Crysttian Arantes Paixão

Grace e Shell Script: ferramentas para exploração de dados

Monografia de Pós-Graduação "*Lato Sensu*" apresentada ao Departamento de Ciência da Computação para obtenção do título de Especialista em "Administração em Redes Linux"

Orientador
Prof. Joaquim Quinteiro Uchôa

Co-Orientador
Profa. Iraziet da Cunha Charret

Lavras
Minas Gerais - Brasil
2007

Crysttian Arantes Paixão

Grace e Shell Script: ferramentas para exploração de dados

Monografia de Pós-Graduação "*Lato Sensu*" apresentada ao Departamento de Ciência da Computação para obtenção do título de Especialista em "Administração em Redes Linux"

Aprovada em *03 de Dezembro de 2010*

Prof. Sanderson L. Gonzaga de Oliveira

Prof. Joaquim Quinteiro Uchôa
(Orientador)

Profa. Iraziet da Cunha Charret
(Co-Orientador)

Lavras
Minas Gerais - Brasil
2007

Dedico esta monografia à minha Mãe, Cássia, aos meus irmãos, Richard e Emanuelle, aos meus avós, Maria Isabel e Gabriel e ao meu Tio João.

Agradecimentos

Uma vitória é constituída por muitas etapas e, em cada uma delas, é possível contar com a colaboração, apoio, incentivo, compreensão, torcida e expectativa de várias pessoas que ajudam a fazer a diferença. Assim agradeço a muitas pessoas que ajudaram a fazer a diferença. Aos mestres sem dúvida dedico grande parte de meu sucesso. Ao grupo de física da UFLA do qual já participo há muitos anos e que sempre me auxiliou nas dúvidas e nas colaborações com o projeto. Aos colegas de especialização com os quais convivi durante a fase de estudos. Aos meus orientadores, Joaquim Quinteiro Uchôa e Iraziet da Cunha Charret, pela ajuda, empenho e paciência durante o desenvolvimento deste trabalho. Nessa caminhada, perdemos pessoas que foram e são fundamentais para o nosso sucesso, mas que por motivos que somente Deus pode explicar não estão mais entre nós. Em especial, ao meu Avó Gabriel, a quem durante toda a minha vida, considerei como um pai. À minha avó Maria Isabel (Vó Tota) e Tio João que, quando tinha uma folga, estava em sua casa para descansar e revigorar para enfrentar os estudos. A minha mãe, Cássia, que lutou por toda uma vida para poder dar a mim e meus irmãos oportunidade de alcançarmos o sucesso. Hoje posso dizer que todo o seu empenho foi de grande importância para o meu sucesso, à você, meu amor e gratidão eternos. À minha família, que as vezes, mesmo sem saber o que eu estudava, e nem entender o porquê de tantas horas de esforço, dedicação e em muitas horas de leitura solitária, nunca negaram apoio e incentivo. Somente pelo fato de ter como a que eu tenho, posso afirmar que sou um vencedor, meu amor e gratidão eternos a vocês. Agradeço à UFLA (Univesidade Federal de Lavras) pela bolsa concedida para auxiliar no desenvolvimento desse projeto, pela estrutura e oportunidade. À banca, composta por Sanderson L. Gonzaga de Oliveira e Iraziet da cunha Charret que aceitaram o convite que lhes foi feito e, dessa forma, e colaboraram para a conclusão deste projeto. Meus sinceros agradecimentos a todos que os que participaram desta fase importante da minha vida. A Deus, pela inspiração, pela força, pelos momentos concedidos para fazer com que essa vitória pudesse ser valorizada, eu agradeço eternamente.

Sumário

Lista de Figuras

Lista de Tabelas

Resumo

Neste trabalho é apresentado o *software* Grace em conjunto com o Shell Script, utilizados como ferramenta para análise de arquivos de dados. Foi desenvolvido um *script* que apresenta os comandos utilizados, detalhando as funcionalidades envolvidas. Foi possível otimizar o tempo gasto com a produção de gráficos a partir de uma grande quantidade de arquivos de dados. Devido à automatização do processo, reduzem-se bastante os erros cometidos na realização das tarefas, usando *Shell Script*. Como resultado, obtem-se gráficos com alta qualidade devido à utilização do *software* Grace, com grande redução de tempo de confecção e significativa diminuição de erros.

Palavras-Chave: Grace; *Shell*; *script*; ferramenta; análise; dados.

Capítulo 1

Introdução

Neste trabalho vamos apresentar uma solução baseada no uso do software Grace[1] e Shell Script[2] para a confecção de gráficos. Durante o desenvolvimento de uma pesquisa na Universidade Federal de Lavra - UFLA (PAIXÃO, 2007), eram geradas grandes quantidades de arquivos de dados. Esses arquivos de dados eram manipulados e depois plotados através do software Grace para realizar um estudo mais apurado. Os arquivos eram gerados através de simulações computacionais em que os dados gerados estavam relacionados a determinados valores de parâmetros.

Devido à grande quantidade de arquivos a serem manipulados usando os recursos do software Grace, demandava-se tempo e inúmeras manipulações, que na maioria das vezes eram repetitivas. Basicamente a criação de gráficos era feita selecionando o grupo de arquivos de dados a serem manipulados. Utilizava-se o Grace para abrir os arquivos e plotá-los para realizar as análises.

Os comandos do Grace usados eram sempre os mesmos e sempre em uma determinada ordem para se obter os gráficos desejados. As configurações da simulação eram constantemente alteradas para realizar uma análise completa do problema estudado logo para cada mudança, todo o processo de confecção dos gráficos devia ser repetido.

Analisando o manual do Grace foi possível verificar que é possível utilizá-lo através de linha de comando. Linha de comando é um meio em que é possível utilizar um software instalado no computador interagindo de forma textual com ele. Antes de se ter as interfaces gráficas, tudo era feito através de linhas de co-

[1] http://plasma-gate.weizmann.ac.il/Grace/
[2] http://www.freeos.com/guides/lsst/

mando. Na forma textual são usadas palavras que o software reconhece, neste caso os comandos a serem executados. Por exemplo, no Linux é possível realizar diferentes atividades usando a linha de comando, neste caso utiliza-se o Shell que é responsável em fazer a interface entre o usuário e o software. No Grace é possível interagir com o software de forma textual e direta, sem a necessidade de utilizar a sua interface gráfica.

O Grace permite a interação de forma direta através da passagem de comandos (palavras) diretamente ao software indicando as ações ou através de um arquivos com as ações a serem feitas. Essas ações são representadas através de comandos preeestabelecidos que o software reconhece. Cada atividade feita na interface gráfica corresponde a um comando específico e que está disponível, na maioria das vezes, para ser utilizado através de linha de comando em interação com o software. Logo é possível fazer todo o procedimento para se confeccionar um gráfico através de comandos pré-estabelecidos e que quando repassados ao sofware em uma determinada ordem, gera os mesmos resultados alcançados usando a interface gráfica. Assim, torna-se possível utilizar o Grace, estabelecendo sempre os comandos e as respectivas ordens de execução para gerar os gráficos desejados. Podemos considerar como uma forma de padronizar o processo de criação de gráficos.

Para automatizar ainda mais as tarefas de confecção dos gráficos, também foi utilizado o Shell Script que potencializa a utilização de software através de linha de comandos.

Neste trabalho vamos apresentar a utilização dessas duas ferramentas, Grace e Shell Script em conjunto, para construir gráficos de alta qualidade e de forma automatizada. É realizado uma descrição do uso do software Grace, no capítulo 2.1 e do Shell Script, no capítulo 2.2. No capítulo 3 é descrito o problema que motivou a utilização das ferramentas apresentadas em conjunto. No capítulo 4.1 é descrito um dos vários modos para utilizar essas duas ferramentas em conjunto para construir diversos gráficos. No último capítulo, apresentamos nossas conclusões e considerações finais. No anexo consta o código do script criado para realizar a confecção dos gráficos.

