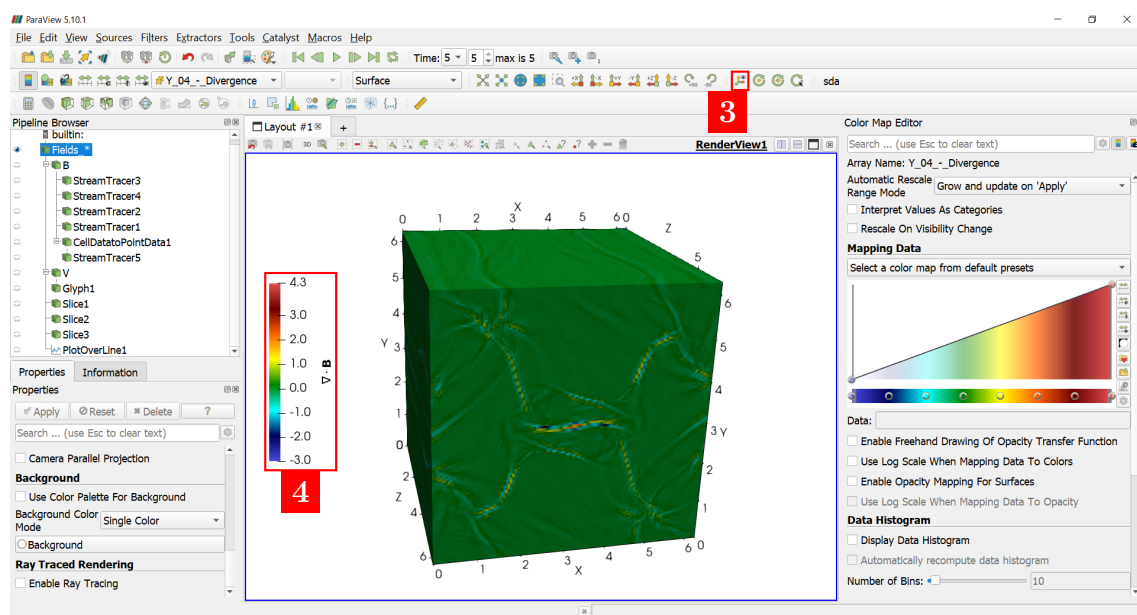
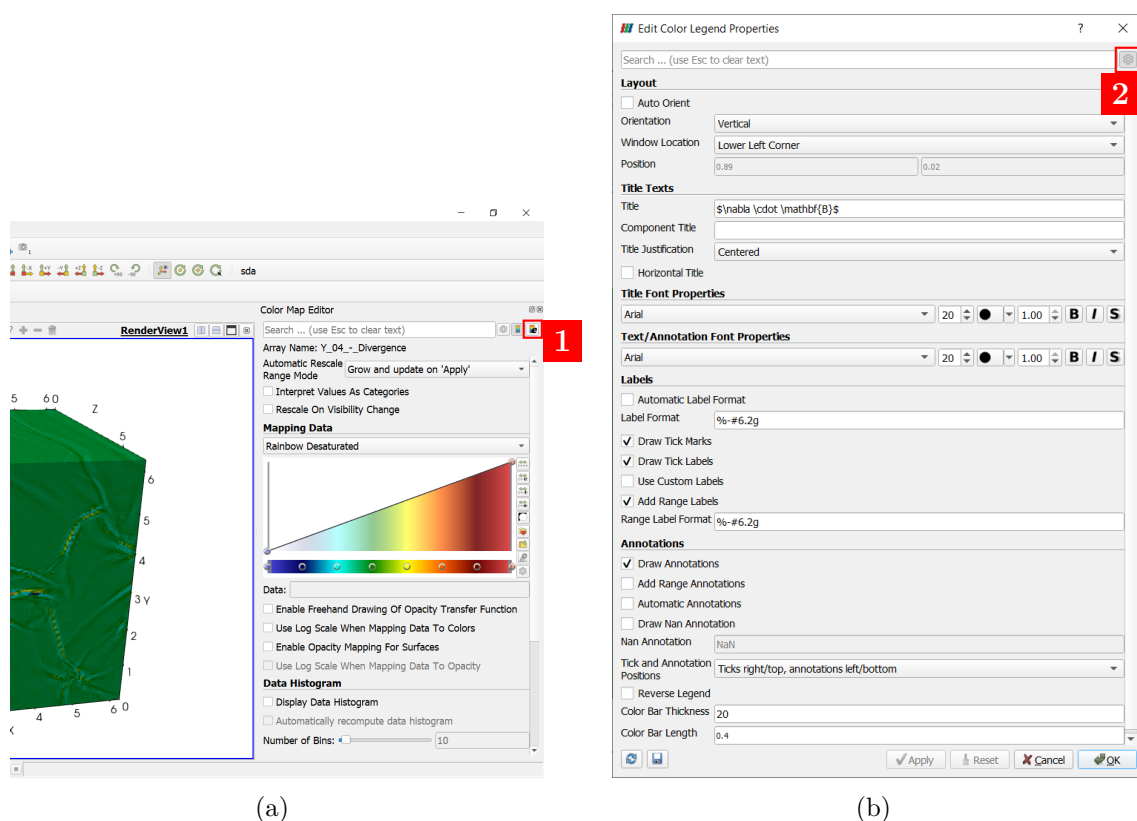
(b)



