



Ministério da  
**Ciência, Tecnologia  
e Inovação**



sid.inpe.br/mtc-m21b/2014/04.14.17.39-PUD

## INTRODUÇÃO AO GRID ANALYSIS AND DISPLAY SYSTEM (GrADS)

José Guilherme Martins dos Santos

URL do documento original:

<<http://urlib.net/8JMKD3MGP5W34M/3G5LD6B>>

INPE  
São José dos Campos  
2014

**PUBLICADO POR:**

Instituto Nacional de Pesquisas Espaciais - INPE

Gabinete do Diretor (GB)

Serviço de Informação e Documentação (SID)

Caixa Postal 515 - CEP 12.245-970

São José dos Campos - SP - Brasil

Tel.:(012) 3208-6923/6921

Fax: (012) 3208-6919

E-mail: pubtc@sid.inpe.br

**CONSELHO DE EDITORAÇÃO E PRESERVAÇÃO DA PRODUÇÃO INTELLECTUAL DO INPE (RE/DIR-204):****Presidente:**

Marciana Leite Ribeiro - Serviço de Informação e Documentação (SID)

**Membros:**

Dr. Antonio Fernando Bertachini de Almeida Prado - Coordenação Engenharia e Tecnologia Espacial (ETE)

Dr<sup>a</sup> Inez Staciarini Batista - Coordenação Ciências Espaciais e Atmosféricas (CEA)

Dr. Gerald Jean Francis Banon - Coordenação Observação da Terra (OBT)

Dr. Germano de Souza Kienbaum - Centro de Tecnologias Especiais (CTE)

Dr. Manoel Alonso Gan - Centro de Previsão de Tempo e Estudos Climáticos (CPT)

Dr<sup>a</sup> Maria do Carmo de Andrade Nono - Conselho de Pós-Graduação

Dr. Plínio Carlos Alvalá - Centro de Ciência do Sistema Terrestre (CST)

**BIBLIOTECA DIGITAL:**

Dr. Gerald Jean Francis Banon - Coordenação de Observação da Terra (OBT)

**REVISÃO E NORMALIZAÇÃO DOCUMENTÁRIA:**

Marciana Leite Ribeiro - Serviço de Informação e Documentação (SID)

Yolanda Ribeiro da Silva Souza - Serviço de Informação e Documentação (SID)

**EDITORAÇÃO ELETRÔNICA:**

Maria Tereza Smith de Brito - Serviço de Informação e Documentação (SID)

André Luis Dias Fernandes - Serviço de Informação e Documentação (SID)



Ministério da  
**Ciência, Tecnologia  
e Inovação**



sid.inpe.br/mtc-m21b/2014/04.14.17.39-PUD

## INTRODUÇÃO AO GRID ANALYSIS AND DISPLAY SYSTEM (GrADS)

José Guilherme Martins dos Santos

URL do documento original:

<<http://urlib.net/8JMKD3MGP5W34M/3G5LD6B>>

INPE  
São José dos Campos  
2014



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.

## SUMÁRIO

	<b><u>Pág.</u></b>
1	INTRODUÇÃO ..... 1
2	INSTALANDO O GRADS..... 2
3	OPÇÕES DE LINHAS DE COMANDO ..... 3
4	CONFIGURAÇÕES GERAIS..... 4
4.1.	Habilitando o slogan do GrADS ..... 4
4.2.	Configurando o modo de visualização ..... 4
4.3.	O comando query ou q..... 4
5	INICIANDO O GRADS ..... 9
6	ABRINDO ARQUIVOS..... 10
7	VISUALIZANDO ARQUIVOS ..... 11
8	VARIANDO OU FIXANDO A LATITUDE E LONGITUDE ..... 12
9	VARIANDO O NÍVEL DE PRESSÃO ..... 13
10	FIXANDO OU VARIANDO O TEMPO..... 14
11	TIPOS DE GRÁFICO..... 18
12	CORES USADAS NO GRADS..... 19
13	ESTILOS DE LINHAS DISPONÍVEIS NO GRADS ..... 20
14	MARCADORES DISPONÍVEIS NO GRADS..... 21
15	MODIFICANDO OS RÓTULOS DOS CONTORNOS..... 22
16	MODIFICANDO A ESPESSURA DAS LINHAS DOS CONTORNOS ..... 23
17	MODIFICANDO O INTERVALO DOS CONTORNOS ..... 24
18	VISUALIZANDO VALORES ACIMA DE UM DETERMINADO VALOR ..... 25

19	VISUALIZANDO VALORES ABAIXO DE UM DETERMINADO VALOR .....	26
20	VISUALIZANDO VALORES DADO UM INTERVALO ESPECIFICO .....	27
21	ESPECIFICANDO O INTERVALO DE VARIAÇÃO DO EIXO X E Y .....	29
22	VISUALIZANDO VALORES FORA DE UM LIMITE ESPECIFICADO .....	30
23	GRÁFICO DE BARRA.....	31
24	GRÁFICO DE BARRAS SEM PREENCHIMENTO .....	32
25	GRÁFICO DE FLECHA DO VENTO .....	35
25.1.	Comprimento da escala da flecha.....	35
26	ESCREVENDO TEXTO NO GRÁFICO.....	36
26.1.	Adicionando título.....	36
26.2.	Adicionando títulos aos eixos x e y .....	36
27	ADICIONANDO TEXTO EM LOCAL ESPECIFICO NO GRÁFICO .....	37
28	INVERTENDO OS EIXOS X E Y EM UM GRÁFICO .....	38
29	PROJEÇÕES DISPONÍVEIS NO GRADS .....	39
30	INVERTENDO A ORDEM DOS EIXOS X E/OU Y.....	41
31	ATRIBUINDO CARACTERÍSTICAS AO MAPA.....	42
32	ADICIONANDO STRINGS NOS RÓTULOS DOS CONTORNOS.....	43
33	FIXANDO VALORES PARA O EIXO Y .....	44
34	ALTERANDO CARACTERÍSTICAS DOS EIXOS X E Y.....	45
35	DESLOCANDO O EIXO X E/OU Y EM UM GRÁFICO .....	46
36	GERANDO GRÁFICO DE CORTE VERTICAL (HOVMÖLLER).....	47
37	OBTENDO AS COORDENADAS X E Y DE UM PONTO .....	48
38	DESENHANDO UM MARCADOR .....	49
39	FUNÇÕES INTRÍNSECAS DO GRADS.....	50

39.1.	Capturando uma linha usando a função sublin .....	50
39.2.	Capturando uma palavra usando a função subwrd .....	51
39.3.	Capturando parte de uma palavra usando a função substr .....	52
40	DESENHANDO UMA CAIXA NO GRÁFICO .....	55
41	BARRA DE CORES NO GRADS .....	57
42	SALVANDO FIGURAS NO GRADS .....	54
43	CRIANDO UM ARQUIVO “.GS” NO GRADS .....	55
44	EXECUTANDO ARQUIVO GS .....	56
45	PROGRAMAS ADICIONAIS .....	57
45.1.	Usando o script grib2ctl.pl .....	57
45.2.	Usando o script lats4d.gs .....	57
45.3.	Procedimentos para instalar EOF no GrADS .....	58
45.4.	Instalando o regrid2 no GrADS.....	64
46	LINGUAGEM DE PROGRAMAÇÃO NO GRADS .....	67
46.1.	Operadores.....	67
46.2.	Fluxo de controle.....	68
46.2.1.	Comando while.....	68
46.2.2.	Comando if .....	68
47	SALVANDO ARQUIVO TEXTO NO GRADS .....	74
48	APLICAÇÕES DIVERSAS .....	77
48.1.	Gráfico de linha de uma área .....	77
48.2.	Gráfico Hovmöller (x = longitude e y = tempo).....	77
48.3.	Gráfico Hovmöller (x = latitude e y = altura).....	78
48.4.	Gráfico Hovmöller (x = tempo e y = altura).....	80

48.5.	Gráfico Hovmöller (x = tempo e y = latitude).....	81
48.6.	Mascarando o continente ou oceano .....	82
48.7.	Criando escala de cores personalizada .....	85
48.8.	Usando o script clave.gs para gerar climatologia .....	86
48.9.	Usando o script drawbox.gs para desenhar uma caixa.....	87
48.10.	Usando o script rmean.gs para calcular média móvel .....	89
48.11.	Convertendo a Temperatura de Kelvin para Celsius.....	91
48.12.	Configurando os vetores no gráfico de vento.....	91
48.13.	Calculando a vorticidade .....	94
48.14.	Calculando a divergência.....	95
48.15.	Criando template com dados Netcdf .....	96
48.16.	Preenchendo regiões específicas do gráfico .....	97
48.17.	Modificando as linhas horizontais e verticais .....	99
48.18.	Modificando a cor e espessura da borda da janela do mapa .....	100
49	FUNÇÕES MATEMÁTICAS.....	102
49.1.	Funções trigonométricas.....	102
49.2.	Formatando números .....	102
49.3.	Número real no formato decimal .....	103
49.4.	Obtendo a parte inteira de um número decimal.....	103
49.5.	Calculando o logaritmo natural.....	104
49.6.	Calculando potência .....	104
49.7.	Calculando raiz quadrada.....	105
49.8.	Obtendo o valor absoluto .....	105
49.9.	Calculando exponencial .....	105