Capítulo 2

Conceitos Básicos

2.1 Grace

O Grace é um *software* que segue a filosofia WYSIWYG, que é o acrônimo da expressão em inglês "What You See Is What You Get". A medida que as atividades são realizadas no Grace, o gráfico é construído e o resultado dessas manipulações é exatamente o que se vê. Ele possui uma interface gráfica que possibilita ao usuário ir acessando os recursos disponíveis e desenvolvendo as atividades logo podemos considerar como uma vantagem apresentada pelo *software*, pois o resultado das manipulações já ficam visíveis. A sua interface pode ser vista na figura 2.1. Apresenta outra vantagem, sua simplicidade com relação a disponibilidade dos recursos para o usuário, em que algumas opções são quase que intuitivas.

Este *software* também apresenta um controle preciso para as características dos gráficos, além de possibilidar a geração de gráficos de alta qualidade que podem ser exportados para diferentes formatos, por exemplo, PDF, PS, EPS, JPG dentre outros. Apresenta também várias opções referentes a cores, fontes, tipos de linhas, dentre as inúmeras opções de formatação final que podem ser aplicadas aos gráficos.

Além da confecção de gráficos, é um aplicativo que possibilita outras análises dos dados, como por exemplo, a extração de características, ajuste de curvas, construção de histogramas, interpolação de curvas, cálculo de correlação, convolução e covariação, programação de eventos específicos e outros.

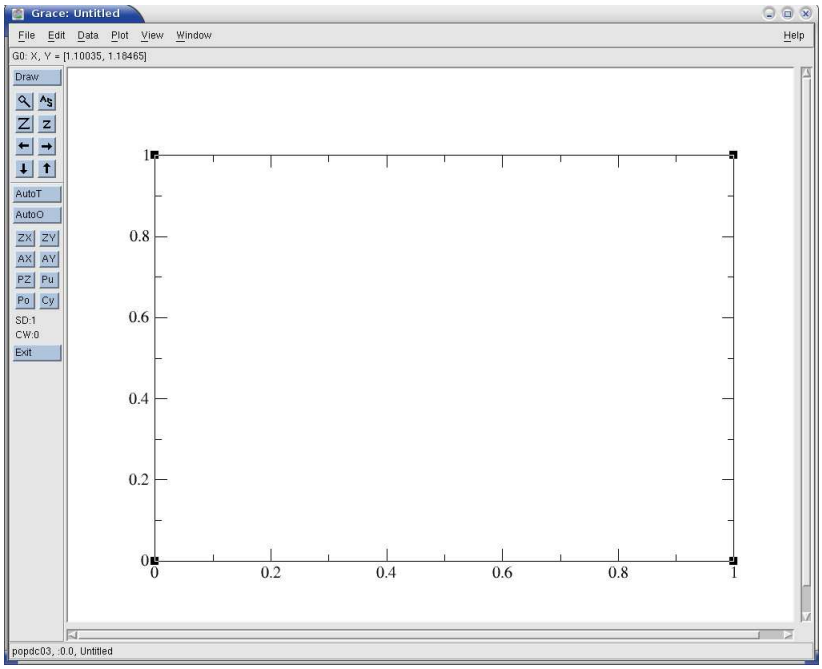

Figura 2.1: Interface do Grace

Todas as características apresentadas até esse momento descrevem que para se construir um gráfico usando o Grace é necessário que se manipule os seus recursos até que o resultado esperado seja alcançado.

Outra característica importante com relação ao Grace é a possibilidade de executar uma série de comandos, as mesmas utilizadas na interface, através de um arquivo de comandos em lote, o *batch*. Um *batch* contem os comandos, os mesmos que são utilizados na interface gráfica, em uma sequência. Por exemplo, para se construir um gráfico simples, basta apenas importar os dados. No Grace para importar os dados, basta selecioná-los para serem importados, para isso é necessário acessar o menu *Data*, *Transformations* e utilizar a opção *Import*, ASCII e o arquivo de dados será importado e plotado em um gráfico. O mesmo pode ser feito com um *batch*, bastando apenas inserir os comandos necessários para a importação dos dados. Os comandos utilizados são descritos na seção 4.1.

A possibilidade de utilizar o Grace através de arquivos *batch* de forma automatizada, sem a necessidade de se intergir com o *software* foi um dos motivos de sua escolha. Existem outros programas que criam gráficos com alta qualidade, por exemplo o Gnuplot[1], mas este *software* não permite utilizá-lo através de arquivos *batch* de forma automatizada sem ter a necessida de se interagir com ele.

Um outro exemplo seria a construção de um gráfico a partir de uma série de comandos. Caso seja necessário criar dezenas de gráficos, esses comandos devem ser executados novamente, sendo repetidos dezenas de vezes. Alguns desses comandos devem ajustar as escalas dos gráficos e os tamanhos das fontes dependendo da quantidade de manipulações que devem ser feitas, isso pode propiciar um erro. Já no caso do *batch*, como todo comando fica registrado e sempre na ordem a ser executado, isto faz com que as tarefas possam ser automatizadas, o que reduz as chances de ocorrer erros e garante que sempre os mesmos comandos sejam repetidos na ordem desejada.

No desenvolvimento do trabalho de pesquisa já mencionado, devido ao número elevado de comandos e gráficos a serem criados, este recurso do Grace foi a solução para a padronização dos processos de construção dos gráficos. Apesar de ser um recurso avançado para a utilização do aplicativo, o aprendizado é simples. A maioria dos recursos da interface pode ser associado a um comando específico para ser executado em um *batch*.

[1] http://www.gnuplot.info/

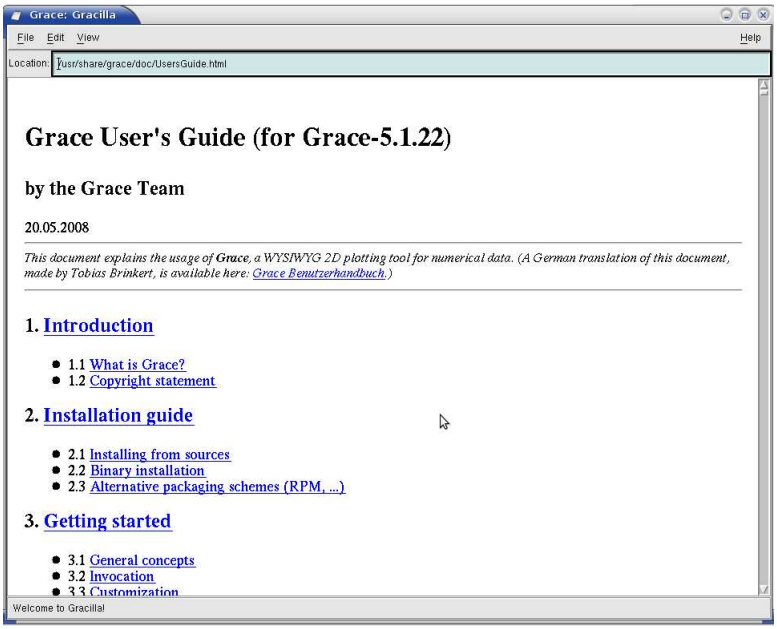

Figura 2.2: Menu de Ajuda do Grace

O Grace possui um menu de ajuda, que pode ser visto na Figura 2.2, que fornece instruções de como utilizar a interface gráfica e disponibiliza junto a cada opção o respectivo comando que pode ser utilizado para a construção do *batch*.

O Grace está disponível para ser instalado em sistemas Linux e Windows porém, em sistemas Windows, algumas de suas funcionalidades não estão disponíveis. Existem alguns projetos que utilizaram o Grace como base e outros que surgiram a partir de seus *forks*. O idioma padrão é o Inglês mas também está disponível em outros idiomas. Também é possível utilizá-lo com linguagens de programação, como C e Fortran. Estas informação podem ser obtidas no site do desenvolvedor do Grace[2].