(c)

DICA: LEMBRE-SE QUE A DEFINIÇÃO DO MAPA DE COR É TÃO IMPORTANTE QUANTO A ESCOLHA DA COR DE FUNDO DA IMAGEM, UMA VEZ QUE O FUNDO ALTERA NOSSA CAPACIDADE DE PERCEBER CONTRASTES. SUA ESCOLHA PODE FACILITAR OU DIFICULTAR A DISTINGUIR AS DIFERENTES INFORMAÇÕES CONTIDAS NUMA VISUALIZAÇÃO.

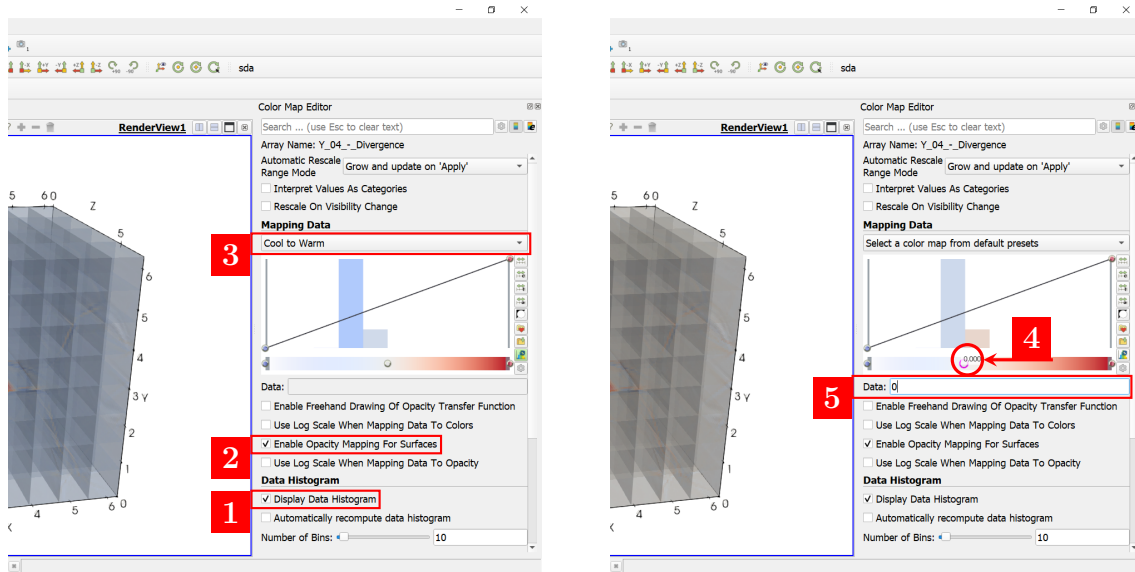
Figura 6.4 - Editando as propriedades de texto da barra de cor.