49.10.	Resto da divisão.....	106
49.11.	Parte inteira do resto da divisão .....	106
49.12.	Obtendo o tamanho de uma string.....	107
49.13.	Verificando o valor de uma string .....	107
49.14.	Posição de uma string .....	109
49.15.	Gráfico de linha preenchido.....	109
50	TIPOS DE FONTES .....	111
51	LINKS INTERESSANTES SOBRE O GRADS .....	114
52	REFERÊNCIAS BIBLIOGRÁFICAS .....	115



# 1 INTRODUÇÃO

Esta apostila destina-se aos usuários que estão iniciando suas pesquisas por meio do pacote gráfico Grid Analysis and Display System (GrADS). A finalidade deste material consiste em dar uma base inicial para quem nunca utilizou este pacote, então não cabe aqui fazer um levantamento sobre sua história, mas sim dar o máximo de informações com exemplos práticos para que os usuários possam iniciar suas atividades.

Quando iniciamos nossas pesquisas com a utilização do GrADS normalmente nas aulas de Meteorologia Sinótica e Dinâmica, ficamos sem entender praticamente nada, pois são muitos comandos utilizados. Por isso, este tutorial tem como objetivo ajudar novos usuários bem como servir de material de consulta rápida para possíveis dúvidas.

## 2 INSTALANDO O GRADS

Será utilizada a versão do GrADS 1.9 para o sistema operacional Linux 32 bit. Os arquivos de instalação podem ser baixados do seguinte endereço eletrônico <http://www.iges.org/grads/>. Após baixar os arquivos de instalação, descompacte-os e crie o diretório “grads” em “/usr/local/lib” (você deve ter privilégios de superusuário). Todos os arquivos de biblioteca do GrADS (arquivos gs, dat e qualquer script) ficarão neste local. Os arquivos executáveis (gradsc, gradenc, gribmap, etc) deverão ser copiados para “/usr/local/bin”.

Para que o GrADS funcione na sua área, é necessário adicionar a linha abaixo no arquivo “.bashrc” que encontra-se no seu home. Para visualizar este arquivo vá para o seu home e digite no terminal do Linux “ls -a” para ver os arquivos ocultos do sistema. Esses arquivos possuem um ponto na frente do arquivo. Adicione a linha abaixo no seu “.bashrc” e não esqueça de atualizá-lo digitando “source .bashrc”.

```
export GASCRP=/usr/local/lib/grads
```

### 3 OPÇÕES DE LINHAS DE COMANDO

Opções:

**-c “comando”** → abre o GrADS e executa o “comando”

**Exemplo:** `gradsnc -c “sdfopen air.mon.nc”`

Isso executará o GrADS e abrirá o arquivo “.nc”

**-b** → abre o GrADS sem o terminal gráfico (modo batch).

**Exemplo:** `gradsnc -b`

**-l** → abre o GrADS no modo landscape (retrato). O tamanho da janela para esta opção é 11 x 8.5 polegadas.

**Exemplo:** `gradsnc -l`

**-p** → abre o GrADS no modo portrait (paisagem) . O tamanho da janela para esta opção é 8.5 x 11 polegadas

**Exemplo:** `gradsnc -p`

Os comandos podem ser combinados. Veja o exemplo abaixo.

**Exemplo:** `gradsnc -lc “script.gs”`

Isso executará o “script.gs” que será aberto no modo landscape.

## 4 CONFIGURAÇÕES GERAIS

### 4.1. Habilitando o slogan do GrADS

Habilita (ga-> **set grads on**) ou desabilita (ga-> **set grads off**) a propaganda do GrADS (que fica na parte inferior a esquerda como também a data que se encontra do lado direito).

### 4.2. Configurando o modo de visualização

Esta opção configura a janela gráfica do GrADS de acordo com uma cor pré-definida pelo usuário.

**Importante:** O sinal “;” (ponto e vírgula) é interpretado pelo GrADS como sendo a próxima linha do script “.gs”. O símbolo “\*” (asterisco) representa um comentário que não será interpretado pelo script.

**Sintaxe :** **set display color <cor>**. Onde <cor> pode ser white ou black.

#### **Exemplo:**

'reinit' ;\*reinicia o GrADS toda vez que o script for executado

'set display color white' ;\*fixa a cor branca para janela gráfica do GrADS

'c' ;\*limpa a tela para que o comando anterior tenha efeito

'sdfopen vwnd.mon.mean.nc' ;\*abre o arquivo “.nc”

'd vwnd' ;\*visualiza a variável

### 4.3. O comando query ou q

As opções mais interessantes são: config, files, file n, dims, gxinfo, pos, time, clear, reset, reinit, q cltinfo e quit.

#### **Exemplos:**

**ga-> q cltinfo** → fornece detalhes sobre o arquivo aberto

dset air.mon.mean.nc

title Monthly NCEP/DOE Reanalysis 2

undef -9.96921e+36

xdef 144 linear 0 2.5 → número de pontos (144) na direção x, longitude inicial (0) e o incremento (2.5° graus)

ydef 73 linear -90 2.5 → número de pontos (73) na direção y, latitude inicial (-90) e o incremento (2.5° graus)

zdef 5 levels 1000 925 850 700 600 → número de níveis verticais (17) e quais níveis estão presente no seu arquivo

tdef 349 linear 00Z01JAN1979 1mo → número total de tempos (349), primeiro tempo do arquivo (00Z01JAN1979) e seu incremento (1mo = 1mês)

vars 1 → número de variáveis do arquivo

air 17 -999 Monthly Air Temperature on Pressure Levels → nome da variável (air), número de níveis verticais (17), valor tabelado (-999) e uma breve descrição da variável

**ga-> q config** → fornece informações da versão do GrADS em uso bem como os tipos de arquivos que podem ser abertos

Config: v1.9b4 32-bit little-endian readline sdf/xdf netcdf lats printim

Grid Analysis and Display System (GrADS) Version 1.9b4

Copyright (c) 1988-2005 by Brian Doty and IGES

Center for Ocean-Land-Atmosphere Studies (COLA)

Institute for Global Environment and Society (IGES)

This program is distributed WITHOUT ANY WARRANTY

See file COPYRIGHT for more information.

Built Mon May 23 14:17:20 EDT 2005 for i686-pc-linux-gnu

This version of GrADS has been configured with the following options:

- o This is a 32-bit LITTLE ENDIAN machine version.
- o Command line editing (readline) ENABLED.

o CIRES/CDC (<http://www.cdc.noaa.gov>) SDF/XDF interface ENABLED.

Use sdfopen/xdlopen to read NetCDF files.

o DTYPE netcdf is ENABLED; DTYPE hdf5 is DISABLED.

o OPeNDAP (a.k.a. DODS) gridded data interface DISABLED.

o OPeNDAP (a.k.a. DODS) station data interface DISABLED.

o PCMDI (<http://www-pcmdi.llnl.gov>) LATS interface ENABLED.

This version is configured to write GRIB and NetCDF files.

o DAO (<http://dao.gsfc.nasa.gov>) Athena Widget GUI DISABLED.

o NRL/DAO/PCMDI XA or ImageMagick Image Output DISABLED.

o printim command for direct png/gif output ENABLED.