2.2 Shell Script

Como já foi mencionado, antes de existir a interface gráfica para utilização dos *software*, estes eram utilizados através de linha de comando. As ações que o

[2]http://plasma-gate.weizmann.ac.il/Grace/

software tinha que executar eram passadas através de modo texto. Para que essa atividade fosse executada era necessário ter um meio, um interpretador, que permitisse a interação entre usuário e *software*. Esse interpretador devia receber os comandos digitados e fazer com que o *software* fosse executado com a opção escolhida. Um interpretador é um *software* e todos os sistemas operacionais utilizam de um para executar as mais diferentes tarefas. Um exemplo de interpretador é um terminal de comandos. Em sistemas baseado no Unix esses terminais recebem o nome de Shell. Com um Shell é possível interagir com o sistema operacional e com os *software* que estiverem disponíveis. O Shell faz a ligação entre o usuário, o sistema e seus recursos.

Para utilizar o sistema através do Shell, como mencionado, são necessárias algumas palavras pré-estabelecidas, neste caso essas palavras fazem referências ao *software* ou parâmetros de *software*, mas que na maioria da vezes recebem a denominação de comandos. Então para executar um *software* é necessário digitar o comando que faz com que ele seja executado na maioria das vezes os comandos possuem o mesmo nome do *software*. Por exemplo, para abrir o Grace atraveś de um Shell, basta digitar o comando *xmgrace* no terminal. Um exemplo de terminal em execução pode ser visto na figura 2.3.

Quando um terminal é aberto, como pode ser visto na figura 2.3, você esta executando um *shell* que na maioria das vez é o Bash[3]. O Bash é o *shell* desenvolvido para o Unix sobre a licença da GNU[4]. Vale ressaltar que os comandos do sistema Linux também podem ser executados via terminal e podem ser utilizdos em *Shell Script*. Um resumo dos comandos podem ser encontrados em Macan (2004) e uma descrição mais detalhada no projeto Foca Linux[5].

Existem inúmeros comandos que podem ser utilizados através de um Shell, por exemplo, no Bash, uma coletânea dos comandos podem ser encontrados no site[6] com os respectivos parâmetros. Esses parâmetros são opções extras para se utilizar o *software* de maneira específica. Por exemplo, para ver o conteúdo de um arquivo através do terminal, basta digitar *cat* seguido pelo nome do arquivo logo o conteúdo do arquivo será impresso na tela. Para verificar o horário e a data do sistema basta digitar *date*, podemos também visualizar o calendário com o comando *cal*. Para imprimir uma mensagem na tela, basta usar o comando *echo* seguido de seu parâmetro, neste caso a mensagem a ser impressa. Existem inúmeros comandos, cada um com uma finalidade e com diferentes opções de execução. Veja que

[3] http://www.gnu.org/software/bash/bash.html
[4] http://www.gnu.org/licenses/gpl.html
[5] http://focalinux.cipsga.org.br/
[6] http://onlamp.com/linux/cmd/

com esses comandos é possível manipular, criar e apagar arquivos presentes no sistema.

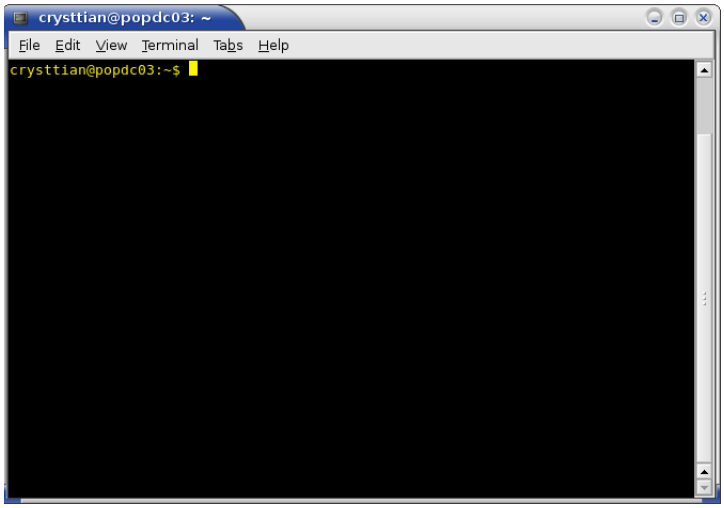

Figura 2.3: Exemplo de um terminal em execução dentro do Linux

Assim, para que um administrador ou usuário normal possa executar as suas tarefas habituais através de um terminal, torna-se necessário conhecer e executar diferentes comandos, quase sempre em uma mesma sequência. Dependendo da tarefa a ser executada e por se tratar de uma atividade repetitiva, ela acaba por ser monótona. Para fazer com que essas tarefas sempre sejam executadas de maneira correta e o mais rápido possível são utilizados *scripts*. Esses *scripts* viabilizam a execução das tarefas, diminuindo os possíveis erros e automatizando as tarefas a serem executadas.

Um *script* pode conter os comandos disponíveis no sistema. Logo, ao invés de digitar todos os comandos, basta simplesmente executar o *script* que contenha os comandos na ordem que devem ser executados, dentro do *shell* adequado. Durante a execução do *script*, os comandos vão sendo interpretados um a um pelo *shell*, e vão sendo executados. As tarefas rotineiras e que demandavam tempo e atenção, dependendo das atividades que devem ser executadas, com a utilização de *Shell Script* tornam-se simples.

Devido a essa facilidade de agendar os comandos para serem executados por *Shell Script*, tornou-se necessário para os administradores aprenderem a criar algoritmos para atender as mais variadas necessidades (JARBAS, 2008). A pro-

gramação em *Shell Script* propicia o desenvolvimento de *script*s que permitem praticamente executar todas as tarefas com comandos do *shell* e do sistema para os mais variados fins. A automatização de tarefas é uma das grandes vantagens em se utilizar *shell script*. Existem muitas referêncas para a programação em *Shell Script* disponíveis na internet[7] e diversos livros que tratam do assunto destacando Saade (2001), Morimoto (2006) e Jargas (2008).

Para automatizar as tarefas a serem feitas, priorizamos a utilização de *Shell Script* em que os comandos do sistema foram utilizados em conjunto com as funcionalidades do Grace. Em suma, com o *Shell Script*, os arquivos de dados e opções a serem passadas para o Grace eram organizadas e o *script* ficava responsável por executar o Grace através da linha de comando passando os valores dos parâmetros necessários para contruir os gráficos desejados. Logo todo o processo em que o Grace era utilizado através de sua interface gráfica ficou resumido à execução de um *script*.

Vamos apresentar o problema e a solução criada utilizando o Grace e o *Shell Script* em conjunto.

[7] http://onlamp.com/linux/cmd/

Capítulo 3

Descrição do Problema

Como já mencionamos, no desenvolvimento da pesquisa (PAIXÃO, 2007) eram gerados grandes quantidades de arquivos e para cada determinado conjunto desses arquivos era necessário construir um gráfico para que os dados e os parâmetros da simulação fossem analisados. Tal tarefa era muito dispendiosa, pois demandava tempo e o número de repetições tornava a atividade monótona e suscetível a erro. Para minimizar esses possíveis erros e automatizar a geração dos gráficos optou-se em adotar o uso do *Shell Script* em conjunto com o Grace.

Durante a pesquisa, os arquivos eram criados por simulações de um determinado processo de avaliação de imagens com *laser*, denominado Biospeckle. Na simulação existiam diferentes parâmetros, os mais importantes e que influenciavam o número de arquivos a serem gerados eram os tamanhos de passo (60 valores), número de fontes (3 valores) portanto, temos até agora ao todo 180 arquivos a serem gerados na simulação. Os arquivos possuem 50.000 dados numéricos dispostos em uma coluna. Basicamente havia 3 grupos de arquivos, cada grupo para um determinado número de fontes, logo tínhamos 3 grupos de dados, cada um com 60 arquivos para serem manipulados até ser gerado o gráfico desejado.