(c)

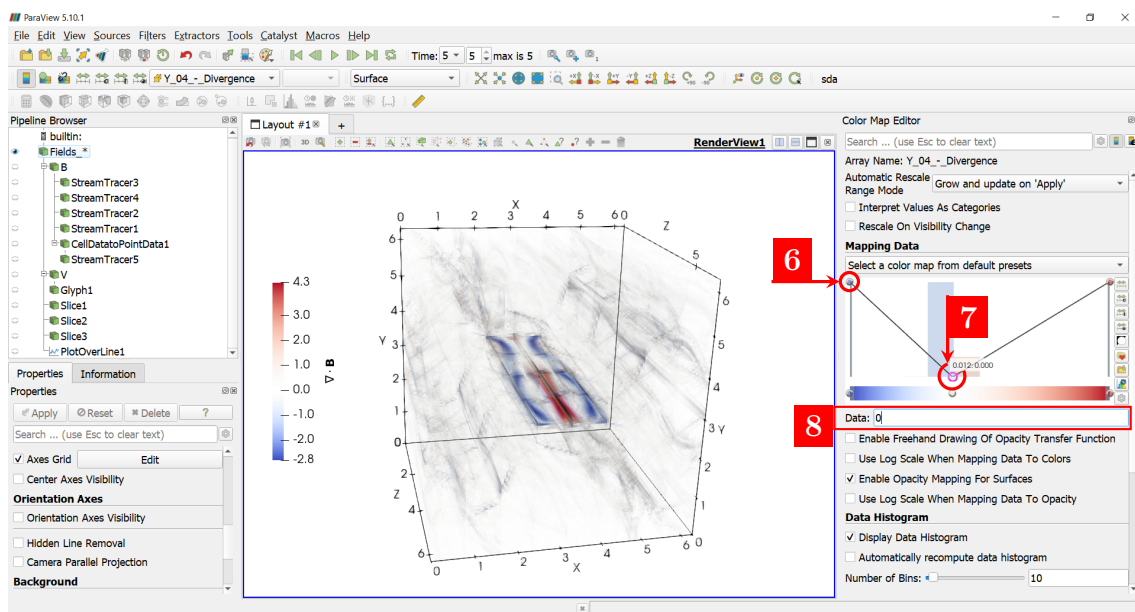
DICA: O PARAVIEW É CAPAZ DE RENDERIZAR EXPRESSÕES MATEMÁTICAS EM TEX QUANDO ESCRITAS DENTRO DE UM PAR \$\$\$. PARA RENDERIZAR UM SINAL DE DÓLAR SEM INICIAR UMA EQUAÇÃO, USE \\$.

Figura 6.5 - Editando as funções de transferência de opacidade e cor.



(a)

(b)



(c)

DICA: ATIVAR O HISTOGRAMA PERMITE VISUALIZAR A DISTRIBUIÇÃO DOS VALORES AO LONGO DA BARRA DE COR, O QUE PODE AUXILIAR MUITO NO USO DAS FUNÇÕES DE TRANSFERÊNCIA PARA O DESTAQUE DE INFORMAÇÕES ESPECÍFICAS. O HISTOGRAMA TAMBÉM PODE INFORMAR A COR PREDOMINANTE DO MAPEAMENTO COM BASE NA MODA DA DISTRIBUIÇÃO.

7 FORMATO DE DADOS .VTK

O Visualization Toolkit (VTK) (HANWELL et al., 2015) é uma biblioteca de código aberto para processamento e visualizações de dados. Ele suporta diversos algoritmos para realizar análises científicas e estatísticas, manipular e combinar objetos, gerar renderizações 2D e 3D, entre outros. Multiplataforma, a biblioteca VTK foi construída primariamente em C++ e tem grande interoperabilidade com Tcl, Java e Python. Por sua vez, a ferramenta de visualização **ParaView** utiliza o VTK na sua construção. Tanto o Visualization Toolkit quanto o **ParaView** são desenvolvidos pela Kitware Inc. (<<https://www.kitware.com/>>) e disponibilizados livremente.