(via the GD Library -- <http://www.boutell.com/gd>)

For additional information please consult <http://grads.iges.org/grads/>

**ga-> q files** → mostra todos os arquivos abertos

File 1 :

Descriptor: uwnd.1979.2006.ctl

Binary: uwnd.1979.2006.bin

File 2 :

Descriptor: vwnd.1979.2006.ctl

Binary: vwnd.1979.2006.bin

**ga-> q file 1** → fornece informações sobre o arquivo que está aberto, no caso de ter sido aberto dois ou mais arquivos. O 1 corresponde ao número do arquivo que está aberto

File 1 :

Descriptor: uwnd.1979.2006.ctl

Binary: uwnd.1979.2006.bin

Type = Gridded

Xsize = 144 Ysize = 73 Zsize = 1 Tsize = 40908

Number of Variables = 1

uwnd 1 33 \*\* u inv comp-u[m/s]

**ga-> q file 2** → fornece informações sobre o arquivo 2 que está aberto. Tem a mesma aplicação da descrição anterior (q file 1)

File 2 :

Descriptor: vwnd.1979.2006.ctl

Binary: vwnd.1979.2006.bin

Type = Gridded

Xsize = 144 Ysize = 73 Zsize = 1 Tsize = 40908

Number of Variables = 1

vwnd 1 33 \*\* u inv comp-u[m/s]

**ga-> q dims** → fornece informações sobre a dimensão do arquivo aberto

Default file number is: 1

X is varying Lon = 0 to 360 X = 1 to 145

Y is varying Lat = -90 to 90 Y = 0.999999 to 73

Z is fixed Lev = 700 Z = 1

T is fixed Time = 00Z01JAN1979 T = 1

**ga-> q gxinfo** → fornece informações sobre como está configurada a página, informações sobre os eixos x e y e qual o tipo de projeção usada

Last Graphic = Clear

Page Size = 11 by 8.5

X Limits = 0 to 11

Y Limits = 0 to 8.5

Xaxis = None Yaxis = None

Mproj = 2

**ga-> q pos** → Obtém as coordenadas da página. Digite o comando “q pos” e dê um clique com o mouse em alguma área da página com isso serão retornados alguns valores como no resultado abaixo. O valor 4.15556 corresponde ao ponto x e o valor 3.76079 corresponde ao ponto y.

Position = 4.15556 3.76079 1 0

**ga-> q time** → fornece o tempo fixado pelo usuário

Time = 00Z01JAN1979 to 00Z01JAN1979 Mon to Mon

**clear ou c** → limpa a tela do GrADS mantendo todas as configurações

**reset** → limpa a tela do GrADS sem fechar os arquivos

**reinit** → limpa a tela do GrADS e fecha todos os arquivos

**quit** → encerra a sessão do GrADS

## 5 INICIANDO O GRADS

Ao abrir um terminal do Linux, digite o comando “gradsnc” ou simplesmente “gradsc”. Uma nova janela deverá surgir e com isso você já está pronto para usar o GrADS e criar seus gráficos.

## 6 ABRINDO ARQUIVOS

Existem alguns formatos de arquivo mais comuns utilizados no GrADS, são eles: *.bin*, *.gra*, *.grb* e o *.nc*.

Para abrir arquivos “.ctl” deve-se usar o comando open.

**Exemplo** : ga-> open 9503.ctl

Para abrir arquivos “.nc” deve-se usar o comando sdfopen.

**Exemplo** : ga-> sdfopen 9503.nc

## 7 VISUALIZANDO ARQUIVOS

Alguns comandos precisam ser usados para visualizar informações sobre os arquivos abertos, são eles:

**set** – serve para especificar quando, onde e como os dados devem ser mostrados (set lat -20 10; set lon -100 -20; set t 1 20; set z 3, set lev 850, set time 00z22mar1998, set font 2 etc.).

**d ou display** – mostra o campo desejado (ga-> d temp , mostra o campo de temperatura, temp).

**Importante: No GrADS, quando se abrem dois ou mais arquivos, cada um deles recebe uma numeração própria.**

### Exemplo:

```
'reinit'
```

```
'sdfopen uwnd.mon.mean.nc' ;*abrindo o arquivo 1
```

```
'sdfopen vwnd.mon.mean.nc' ;*abrindo o arquivo 2
```

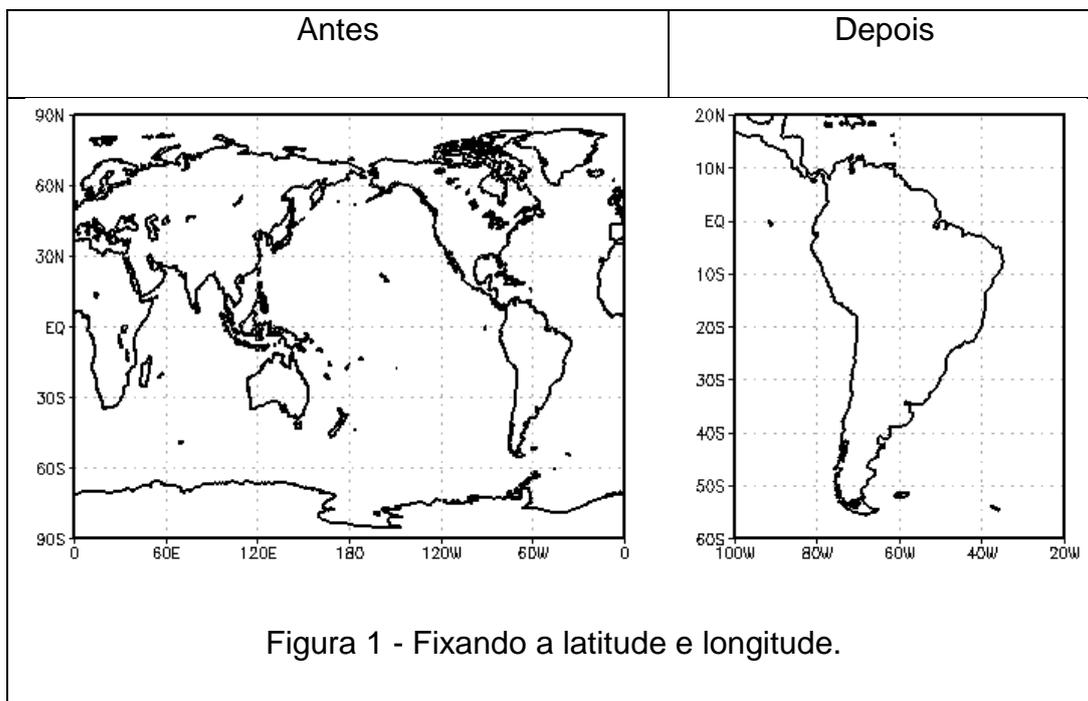
```
'd uwnd' ;*por padrão, o primeiro arquivo aberto não precisa colocar 1 ao lado do nome da variável ao dar o display
```

```
'c' ;*limpa a tela para mostrar a segunda variável
```

```
'd vwnd.2' ;*no caso do segundo arquivo aberto (vwnd.mon.mean.nc), ao visualizá-lo, deverá ser colocado “.2” ao lado do nome da variável. O GrADS entende que o usuário deseja visualizar a variável do arquivo 2.
```

## 8 VARIANDO OU FIXANDO A LATITUDE E LONGITUDE

As latitudes variam de sul (valores negativos) para norte (valores positivos). As longitudes variam de oeste (valores negativos) para leste (valores positivos).



```
'reinit'
```

```
'sdfopen uwnd.mon.mean.nc'
```

```
'sdfopen vwnd.mon.mean.nc'
```

```
'set lat -60 20'
```

```
'set lon -100 -20'
```

## 9 VARIANDO O NÍVEL DE PRESSÃO

Para fixar os níveis de pressão pode-se usar 'set z ' ou 'set lev '.

ga-> set z 1

LEV set to 1000 1000

ga-> set z 2

LEV set to 925 925

ga-> set z 3

LEV set to 850 850

ga-> set z 17

LEV set to 10 10

Outra forma de fixar os níveis.

ga-> set lev 1000

LEV set to 1000 1000

ga-> set lev 925 500

LEV set to 925 500

## 10 FIXANDO OU VARIANDO O TEMPO

```
ga-> set t 1
```

```
Time values set: 1979:1:1:0 1979:1:1:0
```

```
ga-> set t 1 20
```

```
Time values set: 1979:1:1:0 1980:8:1:0
```

ou

```
ga->set time 00z01jan1950 00z20mar1990
```

```
Time values set: 1950:1:1:0 1990:3:20:0
```

```
ga-> set time 00z01jan1950
```

```
Time values set: 1950:1:1:0 1950:1:1:0
```

## 11 TIPOS DE GRÁFICO

Existem vários tipos de saídas gráficas. Aqui, são mostrados apenas alguns modelos.