O processo para criar um gráfico é:

- Importar arquivo por arquivo, cada um referente a um tamanho de passo, sendo a ordem de importação deve estar de acordo com a ordem crescente do tamanho de passo;
- Com os dados importados é necessário criar um histograma para cada um. Para se construir o histograma é necessário encontrar o maior valor entre os

dados de todas as séries e posteriormente fazer um histograma. Para criar um histograma é necessário informar a faixa de valores que o histograma será construído, neste caso a faixa vai de 0 até o maior valor encontrado;

- Com os histogramas já criados, é necessário calcular o HMW, uma medida de largura do histograma. Após efetuar o cálculo é necessário exportar o arquivo com os valores de HMW para manipulá-lo posteriormente;
- Com o arquivo de valores de HMW, deve-se associar cada valor do HMW com o respectivo valor de tamanho de passo que o gerou os valores devem ser editados manualmente;
- Com o arquivo criado com os valores de HMW e o tamanho do passo, plota-se o gráfico do HMW em função do tamanho do passo;
- No gráfico plotado as escalas dos eixos são ajustadas para a escala logarítmica;
- Os ajustes finais devem ser feitos e o gráfico é exportado no formado de imagem EPS;

No Grace, esse processo é gerado executando os comandos consideramos que o Grace já esteja aberto:

- Para importar os dados, deve-se acessar o menu Data, Import, ASCII, a tela para selecionar os arquivos a serem importados é aberta, os 60 arquivos desejados devem ser selecionados um a um, na ordem crescente do tamanho do passo que gerou os arquivos, para serem importados;
- O maior valor presente nos arquivos de dados pode ser obtido acessando o arquivo criado pelo programa da simulação, que já calcula esse valor;
- Para criar os histogramas, acesse o menu Data, Transformation, Histogram. Insira o menor valor, neste caso 0 e o maior calculado na simulação.
- Para calcular os valores de HMW, acesse o menu Data, Feature extration, selecione a opção Half Maximal Witdh - HMW, será gerado um arquivo com os valores de HMW para cada histograma calculado;
- Para exportar o arquivo com os valores de HMW, selecione o menu Data, Export, ASCII, selecione o arquivo com os valores de HMW, digite o nome do arquivo a ser exportado e o exporte;

- Com o arquivo exportado, abra-o em um editor de texto, e para cada valor de HMW, insira o valor do tamanho do passo correspondente e salve o arquivo;

- Importe o arquivo criado com os valores de HMW e o tamanho do passso, para colocar os eixos do gráfico na escada logaritímica, acesse o menu Plot, Axes Properties e altere a escala dos eixos X e Y para a escala logarítmica, também coloque os nomes dos eixos e faça os ajustes finais caso seja necessário. Por exemplo, para colocar uma legenda nos dados plotados, dê um clique duplo sobre a curva desejada e insira o nome que deve aparecer na legenda no campo String na opção Legend;

- Salve o gráfico criado, agora ele será exportado em um formato para ser utilizado como figura. Para isso acesse o menu File, Print Setup, e configure o nome do arquivo a ser exportado e o tipo de arquivo que será gerado, existem diferentes formatos como EPS, JPG, dentre outros. Para exportar o arquivo, selecione File e depois a opção Print, pronto o gráfico será impresso para o formato desejado. Basta agora fechar o Grace;

Todo o procedimento era feito para os três grupos de arquivos, um por vez, sendo repetidos 8 vezes, gerando 24 gráficos desejados no total.

Para ilustrar esse processo, foi criado a Figura 3.1, nela podemos verificar as etapas para a geração dos gráficos para cada conjunto de dados.

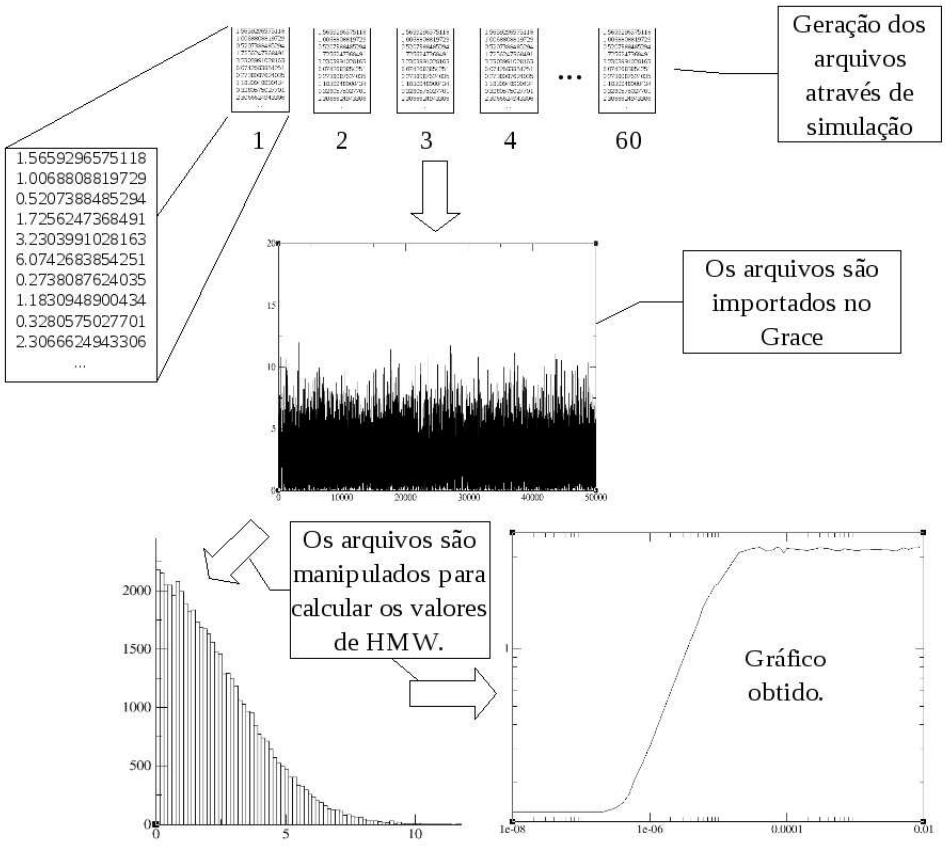

Figura 3.1: Processo de Criação dos Gráficos através da importação e manipulação dos arquivos no Grace.

Capítulo 4

Solução Adotada

4.1 A solução criada

Todo o procedimento descrito anteriormente, quando feito manualmente demanda muito tempo. Um dos agravantes está no fato de que quando se desejava alterar algum valor de parâmetro do modelo, todo o processo devia ser repetido para gerar novamente o gráfico. Para otimizar essas atividades foram criadas rotinas em *Shell Script* que utilizam os recursos do sistema em conjunto com as funcionalidades do Grace. Agora vamos descrever passo a passo o *script* e os procedimentos adotados. Vale ressaltar que os comandos que serão descritos estão dentro de um único *script*, que quando executado, retorna um gráfico ao final do processo com a imagem gerada a partir dele no formato desejado. Vamos fazer todo o procedimento passo a passo para explicá-lo detalhadamente.

Como mencionamos anteriormente, tinhamos três grupos de arquivos, cada um com sessenta arquivos ao todo. Vamos considerar que a simulação já gerou os arquivos e agora temos que manipulá-los. O primeiro passo é importar todos esses arquivos dos três conjuntos, para o Grace. Mas a importação dos dados é feita grupo por grupo. O *script* importa um grupo de arquivos por vez.

A primeira parte do *script* está relacionado a importar todos os arquivos para o Grace. O comando em *shell* que agrupa os arquivos e os importa para o Grace e todos os outros comandos utilizados estão descritos no código 4.1.1 e 4.1.2.