Com o objetivo de oferecer uma representação poderosa e consistente a ser usada em várias aplicações e plataformas, o Visualization Toolkit também oferece seu próprio formato de arquivo, o .vtk. Pela sua capacidade, esse é o formato recomendado para utilização no **ParaView** e eles podem ser de dois tipos: baseados na sintaxe XML, que suporta muitos recursos, e os chamados *legacy*, menos flexíveis porém mais simples de serem trabalhados. Este será o formato tratado neste capítulo. A Figura 7.1 introduz a estrutura básica de um *legacy* VTK, que consiste de 5 partes:

Parte 1: Identificador da versão, uma linha `# vtk DataFile Version x.x`, sendo `x.x` o número da versão

Parte 2: Cabeçalho, uma string de no máximo 256 caracteres para entitular o dado

Parte 3: Formato usado na codificação do dado, se ASCII ou binário; deve ser indicado com uma única palavra-chave, `ASCII` ou `BINARY`

Parte 4: Essa parte se inicia com a palavra-chave `DATASET` seguida por um desses tipos: `STRUCTURED_POINTS`, `STRUCTURED_GRID`, `UNSTRUCTURED_GRID`, `POLYDATA`, `RECTILINEAR_GRID`, ou `FIELD`. Aqui se define a geometria e topologia dos dados, ou seja, sua estrutura espacial. Essa parte se estende com mais palavras-chave e definições específicas para cada tipo de `DATASET`

Parte 5: Essa parte se inicia com a palavra-chave `POINT_DATA` ou `CELL_DATA` seguida de um número inteiro referente ao total de pontos ou células, e é destinada aos atributos dos dados. Os atributos podem ser `SCALARS`, `VECTORS`, `TENSORS`, `LOOKUP_TABLE` e `FIELD`, e diversos atributos podem ser escritos em sequência. Por exemplo, dois tipos de `FIELD`, o campo escalar densidade e pressão, podem constituir o mesmo arquivo .vtk

Figura 7.1 - Estrutura básica de um arquivo *legacy* .vtk.

```

# vtk DataFile Version 2.0           (1)
Really cool data                     (2)
ASCII | BINARY                       (3)
DATASET type                       (4)
...
POINT_DATA n                       (5)
...
CELL_DATA n
...

```

Fonte: Adaptado de Kitware, Inc. (2010).

As três primeiras partes são obrigatórias, enquanto as partes 4 e 5 são opcionais. Uma descrição exhaustiva das palavras-chave e demais configurações de um arquivo .vtk foge ao objetivo desta nota técnico-científica. Para aprofundamento, o guia de usuário do VTK disponibilizado pela Kitware, Inc. (2010) detalha e exemplifica a construção das 5 partes deste arquivo, e uma versão mais curta pode ser encontrada no seguinte link: <<https://vtk.org/wp-content/uploads/2015/04/file-formats.pdf>>. Dito isto, um modelo de construção de um arquivo equivalente à saída do AMROC é apresentado na sequência. Ele incorpora as palavras-chave e atributos usados em cada uma das 5 partes da Figura 7.1. Nesta representação, as três primeiras linhas contém a versão, o cabeçalho e o formato do arquivo, e equivalem às partes 1, 2 e 3. A parte 4 está definida da linha 4 à 22, e a quinta e última da linha 24 em diante.