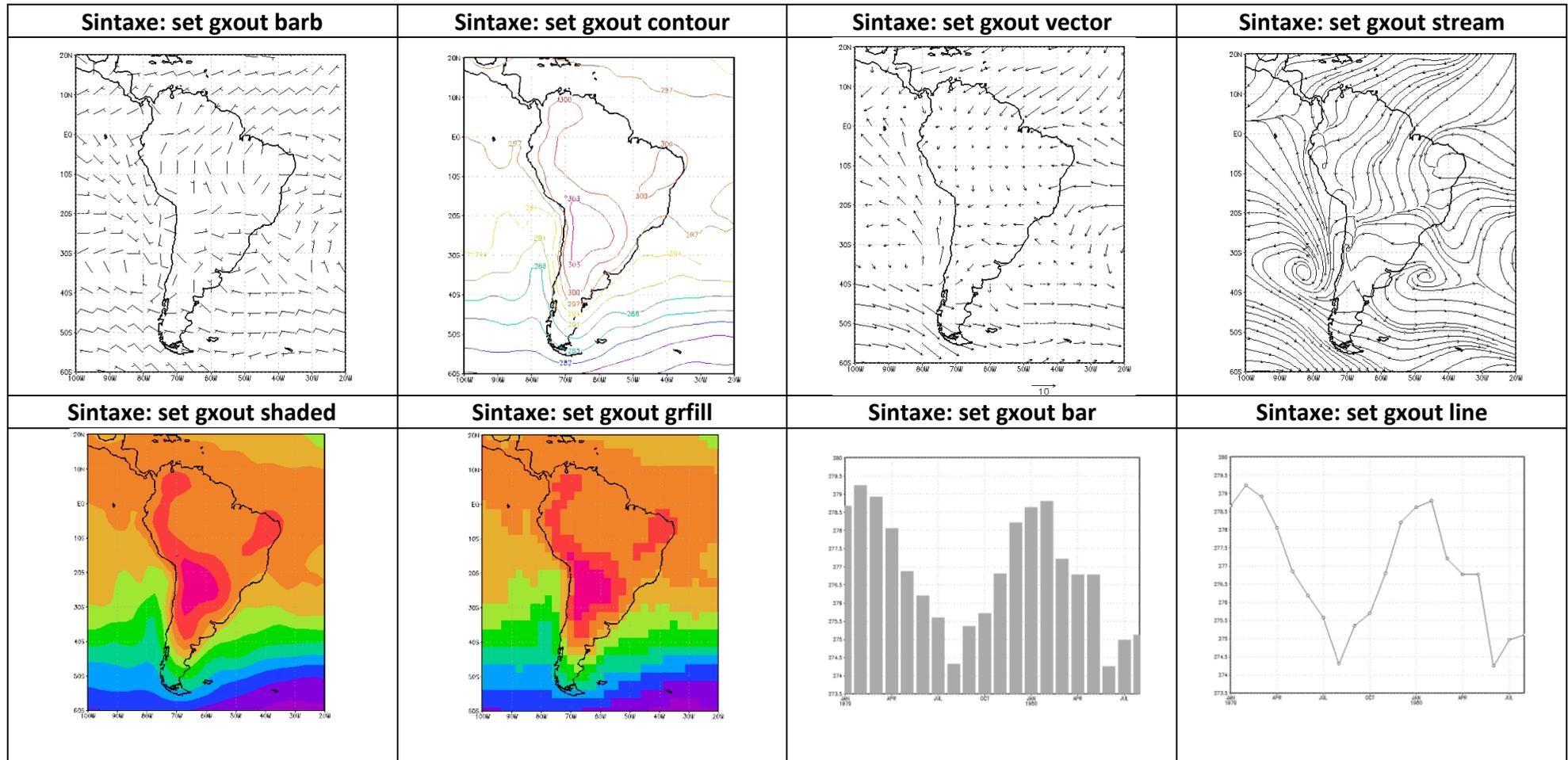


Figura 2 - Tipos de saídas gráficas.

## 12 CORES USADAS NO GRADS

O GrADS disponibiliza 16 cores, porém você pode criar sua própria paleta de cores. Elas são usadas para modificar as cores das figuras como também marcadores, letras, linhas e etc.

**Exemplo:** set ccolor 0 ou set ccolor 7

0		Black	1		white
2		Red	3		green
4		Blue	5		cyan
6		Magenta	7		yellow
8		Orange	9		purple
10		yellow/Green	11		med. blue
12		dark yellow	13		Aqua
14		dark purple	15		grey

Figura 3 - Cores disponíveis no GrADS.

12.1.

### 13 ESTILOS DE LINHAS DISPONÍVEIS NO GRADS

**Exemplo:** set cstyle 4 ou set cstyle 7

0		None	1		solid
2		long dash	3		short dash
4		long dash short	5		dots
6		dot dash	7		dot dot dash

Figura 4 – Linhas disponíveis no GrADS.