Listagem 4.1: Detalhes do *script* - Inicializando variáveis.

```
1 arquivos=""
2 contador=0
```

Na linha 1 do código da Listagem 4.1 é criada a variável *arquivos* que irá armazenar os nomes dos arquivos a serem importados (os sessenta arquivos, um para cada tamanho de passo). Na linha 2 é criada uma variável *contador* para ser usada durante a execução do *script*.

Listagem 4.2: Detalhes do *script* - Obtendo um valor de um arquivo e atribuindo a uma variável.

```
1 size=‘cat data.dat‘
```

No código da Listagem 4.2 é obtido o maior valor dos dados a serem analisados. Este valor é armazenado na simulação no arquivo *data.dat*. O comando *cat* extrai o valor do arquivo, o uso dos apóstrofos, diz que primeiro o valor é extraído do arquivo e posteriormente é atribuído a variável tamanho.

Listagem 4.3: Detalhes do *script* - Criando os aruqivos temporários.

```
1 touch comando.bat
2 touch comandoHistograma.bat
3 touch comandoApagar.bat
4 touch tamanhoPasso.bat
```

Nas linhas do código da Listagem 4.3, os arquivos que serão necessários são criados com o comandos *touch*. O comando *touch* cria um arquivo com o nome passado como parâmetro, sendo um arquivo com conteúdo vazio.

Listagem 4.4: Detalhes do *script* - Criando o arquivo de comandos do Grace.

```
1  for potencia in ‘seq 8 -1 3‘
2  do
3          for base in ‘seq 1 9‘
4          do
5                  nome="P1.D-"$potencia"dados"$base".dat"
6                  arquivo=$arquivo" "$nome
7                  echo "HISTOGRAM(S$contador,MESH(0,$tamanho,100),
                       OFF,OFF)" >> comandoHistograma.bat
8                  echo "KILL S\$contador" >> comandoApagar.bat
9                  echo $base"E-"$potencia >> tamanhoPasso.bat
10                 contador=‘echo \$contador+1 | bc‘
11         done
12 done
```

No código da Listagem 4.4, será criada uma lista com os nomes dos arquivos, um para cada tamanho de passo, para serem importados. Na linha 1 do código 4.4, o comando *for*, cria um sequência de dados de 8 até 3, em que o contador

é acrescido do valor -1, neste caso uma sequência decrescente, pois o tamanho do passo está na faixa de 1×10^{-8} até 1×10^{-3}. Esses valores são atribuídos à variável "potencia", veja que os os valores criados correspondem as potências do intervalo do tamando do passo. Na linha 3 é criado os valores para base com valores de 1 até 9, eles correspondem aos valores inteiros do tamanho do passo. Veja a representação dos valores calculados em cada uma dessas variáveis na tabela 4.1 (alguns valores foram omitidos):

Tamanho do Passo	Base	Potência
1×10^{-8}	1	-8
2×10^{-8}	2	-8
3×10^{-8}	3	-8
...	...	...
1×10^{-5}	1	-5
...	...	...
1×10^{-3}	1	-3
...	...	...
9×10^{-3}	9	-3

Tabela 4.1: Descrição dos valores

Os dois comandos *for* criam a sequência do tamanho do passo utilizado para gerar os nomes dos arquivos a serem importados. Esses valores são armazenados na linha 9 do código 4.4 no arquivo tamanho do passo, para serem utilizados posteriormente.

No código da Listagem 4.4, na quinta linha são criados os nomes dos arquivos de dados a serem importados. Neste caso, o comando para criar os nomes é *nome="P1.D-"$potencia"dados "$base".dat"* cria na ordem crescente do tamanho do passos, os nomes dos arquivos que devem ser importados. A ordem de importação já é criada automaticamente com as interações dos comandos *for*. Por exemplo, quando a variável "potencia" vale 8 e a variável "base" vale 3, o nome do arquivo a ser importado é *"P1.D-8dados3.dat"*, este nome se refere ao arquivo que foi criado com o tamanho de passo igual a 3×10^{-8} durante a simulação.

Na sexta linha do código da Listagem 4.4 são concatenados os nomes dos arquivos em uma única variável, após os 3 primeiros processos de interações dos comandos *for* a variável "arquivo" possui o conteúdo "*P1.D-8dados1.dat P1.D-8dados2.dat P1.D-8dados3.dat*". Ao final do processo, ela irá armazenar todos os nomes dos arquivos a serem importados. Em *Shell Script* o valor de uma variável é acessado colocando o "$" à frente do nome logo nessa linha, podemos ver que o

que é feito, a variável “arquivo” recebe o seu conteúdo concatenado com o valor da variável “nome”.

Na sétima linha do código da Listagem 4.4 é inserido no arquivo *comandoHistograma.bat* os comandos do Grace para a construção dos histogramas. Como já foi mencionado, cada arquivo que é importado para dentro do Grace, corresponde a um tamanho de passo específico. O comando *echo* imprime o que é passado como parâmetro, neste caso o comando do Grace que gera o histograma. Se fosse usado sem a opção “>>”, o comando seria impresso na tela, mas o comando “>>” redireciona a saída do comando *echo*, o que foi passado como parâmetro, para a última linha do arquivo *comandoHistograma.bat*. Vamos supor que estamos na primeira interação dos *for* e que o conteúdo da variável “tamanho” seja 125, logo o texto que é escrito dentro do arquivo é “*HISTOGRAM(S0,MESH(0,**125**,100),OFF,OFF)*”.

Vamos analisar o comando escrito no arquivo. O comando para criar histogramas no Grace é *HISTOGRAM*, *S0* indica que o arquivo que deverá ser usado para gerar o histograma está armazenado na variável *S0* dentro do Grace. Quando o Grace importa um arquivo, este é armazenado dentro de variáveis cujos nomes começam com a letra “S”, seguido da ordem em que foram importados, neste caso o 0, por se tratar do primeiro arquivo importado. Neste caso a variável “contador” tem a função de realizar a contagem correta dos arquivos que vão sendo importados, na linha 2 do código da Listagem 4.4 ele é inicializado com o valor 0 e a medida que os comandos *for* vão sendo executados ela é acrescida de 1, linha 10 do código da Listagem 4.4. O procedimento para incrementar um valor de uma variável no *shell* é um pouco complicado. Para que isso seja feito é necessário fazer com que o comando *echo* passe um texto que contenha o valor da variável “contador” com o incremento para a calculadora do *shell* utilizado, neste caso a calculadora é executada com o comando *bc*. Como esses comandos estão dentro de apóstofros, primeiro a mensagem é repassada a calculadora, que logo em seguida realiza a operação indicada no texto e posteriormente o valor calculado é atribuído a variável “contador”. Durante a execução do *script* a variável “contador” assume os valores de 0 até 59, que dizem respeito aos 60 arquivos a serem importados. O comando “*MESH*” é inserido para criar o intervalo que o histograma será criado, este intervalo, neste caso é de 0 até 125 e será dividido em 100 partes. Os valores dos dados importados vão sendo inseridos dentro desses intervalos para serem contabilizados. Os comandos “*OFF*” desativam alguns recursos que não serão utilizados no comando “*HISTOGRAM*”, para demais esclarecimentos consulte o manual do Grace.

Depois que os histogramas forem criados, a série de dados original pode ser apagada, para isso, na linha 8 do código da Listagem 4.4 é criado o arquivo *co-*

mandoApagar.bat que irá conter os comandos do Grace para remover as séries de dados. Por exemplo, sendo a primeira execução dos *for*, o valor que o comando *echo* imprimi é “KILL S0” e este valor a ser impresso é redirecionado para dentro do arquivo *comandoApagar.bat*. Veja que mais uma vez é usado a variável “contador”. Este comando quando executado no Grace, remove o arquivo de dados que está armazenado na variável *S0* dentro do ambiente do Grace. Ao final das interações vamos ter um arquivo com os comandos para apagar os 60 arquivos importados.