Caso necessário, você pode converter um .vtk binário para ASCII e vice-versa dentro do ParaView. Com o arquivo importado, basta clicar em *File>Save Data*, manter a opção .vtk e escolher a codificação nas configurações avançadas que seguem. Vale ressaltar que o ParaView trabalha com inúmeros formatos, incluindo simples tabelas .csv ou .txt com valores em coluna, composto ou não de cabeçalhos e separados por vírgula, espaços, etc. O filtro *Table To Points* converte três colunas destes arquivos para valores de coordenadas x, y, z . Essa visualização rápida pode ser complementada associando um mapa de cor a uma quarta coluna contendo medições de um campo escalar, de forma similar ao feito na Figura 4.4(c).

```

1 # vtk DataFile Version 3.0
2 HDF data created with AMROC's hdf2file
3 BINARY
4 DATASET UNSTRUCTURED_GRID
5 POINTS n dataType
6 p0x p0y p0z
7 p1x p1y p1z
8  $\vdots$ 
9 p(n-1)x p(n-1)y p(n-1)z
10 CELLS n size
11 numPoints0, i, j, k, l, ...
12 numPoints1, i, j, k, l, ...
13 numPoints2, i, j, k, l, ...
14  $\vdots$ 
15 numPointsn, i, j, k, l, ...
16
17 CELL_TYPES n
18 type0
19 type1
20 type2
21  $\vdots$ 
22 typen-1
23
24 CELL_DATA n
25 FIELD dataName numArrays
26 arrayName0 numComponents numTuples dataType
27 f00 f01 ... f0(numComponents-1)
28 f10 f11 ... f1(numComponents-1)
29  $\vdots$ 
30 f(numTuples-1)0 f(numTuples-1)1 ... f(numTuples-1)(numComponents-1)
31
32 arrayName1 numComponents numTuples dataType
33 f00 f01 ... f0(numComponents-1)
34 f10 f11 ... f1(numComponents-1)
35  $\vdots$ 
36 f(numTuples-1)0 f(numTuples-1)1 ... f(numTuples-1)(numComponents-1)
37
38 arrayName(numArrays - 1) numComponents numTuples dataType
39 f00 f01 ... f0(numComponents-1)
40 f10 f11 ... f1(numComponents-1)
41  $\vdots$ 
42 f(numTuples-1)0 f(numTuples-1)1 ... f(numTuples-1)(numComponents-1)

```

REFERÊNCIAS BIBLIOGRÁFICAS

- AHRENS, J.; GEVECI, B.; LAW, C. 36 – ParaView: An End-User Tool for Large-Data Visualization. In: HANSEN, C. D.; JOHNSON, C. R. (Ed.). **Visualization Handbook**. Burlington: Butterworth-Heinemann, 2005. p. 717–731. ISBN 978-0-12-387582-2. Disponível em: <<<https://www.sciencedirect.com/science/article/pii/B9780123875822500381>>>. 2
- DEITERDING, R. **Parallel adaptive simulation of multi-dimensional detonation structures**. Tese (Doutorado) — Brandenburgische Technische Universität Cottbus, 2003. 2
- DEITERDING, R. Block-structured adaptive mesh refinement - theory, implementation and application. In: **ESAIM: Proceedings**. [S.l.: s.n.], 2011. 2
- DOMINGUES, M. O. **Desenvolvimento de modelagem multiescala para instabilidades locais não-lineares em Astrofísica e Geofísica Espacial**. 2016. FAPESP Auxílio à Pesquisa - Regular. Proc. 15/25624-2. 2
- _____. **Modelagem magneto-hidrodinâmica adaptativa multidimensional no ambiente computacional AMROC para aplicações espaciais**. 2019. CONFAP ; Newton Fund, com FAPESP como instituição parceira no Brasil; UK Academies. Proc. 18/03039-9. Disponível em: <<www.fapesp.br>>. 2
- DOMINGUES, M. O.; DEITERDING, R. **Advanced criteria for dynamic mesh adaption in computational space weather forecasting**. 2016. FAPESP/SPRINT - Projetos de pesquisa - Mobilidade. Proc. 16/50016-9. Disponível em: <<fapesp.br>>. 2
- FARGE, M. Choice of representation modes and color scales for visualization in computational fluid dynamics. In: _____. **Science and Art Symposium 2000: 3rd International Conference on Flow Interaction of Science and Art with Exhibition/Lectures on Interaction of Science & Art, 28.2 — 3.3 2000 at the ETH Zurich**. Dordrecht: Springer Netherlands, 2000. p. 91–110. ISBN 978-94-011-4177-2. Disponível em: <<https://doi.org/10.1007/978-94-011-4177-2_12>>. 30
- HANWELL, M. D.; MARTIN, K. M.; CHAUDHARY, A.; AVILA, L. S. The visualization toolkit (vtk): Rewriting the rendering code for modern graphics cards. **SoftwareX**, v. 1-2, p. 9–12, 2015. ISSN 2352-7110. Disponível em: <<<https://www.sciencedirect.com/science/article/pii/S2352711015000035>>>. 40
- JOBARD, B.; LEFER, W. Creating evenly-spaced streamlines of arbitrary density. In: LEFER, W.; GRAVE, M. (Ed.). **Visualization in Scientific Computing '97**. Vienna: Springer Vienna, 1997. p. 43–55. ISBN 978-3-7091-6876-9. 19
- Kitware, Inc. **The VTK User's Guide**. 2010. Disponível em: <<<https://kitware.github.io/vtk-examples/site/VTKFileFormats/>>>. Acesso em: 18 abr. 2023. 41