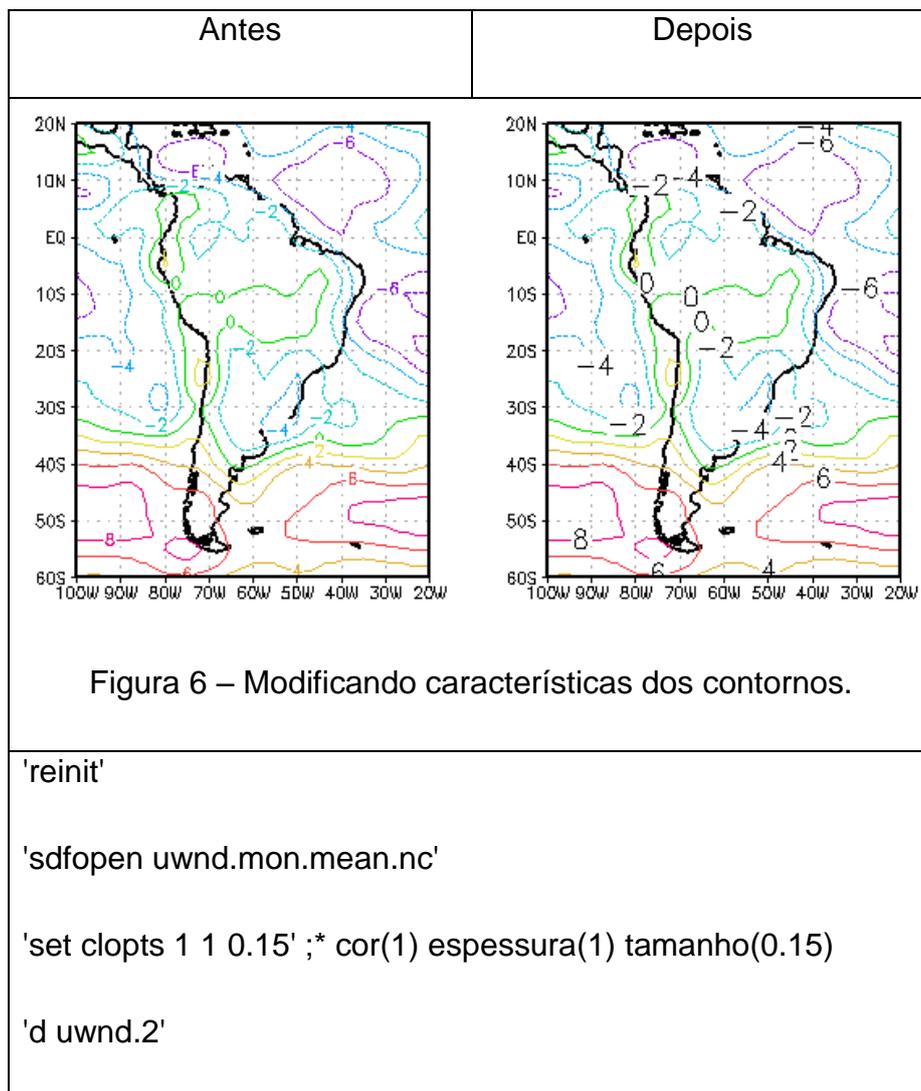
## 14 MARCADORES DISPONÍVEIS NO GRADS

**Exemplo:** set cmark 6 ou set cmark 9

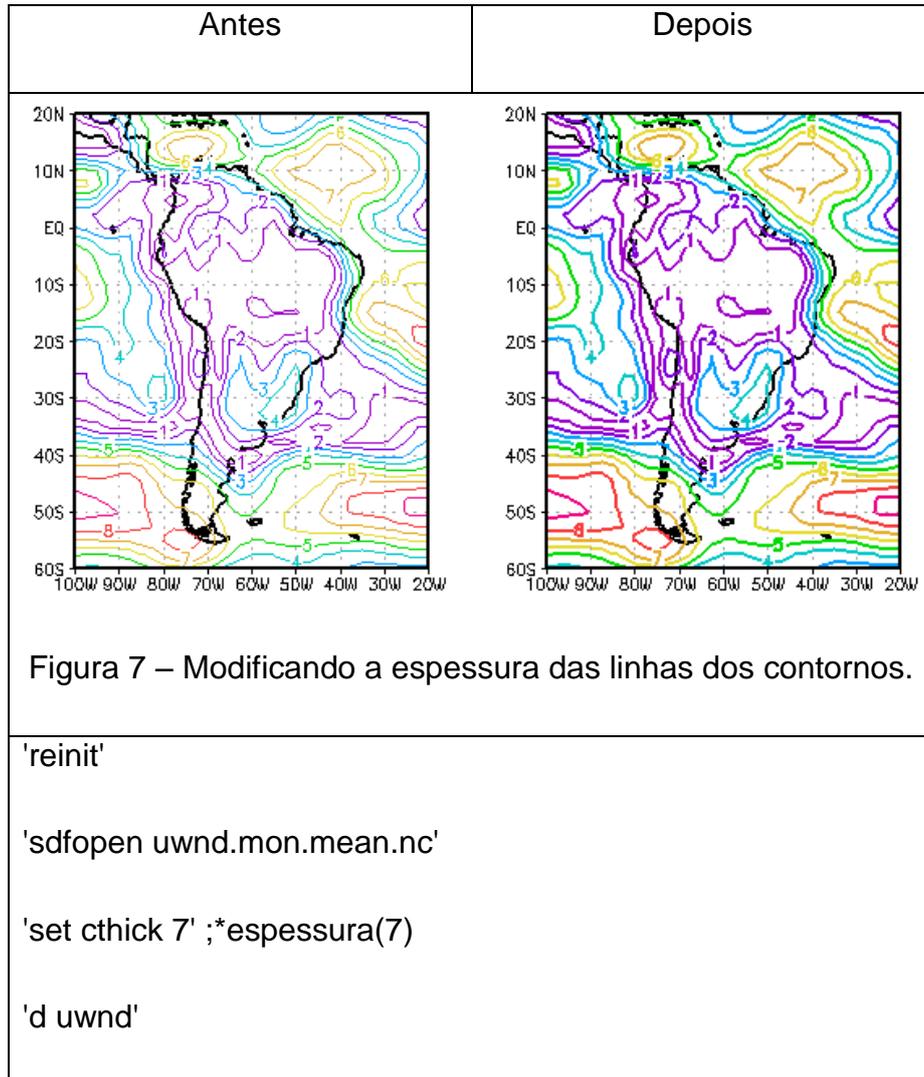
0		Nada	1		cruz
2		círculo aberto	3		círculo fechado
4		quadrado aberto	5		quadrado fechado
6		X	7		diamante
8		triângulo aberto	9		triângulo fechado
10		círculo aberto com linha preta	11		círculo aberto com linha branca

Figura 5 – Marcadores disponíveis no GrADS.

## 15 MODIFICANDO OS RÓTULOS DOS CONTORNOS



## 16 MODIFICANDO A ESPESSURA DAS LINHAS DOS CONTORNOS



## 17 MODIFICANDO O INTERVALO DOS CONTORNOS

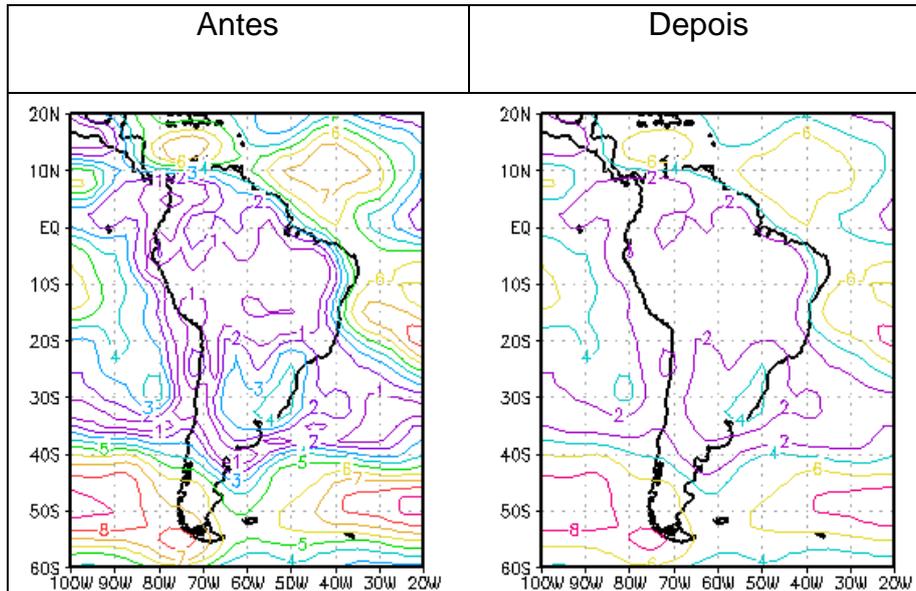


Figura 8 – Modificando o intervalo dos contornos.

```
'reinit'
```

```
'sdfopen uwnd.mon.mean.nc'
```

```
'set lat -60 20 '
```

```
'set lon -100 -20'
```

```
'set cint 2' ;*contornos com intervalos de 2 em 2
```

```
'd uwnd'
```

## 18 VISUALIZANDO VALORES ACIMA DE UM DETERMINADO VALOR

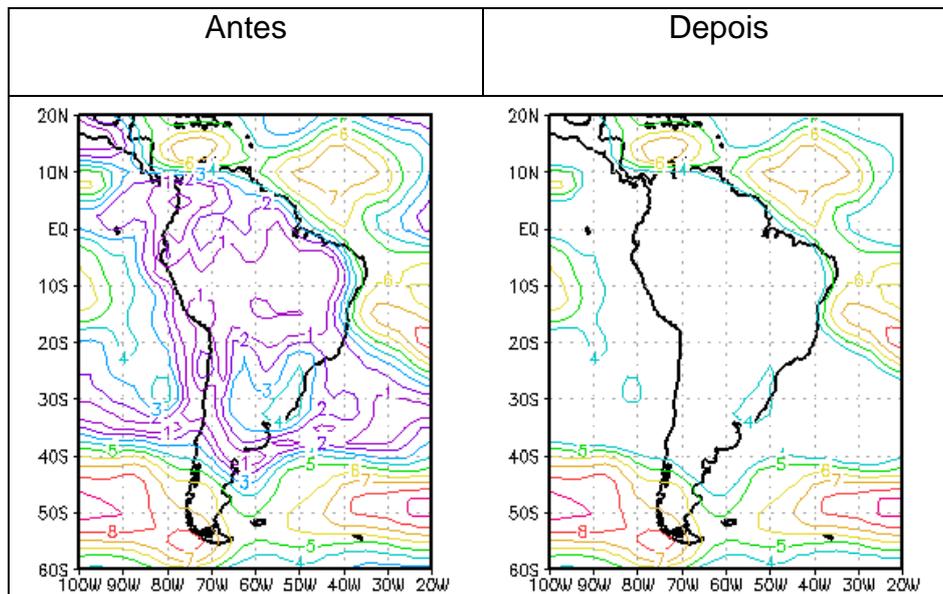


Figura 9 – Visualizando valores acima de um determinado valor.

```
'reinit'
```

```
'sdfopen uwnd.mon.mean.nc'
```

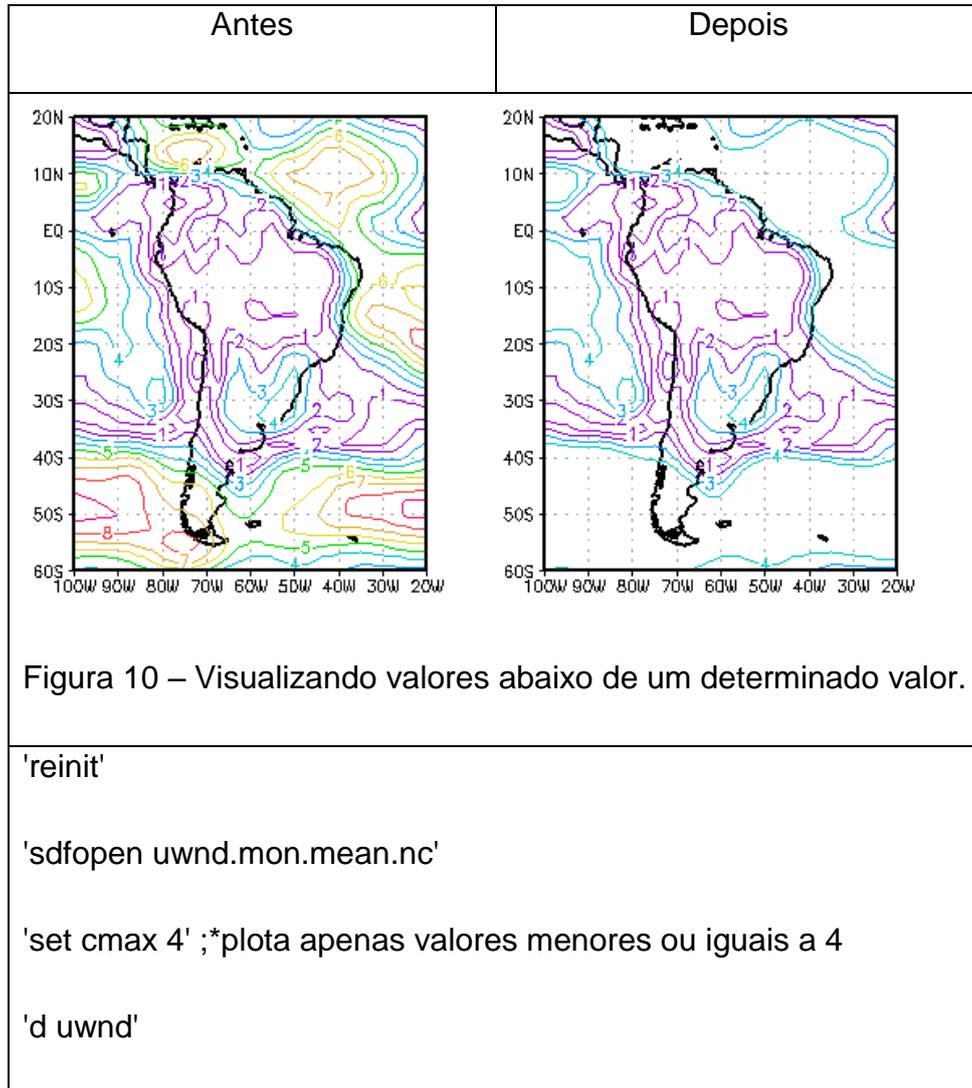
```
'set lat -60 20 '
```

```
'set lon -100 -20'
```

```
'set cmin 4' ;*plota apenas valores maiores e iguais a 4
```

```
'd uwnd'
```