Listagem 4.5: Detalhes do *script* - Agrupando arquivos de comandos.

```
1 cat comandoHistograma.bat >> comando.bat
2 cat comandoApagar.bat >> comando.bat
```

Após as interações dos comandos *for* terminarem, o *script* segue normalmente. Nas linhas do código da Listagem 4.5, os arquivos “comandoHistograma.bat” e “comandoApagar.bat” são unidos para formar o arquivo “comando.bat”. Na linha 1, o conteúdo do arquivo “comandoHistograma.bat” é extraído com o comando *cat* e ao invés de ser impresso na tela é redirecionado com o comando “>>” para dentro do arquivo “comando.bat”. Na linha 2 do código da Listagem 4.5 o conteúdo do arquivo “comandoApagar.bat” é extraído e inserido depois da última linha do arquivo “comando.bat”. O formato do arquivo gerado fica igual ao do código da Listagem 4.6, note que nem todas as linhas foram exibidas (...) para poupar espaço.

Listagem 4.6: Arquivo com os comandos do Grace para importar os arquivos de dados. As linhas que estão representadas por “...” foram omitidas.

```
1  HISTOGRAM(S0,MESH(0,125,100),OFF,OFF)
2  HISTOGRAM(S1,MESH(0,125,100),OFF,OFF)
3  HISTOGRAM(S2,MESH(0,125,100),OFF,OFF)
4  HISTOGRAM(S3,MESH(0,125,100),OFF,OFF)
5  HISTOGRAM(S4,MESH(0,125,100),OFF,OFF)
6  ...
7  HISTOGRAM(S59,MESH(0,125,100),OFF,OFF)
8  KILL S0
9  KILL S1
10 KILL S2
11 KILL S3
12 KILL S4
13 ...
14 KILL S59
```

Agora já temos o arquivo com os comandos para gerar os histogramas e excluir os arquivos dados desnecessários.

Listagem 4.7: Detalhes do *script* - Executando o Grace via linha de comando com o arquivo *batch*.

```
1 xmgrace -nosafe $arquivo  -batch comando.bat -saveall grafico.agr
```

No código da Listagem 4.7, o Grace será executado com os comandos descritos no arquivo "comandos.bat" sobre os dados dos arquivos importados. O comando *"xmgrace -nosafe $arquivo -batch comando.bat -saveall grafico.agr"* faz com que o Grace seja executado, vamos detalhar o comando. O comando "xmgrace" é usado para executar o Grace via linha de comado, a opção "-nosafe" ativa a permissão de edição do gráfico a ser criado, "$arquivo" é a variável criada que indica os nomes dos arquivos a serem importados, "-batch comando.bat" indica para o Grace que é para ser executados sobre o conjunto de arquivos a serem importados os comandos que estão dentro do arquivo "comando.bat", a opção "-saveall grafico.agr" salva o arquivo criado com o comando "grafico.agr". Neste caso o Grace irá importar os dados, criar os histogramas, irá remover os dados importados que são desnecessários e irá salvar o gráfico criado.

Até este ponto já temos os histogramas criados. O Grace irá permanecer aberto para que os valores de HMW possam ser calculados. Para calcular os valores de HMW acesse o menu *Data*, *Features extrations*, selecione a opção *Half Maximal Witdh* - HMW, pronto, os valores de HMW para cada histograma serão calculados e salvos no Grace na variável *S0*. Lembre-se que *S0* armazenava o arquivo de dados, exatamente o primeiro que foi importado, mas o script apagou os arquivos que foram importados e que eram desnecessários, logo a variável *S0* está disponível para receber os dados. Feito isso, acesse o menu *File* e selecione a opção *Save* e depois a opção *Exit* para sair do Grace. Feito isso o *script* seguirá normalmente a execução.

Listagem 4.8: Detalhes do *script* - Escrevendo comandos em um arquivo.

```
1 echo "WRITE G0.S0 FILE \"hmw.dat\" " > comando.bat
2 echo "EXIT" >> comando.bat
```

No código da Listagem 4.8, é criado novamente outro arquivo de comandos para o Grace. Veja que agora não é usado o comando "$>>$" e sim o "$>$", o primeiro comando do *batch* insere um texto ao final do arquivo, respeitando o conteúdo do arquivo, já o comando "$>$" cria um novo arquivo, apagando o conteúdo do arquivo caso exista e inserindo um novo texto. Assim, nessa linha do *script*, o conteúdo do arquivo "comando.bat" é apagado e é inserido o texto "WRITE G0.S0 FILE "hmw.dat"", que corresponde a um comando do Grace, que pega o conteúdo armazenado na variável *S0* do gráfico *G0* (*G0* corresponde ao gráfico que aparece dentro do Grace, podemos ter vários gráficos, os demais são nomeados de *G1, G2*,

...) e gera um arquivo com os seus valores, neste caso o arquivo recebe o nome de "hmw.dat". Pronto, os valores de HMW calculados foram exportados.

Na linha 2 do código da Listagem 4.8, o comando "EXIT" é inserindo dentro do arquivo de comandos "comandos.bat", só que neste caso ele é adicionado, pois o comando utilizado foi o ">>". O comando "EXIT" irá fechar o Grace depois que os dados forem importados.

Listagem 4.9: Detalhes do *script* - Executando o Grace com o arquivo em lote.

```
1 xmgrace grafico.agr -nosafe -batch comando.bat
```

No código da Listagem 4.9, o Grace é novamente executado. Vamos detalhar o comando "*xmgrace grafico.agr -nosafe -batch comando.bat*". Agora o Grace está sendo informado que é para abrir o arquivo "grafico.agr", ativiando a permissão para edição do gráfico e novamente que os comandos para ser executados estão dentro do arquivo "comando.bat", que neste caso so comandos são para exportar os valores do HMW e fechar o Grace.

Executada o código da Listagem 4.9, será criado o arquivo "hmw.dat" com os valores do HMW calculados. Esse arquivo exportado possui duas colunas, a primeira coluna é uma sequência de valores crescentes, de acordo com os dados que foram exportados e a segunda coluna com os valores do HMW calculados.

Listagem 4.10: Detalhes do *script* - Obtendo os valores de um arquivo de dados.

```
1 cat hmw.dat | cut -d" " -f2 > hmwTemp.dat
```

O código da Listagem 4.10, o arquivo exportado "hmw.dat" será manipulado para que seja extraído apenas a segunda coluna com os valores desejados. Para isso é executado o comando *"cat hmw.dat | cut -d" " -f2 > hmwTemp.dat"*, vamos detalhar o comando. Mas antes, vamos ilustrar o formato que o arquivo exportado possui. O formato do arquivo é semelhante ao apresentado a seguir (os valores são apenas um exemplo):

0 1.2
1 2.1
3 3.4
...
59 1.5

Assim, o comando "cat hmw.dat" extrai o conteúdo do arquivo "hmw.dat" e o repassa a outro comando através do comando "|"(pipe), que pega o conteúdo do arquivo extraído pelo *cat* e o repassa como entrada para o comando *cut*.

O comando “*cut -d“ ” -f2 <conteúdo do arquivo repassado>*”, separa o arquivo repassado em duas colunas, as colunas são deliminadas pelo caracter “ ”, a opção “-f2” diz que é para extrair apenas a segunda coluna, ou seja, as que tem os valores 1.2, 2.1, 3.4, ..., 1.5. Veja que a saída do comando “cut” é redirecionada para um arquivo, somente com os valores selecionados, e o arquivo é criado com o nome de “hmwTemp.dat”. Já temos os valores do HMW calculados.