LIU, Z.; MOORHEAD, R.; GRONER, J. An advanced evenly-spaced streamline placement algorithm. **IEEE Transactions on Visualization and Computer Graphics**, v. 12, n. 5, p. 965–972, 2006. 19

LOPES, M. M. **Numerical methods applied to space magnetohydrodynamics for high performance computing**. Tese (Doutorado) — Instituto Nacional de Pesquisas Espaciais, 2019. Disponível em: <<<http://urlib.net/sid.inpe.br/mtc-m21c/2019/04.02.23.51>>>. 3

LOPES, M. M.; DEITERDING, R.; GOMES, A. K. F.; MENDES, O.; DOMINGUES, M. O. An ideal compressible magnetohydrodynamic solver with parallel block-structured adaptive mesh refinement. **Computers & Fluids**, v. 173, p. 293–298, 2018. 2

APÊNDICE A - LINKS ÚTEIS

Informações gerais do ParaView podem ser obtidas nos seguintes links:

Página oficial – <www.paraview.org>

Página da Wikipedia – <<https://en.wikipedia.org/wiki/ParaView>>

Mailing list – <paraview@paraview.org>

Outras referências de interesse como manuais, tutoriais e vídeos são apresentados na Tabela A.1.

Tabela A.1 - Links de Manuais/Tutoriais do Paraview

ParaView	< docs.paraview.org/en/latest/index.html >
Webinars	< www.paraview.org/tutorials/#webinars >
pvpython	< https://kitware.github.io/paraview-docs/latest/python >
SANDIA	< https://itk.org/Wiki/images/1/13/ParaViewTutorial52.pdf >
TACC	< http://hpcuniversity.org/vscse/petascale/downloads/ParaView_tutorial.pdf >
DKRZ	< https://docs.dkrz.de/doc/visualization/sw/paraview/index.html >
IDRIS	< www.idris.fr/jean-zay/pre-post/jean-zay-paraview-doc.html >
OSTI	< www.osti.gov/servlets/purl/1502715 >
ALCF	< https://docs.alcf.anl.gov/cooley/software-and-libraries/paraview-tutorial/ >
SOU	< https://cmg.soton.ac.uk/research/categories/visualisation-and-data-handling-software/paraview/ >
KITWARE (campos-vetoriais)	< www.kitware.com/evenly-spaced-streamlines-2d/ >
Cyprien-Rusu (CFD playlist)	< www.youtube.com/watch?v=1jml3j3BmUg >
Legendas (forma/precisão)	< www.youtube.com/watch?v=v4t2zvfl69I >
Vorticidade	< www.youtube.com/watch?v=F195Xcym2xw >
Busca/plot 1D	< www.youtube.com/watch?v=zBiVhGPHTq8&t=81s >

APÊNDICE B - ANACONDA (PYTHON)

Anaconda é uma ambiente virtual seguro para o uso do Python, em particular do `pvpython`. O site para obter o anaconda é www.anaconda.com/distribution. A instalação do Anaconda não afeta a versão nativa do Python (a do systema operacional). É sempre recomendável instalar a mais recente versão (neste momento Anaconda3-2020.07-Linux-x86_64.sh) e na conta do usuário (versão padrão que deve ser confirmada no *prompt* de instalação).