## 19 VISUALIZANDO VALORES ABAIXO DE UM DETERMINADO VALOR



## 20 VISUALIZANDO VALORES DADO UM INTERVALO ESPECIFICO

**Sintaxe :** `set clevs <val1> <val2> <val3> ...<valn>`

onde: `<val1>`, `<val2>` e `<val3>` são os valores limites que serão plotados.

**Exemplo:** Tem-se um campo de vento zonal e deseja-se plotar apenas os valores de 1 até 6 m.s<sup>-1</sup>. Para isso, basta fazer : 'set clevs 1 2 3 4 5 6'. Este procedimento faz com que sejam plotados apenas os valores de 1 a 6.

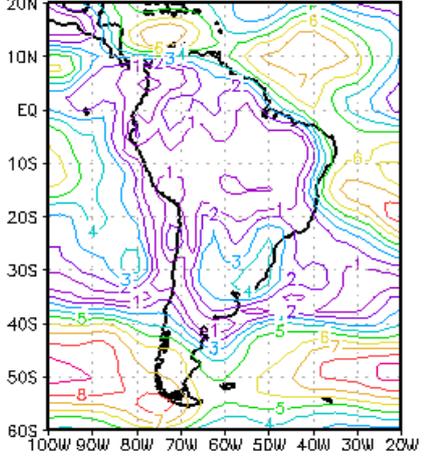
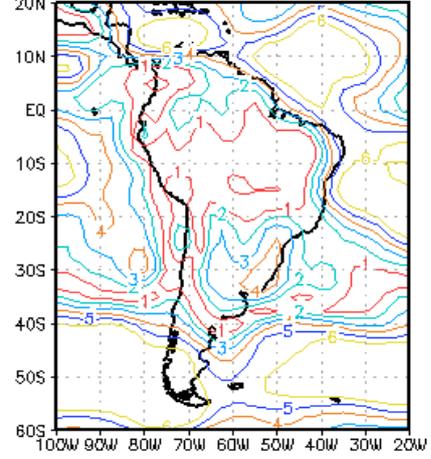
Atribuindo cores aos valores especificados. Isto é útil para definir intervalos e atribuir cores a eles.

**Sintaxe:** `set ccols <val1> <val2> <val3> ...<valn>`

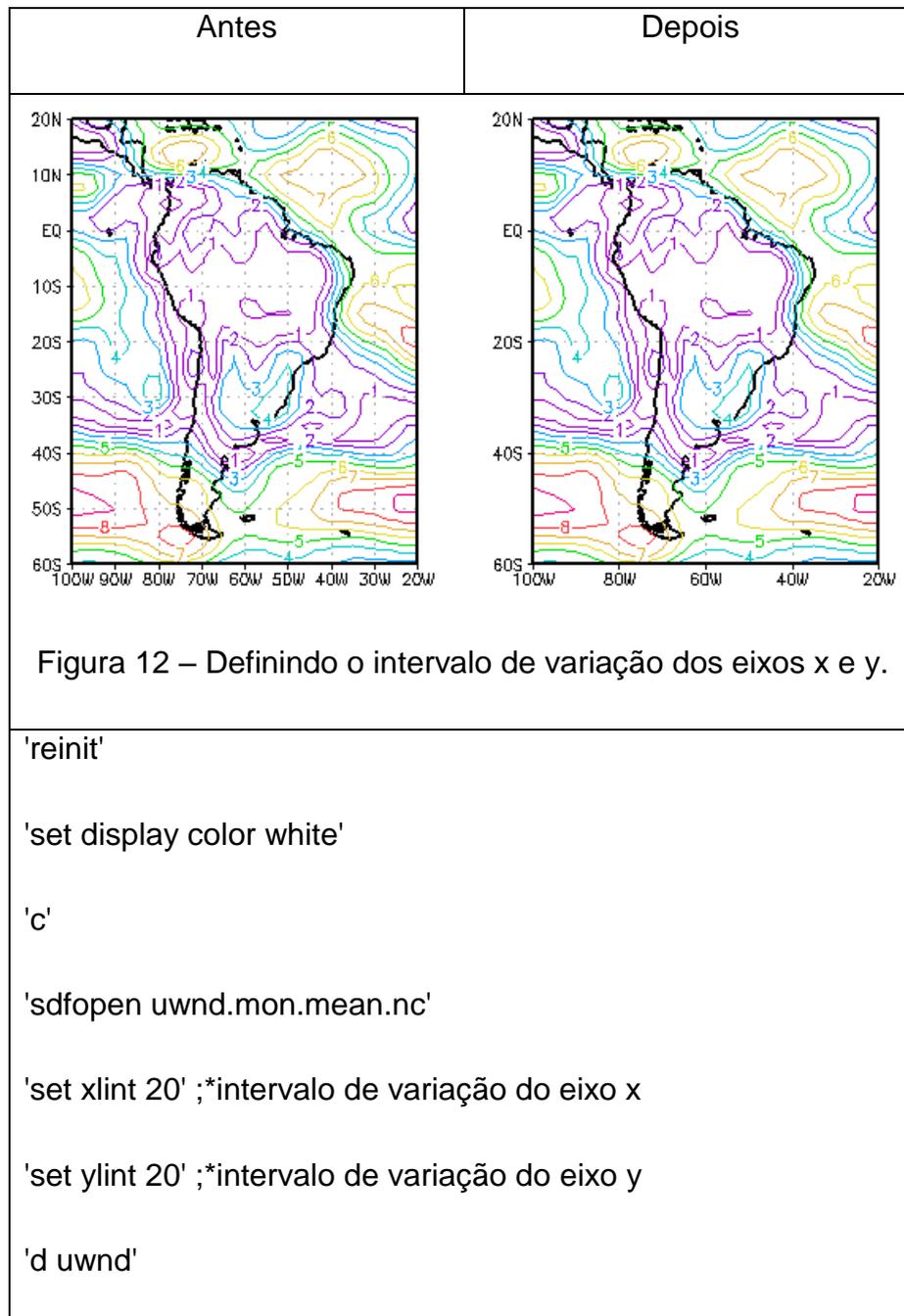
onde: `<val1>`, `<val2>` e `<val3>` são os número das cores (Figura 3) correspondente a cada nível

**Exemplo:** Têm-se os intervalos de valores (vento, temperatura etc.) variando de 1 até 6, totalizando 6 intervalos, logo teremos que selecionar 7 cores.

**Interpretação:** Uma cor será para valores abaixo de 1, uma cor entre 1 e 2, uma cor entre 2 e 3, uma cor entre 3 e 4, uma cor entre 4 e 5, uma cor entre 5 e 6 e uma cor para valores acima de 6. Totalizando assim 7 cores. As cores são selecionadas pelo usuário.