Listagem 4.11: Detalhes do *script* - Agrupando arquivos de dados.

```
1 paste tamanhoPasso.bat  hmwTemp.dat > hmw.dat
```

No código da Listagem 4.11 é criado um novo arquivo da união dos arquivos “tamanhoPasso.dat” e “hmwTemp.dat”. Mas o arquivo resultante possui duas colunas, sendo a primeira com os valores do arquivo “tamanhoPasso.dat” e a segunda com os valores do arquivo “hmwTemp.dat” em que as colunas são separadas por um espaço em branco. O comando que posiciona o conteúdo dos dois arquivos em colunas e coloca um espaço em branco entre elas é o *paste*. Veja que a saída do comando “paste” cria um novo arquivo, o “hmw.dat”, que contem as duas colunas.

Listagem 4.12: Detalhes do *script* - Removendo os arquivos temporários.

```
1 rm hmwTemp.dat
2 rm comando.bat
3 rm comandoHistograma.bat
4 rm comandoApagar.bat
5 rm tamanhoPasso.bat
```

No código da Listagem 4.12 os arquivos que foram criados e utilizados serão apagados pois não são mais necessários. Para isso é usando o comando “rm”. Este comando recebe como parâmetro o nome de um arquivo a ser removido.

Listagem 4.13: Detalhes do *script* - Inserindo os comandos dentro de um arquivo *batch*.

```
1 echo "XAXES SCALE LOGARITHMIC" >> comando.bat
2 echo "xaxis label \"Tamanho do Passo\"" >> comando.bat
3 echo "YAXES SCALE LOGARITHMIC" >> comando.bat
4 echo "yaxis label \"HMW\"" >> comando.bat
5 echo "AUTOSCALE" >> comando.bat
```

No código da Listagem 4.13 é criado o arquivo de configuração para o gráfico a ser criado. Vamos explicar os comandos que são inseridos dentro do arquivo “comando.bat”. Na linha 3 diz ao Grace que o eixo X terá que ser representado na escala logarítmica. Na primeira linha o código da Listagem 4.13 indica ao Grace que a legenda do eixo X é “Tamanho do Passo”. Na linha 3, indica que a escala logarítimica deve também ser usada no eixo Y. Na linha 4, indica que a legenda do

eixo Y é “HMW”. Na linha 5 é informado ao Grace para reajustar a exibição do gráfico, pois as curvas podem não aparecer corretamente.

Listagem 4.14: Detalhes do *script* - Executando o Grace com o arquivo em lote.

```
1 xmgrace hmw.dat -batch comando.bat -saveall "graficoFinal.agr"
```

O código da Listagem 4.14 possui o seguinte comando “*xmgrace hmw.dat -batch comando.bat -saveall "graficoFinal.agr"*”. Este comando faz com que o Grace seja executado (xmgrace), em que o arquivo de dados a ser importado é o “hmw.dat”, que os comandos a serem executados estão no arquivo “comando.bat” e que o gráfico gerado deve ser salvo com o nome de “graficoFinal.agr”. O Grace ainda irá permanecer aberto para que alguma alteração ainda possa ser feita antes de gerar a figura a partir do gráfico. Feitas as edições caso sejam necessárias, as salve acessando o menu *File*, opção *Save* e feche o Grace, acessando a opção *Exit*. O *script* irá continuar a ser executado.

Listagem 4.15: Detalhes do *script* - Removendo um arquivo de imagem EPS.

```
1 rm grafico.eps
```

O código da Listagem 4.15, o arquivo do Grace criado anteriormente, “grafico.eps” será removido com a utilização do comando do sistema “rm”.

Listagem 4.16: Detalhes do *script* - Adicionando comandos a um arquivo *batch*.

```
1 echo " PRINT TO \"grafico.eps\" " > comando.bat
2 echo " HARDCOPY DEVICE \"EPS\" " >> comando.bat
3 echo " PRINT " >> comando.bat
4 echo " EXIT  " >> comando.bat
```

No código da Listagem 4.16 é criado novamente o arquivo de comandos a serem executados pelo Grace. Neste caso, os comandos para exportar o gráfico no formato de uma figura EPS. Vamos analisar os comandos. Na linha 1, o comando “PRINT TO "grafico.eps"” faz com que o gráfico seja exportado com o nome “grafico.eps”. A linha 2 é informado ao Grace que o gráfico será exportado no formato EPS. Na linha 3 faz com que o Grace imprima a figura no formato EPS com o nome “grafico.eps”. Na linha 4 faz com que o Grace seja fechado.

Listagem 4.17: Detalhes do *script* - Executando o Grace com o arquivo em lote.

```
1 xmgrace -nosafe graficoFinal.agr -batch comando.bat
2 rm comando.bat
```

Agora basta executar os comandos no Grace. Isto é feito no código da Listagem 4.17, com o comando *“xmgrace -nosafe graficoFinal.agr -batch comando.bat”*.

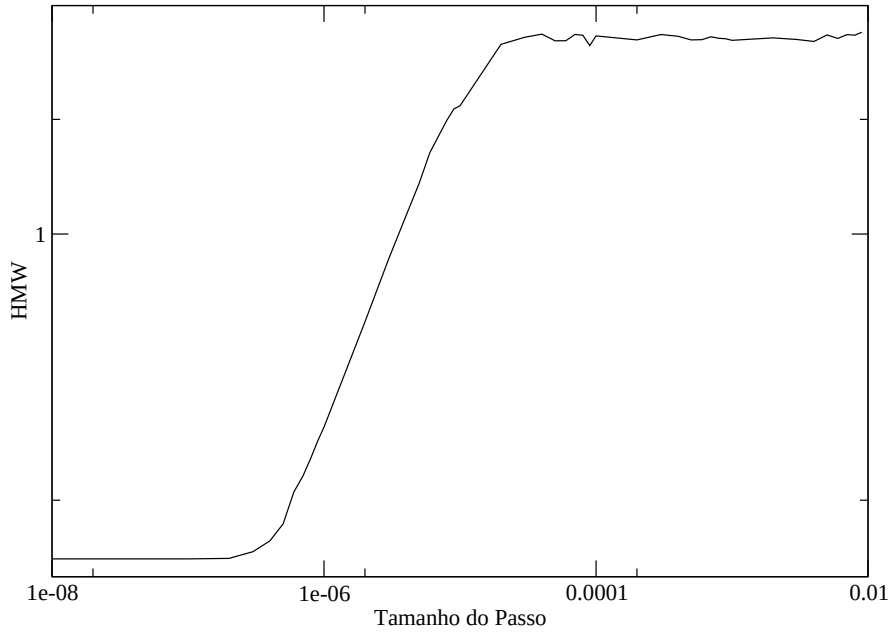

Figura 4.1: Imagem resultante.

Este comando informa ao Grace que é para abrir o arquivo “graficoFinal.agr”, no modo de edição (-nosafe) e executar os comandos do arquivo “comando.bat” (-batch comando.bat). Após executar essa linha, um arquivo com a figura do gráfico será criada seguindo as instruções do arquivo “comando.bat”. Na linha 2 o arquivo de configuração “comando.bat” é removido. A figura baseada no gráfico é criada e exportada. O gráfico exportado no formato EPS pode ser visto na figura 4.1.

4.2 Resultados e Discussões

O Grace como ferramenta de análise gráfica de dados é um excelente *software* com muitos recursos. A sua interface é simples e de fácil manuseio. O Grace permite utilizá-lo através de comandos em lotes ou *batch*. Esta é a característica que mais se destacou e foi utilizada neste trabalho. Logo, as atividades que demandavam que o usuário tivesse que acessar diferentes recursos na interface gráfica ficaram resumidas a comandos. Isto possibilitou utilizar o Grace de forma automatizada, o que diminui em muito a ocorrência de erros, pois é criado um padrão para execução dos comandos do Grace.

Para utilizar o Grace de forma automatizada foi necessário integrar outra ferramenta, o *Shell Script*. Com relação ao *Shell Sript* foi possível verificar que é uma excelente ferramenta com muitos recursos, de como utilizar o sistema, software instalados e manipular arquivos. Basicamente, o *Shell Script* foi utilizado para manipular arquivos de dados e criar os arquivos *batch* com os comandos para o Grace.