Seguem os passos necessários para a instalação:

- a) digite em um terminal `bash`: `Anaconda3-2020.07-Linux-x86_64.sh` para ativar a instalação do Anaconda
- b) carregue o novo `PATH environment variable` que é dado pelo instalador na seção do `shell` no arquivo `~/.bashrc`.
- c) digite em um terminal `bash`:

```
source ~/.bashrc
```

Com isso o `conda's base environment` fica ativado na inicialização, setar o `auto_activate_base` parâmetro para `false` (isto foi feito nesta instalação. É necessário reinicializar a `home session`) depois digite

```
conda config --set auto_activate_base false
```

- d) para verificar a instalação

```
conda activate python3
```

ocorrerá uma linha com Anaconda ...

digite então no mesmo terminal `bash`:

```
conda update conda
```

```
conda update anaconda
```

- e) para adicionar pacotes de interesse na instalação

```
conda install package-of-interest
```

Exemplo:

```
conda install -c anaconda scikit-image
```

```
conda config --add channels conda-forge
```

f) para desinstalar pacotes que não mais são necessários:

```
conda uninstall package-of-interest
```

g) para criar um `virtual environment` para desenvolvimentos

```
conda create -name myenv
```

```
source activate myenv
```

Após o uso do ambiente:

```
source deactivate
```

ou se o ambiente não é mais necessário:

```
conda remove myenv -all
```

Exemplo:

```
conda create -name AnyFantasyDescriptiveName python=3.7.4  
pandas=1.0.0 numpy=1.16.6 matplotlib=3.1.2 scipy=1.3.2  
seaborn=0.9.0
```

Para ativar/desativar o ambiente

```
conda activate AnyFantasyDescriptiveName
```

ou

```
conda deactivate
```

h) Para mais informações sobre o Anaconda

```
conda info -envs
```

É sempre útil verificar a versão do Python. Digite no terminal `bash`:

```
python -version
```

ou para verificar pacotes específicos

```
conda search name-of-package ou conda list ou conda list namePackage
```

Para rodar o python usando diferentes recursos (ou *legacy*):

```
python3 name-of-script
```

No Jupiter (que é um ambiente iterativo via webpage):

```
jupyter notebook
```

No **Spyder** (Scientific Python Development Environment, as an IDE):

spyder

Para desinstalar o Anaconda (do Ubuntu system) são necessários quatro passos:

a) digite em um terminal **bash**:

```
rm -rf /anaconda3
```

b) edite o **PATH environment variable** no **/.bashrc** file and e remova o diretório

```
~/.bashrc
```

```
# added by Anaconda3 installer
```

```
export PATH="/home/linuxize/anaconda3/bin:$PATH"
```

c) remova arquivos ocultos

```
rm -rf ~/.condarc ~/.conda ~/.continuum
```

PUBLICAÇÕES TÉCNICO-CIENTÍFICAS EDITADAS PELO INPE

Teses e Dissertações (TDI)

Teses e Dissertações apresentadas nos Cursos de Pós-Graduação do INPE.

Manuais Técnicos (MAN)

São publicações de caráter técnico que incluem normas, procedimentos, instruções e orientações.

Notas Técnico-Científicas (NTC)

Incluem resultados preliminares de pesquisa, descrição de equipamentos, descrição e ou documentação de programas de computador, descrição de sistemas e experimentos, apresentação de testes, dados, atlas, e documentação de projetos de engenharia.

Relatórios de Pesquisa (RPQ)

Reportam resultados ou progressos de pesquisas tanto de natureza técnica quanto científica, cujo nível seja compatível com o de uma publicação em periódico nacional ou internacional.

Propostas e Relatórios de Projetos (PRP)

São propostas de projetos técnico-científicos e relatórios de acompanhamento de projetos, atividades e convênios.

Publicações Didáticas (PUD)

Incluem apostilas, notas de aula e manuais didáticos.

Publicações Seriadas

São os seriados técnico-científicos: boletins, periódicos, anuários e anais de eventos (simpósios e congressos). Constam destas publicações o Internacional Standard Serial Number (ISSN), que é um código único e definitivo para identificação de títulos de seriados.

Programas de Computador (PDC)

São a sequência de instruções ou códigos, expressos em uma linguagem de programação compilada ou interpretada, a ser executada por um computador para alcançar um determinado objetivo. Aceitam-se tanto programas fonte quanto os executáveis.

Pré-publicações (PRE)

Todos os artigos publicados em periódicos, anais e como capítulos de livros.