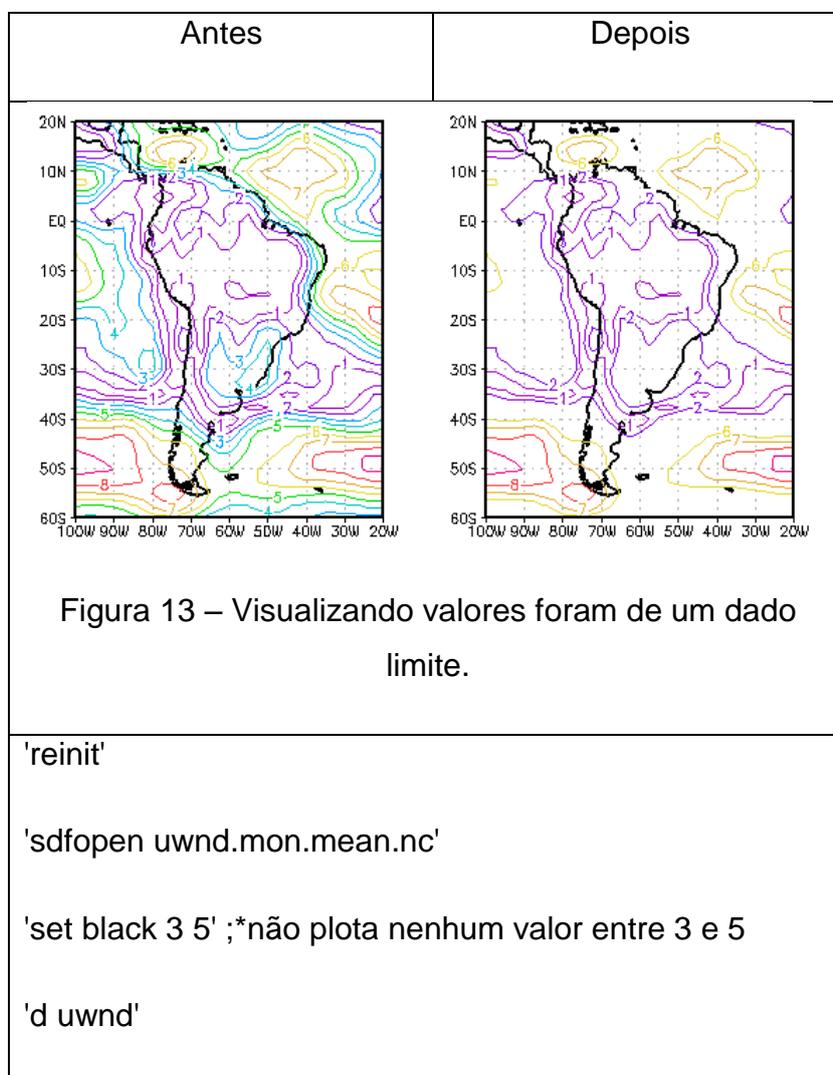
Antes	Depois
	
<p>Figura 11 – Definindo cores e valores para visualização.</p>	
<pre>'reinit'</pre> <pre>'sdfopen uwnd.mon.mean.nc'</pre> <pre>'set clevs 1 2 3 4 5 6';*escala de valores selecionados</pre> <pre>'set ccols 2 5 11 8 4 7 9' ;*cores para cada escala de valor</pre> <pre>'d wnd'</pre>	

## 21 ESPECIFICANDO O INTERVALO DE VARIAÇÃO DO EIXO X E Y



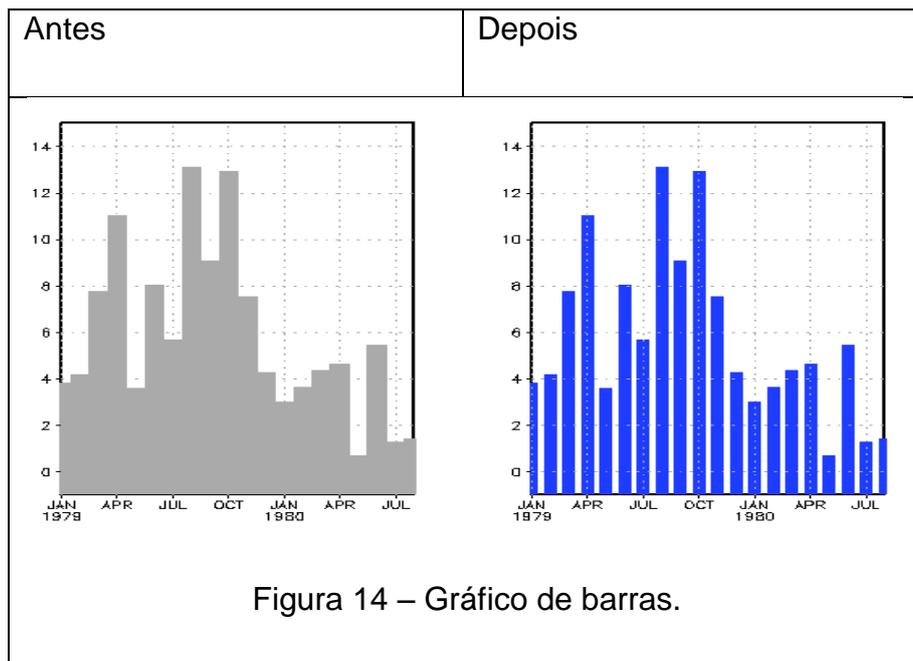
## 22 VISUALIZANDO VALORES FORA DE UM LIMITE ESPECIFICADO

Por exemplo, desejamos visualizar os valores abaixo de 3 e acima de 5. Vejamos o exemplo abaixo para fixação.



## 23 GRÁFICO DE BARRA

Atribuindo espaço entre as barras (**set bargap <val>**). Onde: **<val>** é o valor em porcentagem de 0-100. Por padrão, o gráfico de barras é preenchido. Para o caso de barras sem preenchimento, usar a opção para não preenchimento (outline).



```
'reinit'
```

```
'sdfopen uwnd.mon.mean.nc'
```

```
'set lat -60'
```

```
'set lon -100'
```

```
'set t 1 20'
```

```
'set ccolor 4' ;*fixando cor azul para as barras
```

```
'set gxout bar' ;*fixa o gráfico do tipo barra
```

```
'set bargap 40' ;*espaçamento entre cada barra
```

```
'd uwnd'
```

## 24 GRÁFICO DE BARRAS SEM PREENCHIMENTO

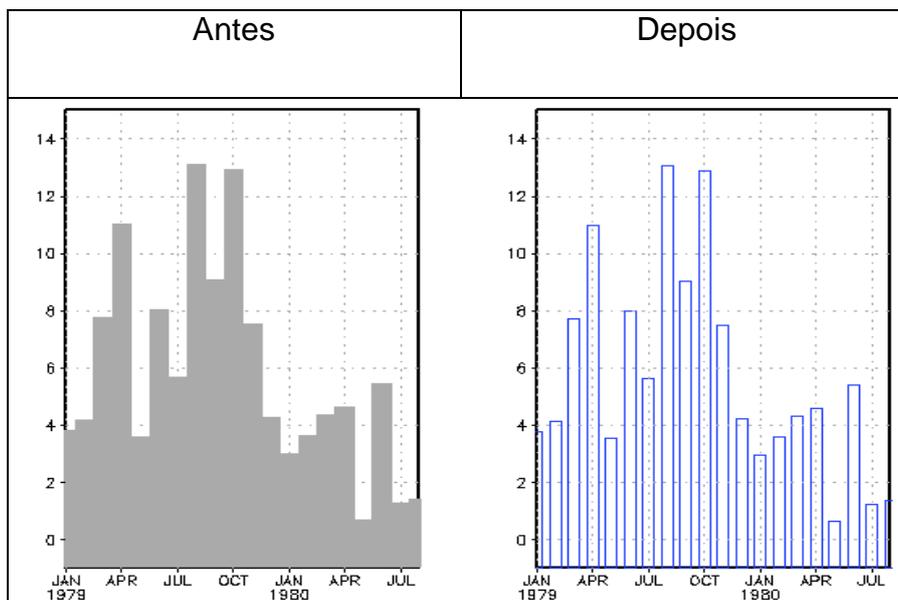


Figura 15 – Gráfico de barras sem preenchimento.

```
'reinit'
```

```
'sdfopen uwnd.mon.mean.nc'
```

```
'set lat -60'
```

```
'set lon -100'
```

```
'set t 1 20'
```

```
'set ccolor 4'
```

```
'set gxout bar'
```

```
'set bargap 40'
```

```
'set baropts outline' ;*barras sem preenchimento
```