Durante o processamento dos dados e a geração dos gráficos, existe a possibilidade de parar o processo para possíveis ajustes quando necessário. Isto permite que seja possível customizar alguma característica, em especial dos gráficos. Neste caso, os recursos do Grace e do *Shell Script* forneceram essa característica ao processo.

Ambas as ferramentas quando utilizadas em conjunto propiciaram reduzir drasticamente o tempo de confecção de gráficos. Nos início do desenvolvimento do trabalho de pesquisa (PAIXÃO, 2007), levava-se uma a duas horas para adequar os dados, plotar, configurar o gráfico e gerar a imagem EPS. Mas essa atividade tinha que ser refeita várias vezes ao dia e prejudicava o desenvolvimento do trabalho. Com a criação do método, todo o procedimento se resumiu a alguns segundos e possibilitou ampliar o plano de pesquisa e melhorar a qualidade do trabalho.

Resumindo, a união do Grace com *Shell Script* foi interessante e abre caminho para desenvolvimento de outras aplicações que demandem agilidade no processamento dos dados com eficiência e robustez para a criação de gráficos com alta qualidade.

Código 4.1.1 Script developed - Parte A.

```
#!/bin/bash
# Script creating for manipulate files and generate graphs using Grace
# Developed by Crysttian Arantes Paixão
# on March 29, 2010


#Initializing the variable that will store the name of your data files to be imported
files=""

#Auxiliary variable to count the files
counter=0

#Highest value within the data files
size=‘cat data.dat ‘

#Creating temporary files

touch command.bat
touch commandHistogram.bat
touch commandErase.bat
touch stepSize.bat

for power in ‘seq 8 -1 3‘
do

        for base in ‘seq 1 9‘
        do
                #Creating a reference to the data file
                name="P1.D-"$power"data"$base".dat"
                #Storing the file name inside the variable
                file=$file" "$name
                #Entering the command of Grace that creates the histograms
                echo "HISTOGRAM(S$counter,MESH(0,$size,100),OFF,OFF)" >> commandHistogram.bat
                #Entering the command of Grace that erases an imported data file, once created the histogram file
                #original data are no longer important to the processes that will be made
                echo "KILL S$counter" >> commandErase.bat
                #Creating the file with the step size to be used later
                echo $base"E-"$power >> sizeStep.bat
                #Incrementing counter
                contador=‘echo $contador+1 | bc‘

        done

done

# Grouping commands parallel execution of tasks
# In this case the command that creates the histogram
cat commandHistogram.bat >> command.bat
#and the command to remove files which do not have more need
cat commandErase.bat >> command.bat

# Generating the files and the HMW
# Generating the graph with data files stored in the variable "file", running
# Predetermined commands to create histograms and removing the original data set.
# In this case, Grace remains open to ascertain the procedures implemented and
# Calculate the HMW what must be done manually, that is done just save your edits and exit the Grace to
# The script will continue
xmgrace -nosafe $file  -batch command.bat -saveall graph.agr
```

Código 4.1.2 Script developed - Parte B.

```
# Exporting the HMW
# Create the command to export the file with the calculated values of HMW
# Command to write data into a file in this case hmw.dat
echo "WRITE G0.S0 FILE \"hmw.dat\" " > command.bat
#command to close the grace after exporting the file
echo "EXIT" >> command.bat
#xmgrace running with the commands above predetermined
xmgrace graph.agr -nosafe -batch command.bat

# Creating the final data
# Handling the file reference step size, variable program
# To be appended to their files with HMW
cat hmw.dat | cut -d" " -f2 > hmwTemp.dat
# Joined files
paste sizeStep.bat  hmwTemp.dat > hmw.dat

# Removing temporary files
rm hmwTemp.dat
rm command.bat
rm commandHistogram.bat
rm commandErase.bat
rm stepSize.bat

# Plot the data and setting up the graph
# Putting the x-axis in logarithmic scale
echo "XAXES SCALE LOGARITHMIC" >> command.bat
# Put the label on the x axis
echo "xaxis label \"Step Size\"" >> command.bat
# Putting the y-axis in logarithmic
echo "YAXES SCALE LOGARITHMIC" >> command.bat
# Put the label in the y-axis
echo "yaxis label \"HMW\"" >> command.bat
# Auto tuning graph
echo "AUTOSCALE" >> command.bat

# Opening the chart for final edits
# Running the formatting settings of the graph
xmgrace hmw.dat -batch command.bat -saveall "graphLast.agr"

# Removing the formerly chart created to prevent messages from grace
rm graph.eps

# Create the configuration file to generate the eps figure chart created
# Defining the type of image
echo " PRINT TO \"graph.eps\" " > command.bat
# Setting the print driver
echo " HARDCOPY DEVICE \"EPS\" " >> command.bat
# Printing the graph
echo " PRINT " >> command.bat
#exit to grace
echo " EXIT  " >> command.bat

# Performing the procedures to create the graph
xmgrace -nosafe graphLast.agr -batch command.bat

# Removing the temporary file
rm command.bat
```

Capítulo 5

Conclusão

Neste trabalho foi demonstrado o uso da ferramenta *Shell Script* em conjunto com o *software* Grace para analisar e gerar gráficos a partir de uma base de dados. Apenas alguns comandos do Grace foram utilizados. O Grace é um *software* que possibilita diferentes análises de dados e em conjunto com o Shell Script se torna uma ferramenta mais completa, possibilitando a otimização dos processos de confecção de gráficos.

Apresentamos a construção de um gráfico baseado em um conjunto de dados através da utilização do Grace em conjunto com *Shell Script*. Gráficos mais complexos podem ser feitos da mesma forma, desde que o *script* seja configurado de maneira adequada.

Com relação à escolha do formato EPS para a imagem gerada, está relacionada com a qualidade da imagem. Os gráficos gerados foram adicionados à dissertação feita em Latex, o que melhora em muito a qualidade de impressão.

Com relação ao ganho de tempo para se construir gráficos, utilizando o procedimento criado, notamos um considerável ganho na execução das tarefas com relação ao método interativo utilizando a interface gráfica do Grace. Ressaltamos que com essa implementação, o processo de geração dos gráficos foi padronizado e os erros na criação dos gráficos foram minimizados.

A utilização do *Shell Script* com o Grace é uma boa opção para auxiliar na produção de gráficos. Além de gerar gráficos de boa qualidade, estes foram gerados de forma automatizada e otimizada. O tempo para confecção de gráficos foi otimizado e em conjunto os erros foram minimizados, logo o processo de análise e criação de gráficos foi melhorado.

Esperamos ter colaborado para auxiliar à outros pesquisadores no processo de criação de gráficos usando o *Shell Script* e o Grace. Na literatura não existem materiais que auxiliem na utilização destas duas ferramentas, logo destacamos a importância de um trabalho como este.

Como trabalho futuro pretende-se utilizar *Shell Scipt* e Grace para construir animações com dados resultantes de simulações. Também esperamos desenvolver um material que apresente uma introdução de como utilizar o Grace em conjunto com *Shell Script*.

Referências Bibliográficas

JARGAS, A. M. *Shell Script Profissional*. São Paulo: Novatec Editora, 2008.

MACAN, E. *Linux Comandos De Usuários - Guia de Consulta Rápida*. São Paulo: Novatec Editora, 2004.

MORIMOTO, C. E. *Linux, Entendendo o Sistema*. São Paulo: GDH Press e Sul Editores, 2006.

PAIXÃO, C. A. *Desenvolvimento de um modelo dinâmico para o Biospeckle*. Dissertação (Mestrado) — Departamento de Engenharia Agrícola, Universidade Federal de Lavras - UFLA, Lavras, 2007.

SAADE, J. *Bash - Guia de consulta rápida*. São Paulo: Novatec Editora, 2001.