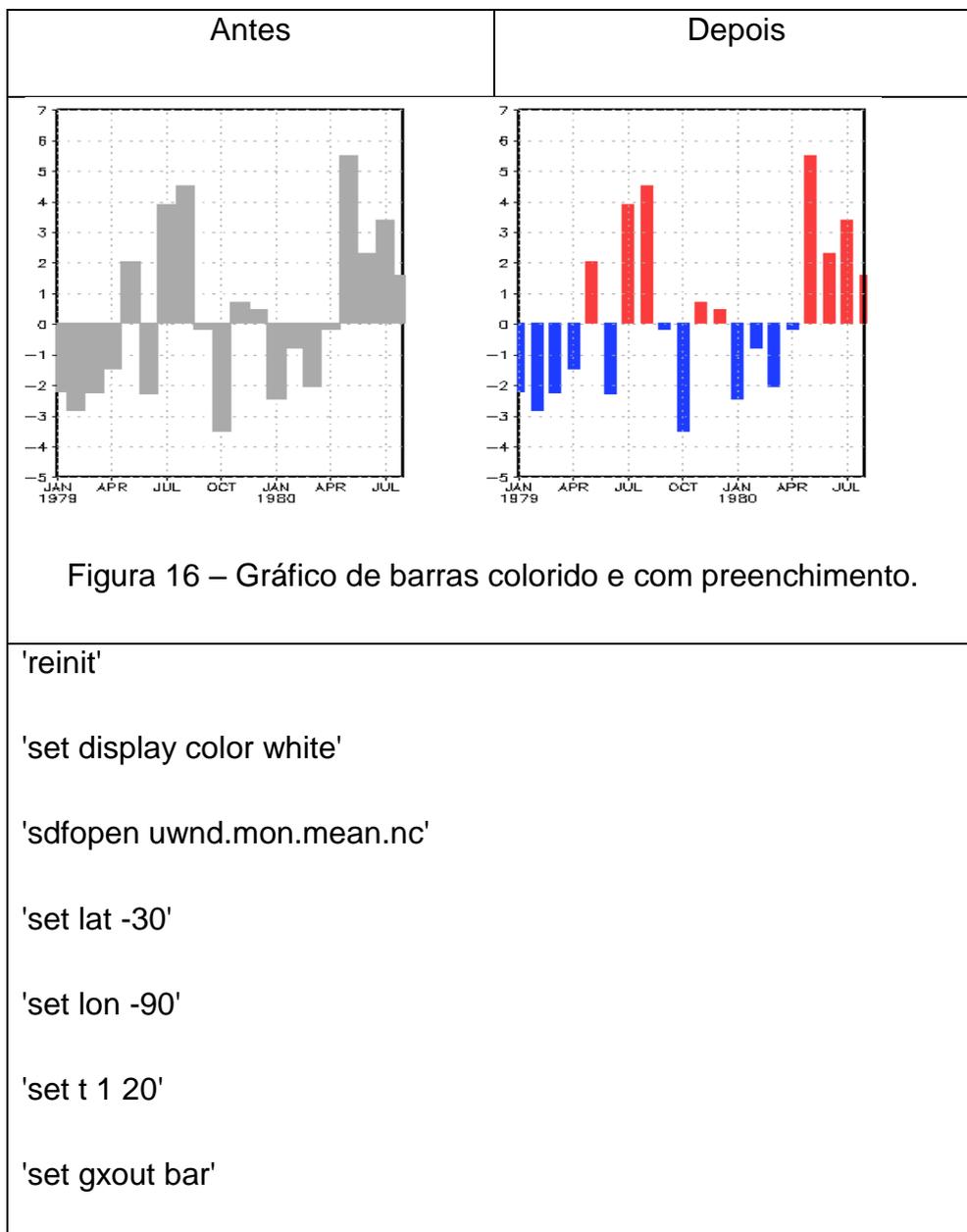
```
'd uwnd'
```

**Importante:** O comando “barbase” é usado em gráficos de barras, o qual é útil quando se deseja fazer gráficos que apresentem valores positivos e negativos.

**Sintaxe:** set barbase <val>

Onde: <val> é o valor desejado. Caso o usuário deseje fazer um gráfico com valores positivos e negativos é aconselhável usar o valor 0.

**Exemplo:** set barbase 0



```
'set bargap 40'
```

```
'set barbase 0' ;*valores positivos e negativos oscilam no valor zero
```

```
'set ccolor 2' ;* selecionar cor vermelha para valores positivos
```

```
'd maskout(uwnd,uwnd)' ;*não plota valores positivos
```

```
'set ccolor 4' ;* selecionar cor azul para valores negativos
```

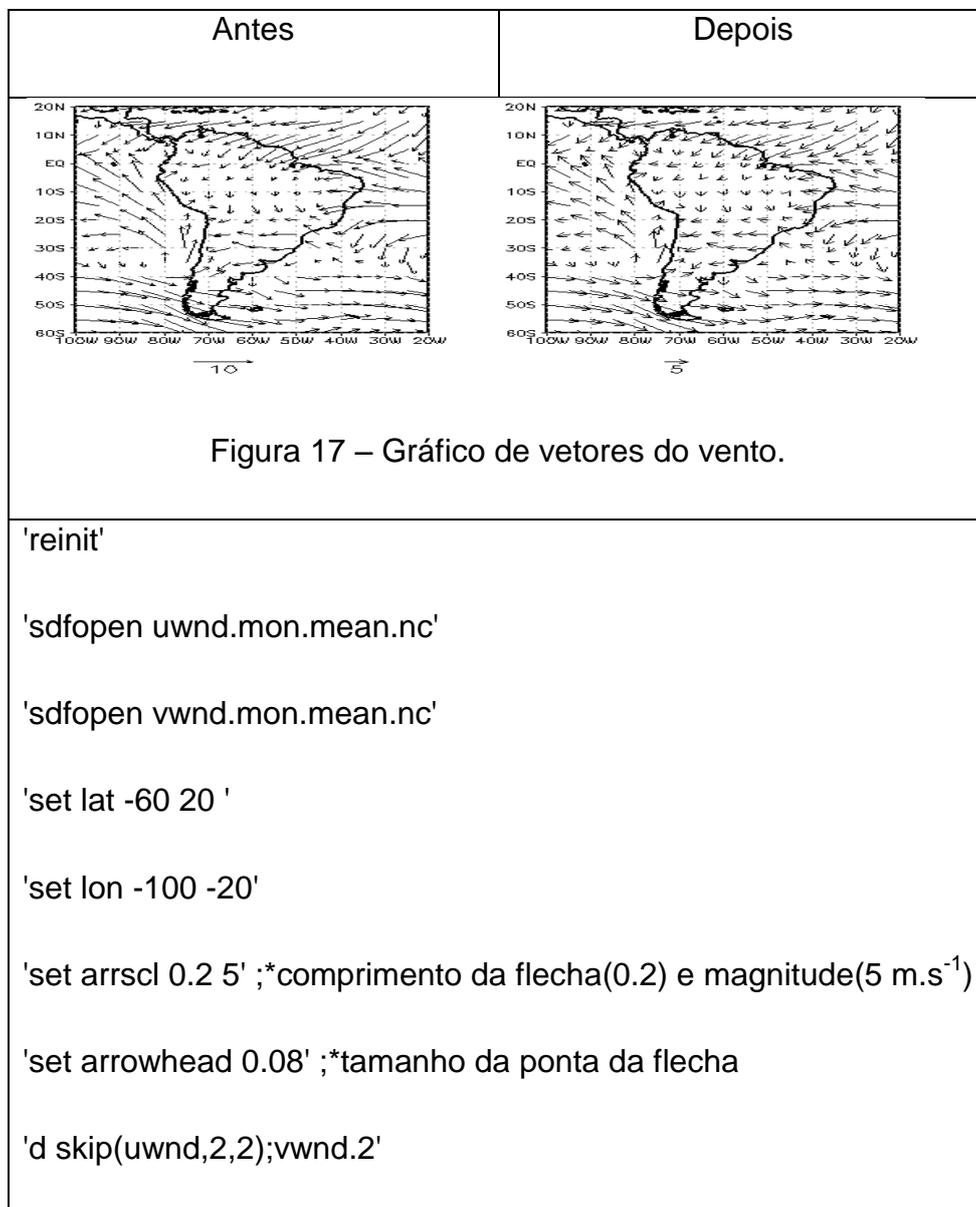
```
'd maskout(uwnd,-uwnd)' ;*não plota valores negativos
```

## 25 GRÁFICO DE FLECHA DO VENTO

### 25.1. Comprimento da escala da flecha

**Sintaxe:** set arrscl <tamanho> <magnitude>

O <tamanho> é o comprimento da flecha em unidades de plotagem (polegadas na página virtual) e deve estar entre 0.5 e 1.0. A <magnitude> é a grandeza do vetor que produzirá uma flecha de especificado tamanho. Os outros comprimentos de flecha serão escalados apropriadamente. Se a magnitude não é dada, então todas terão o mesmo comprimento.



## 26 ESCREVENDO TEXTO NO GRÁFICO

### 26.1. Adicionando título

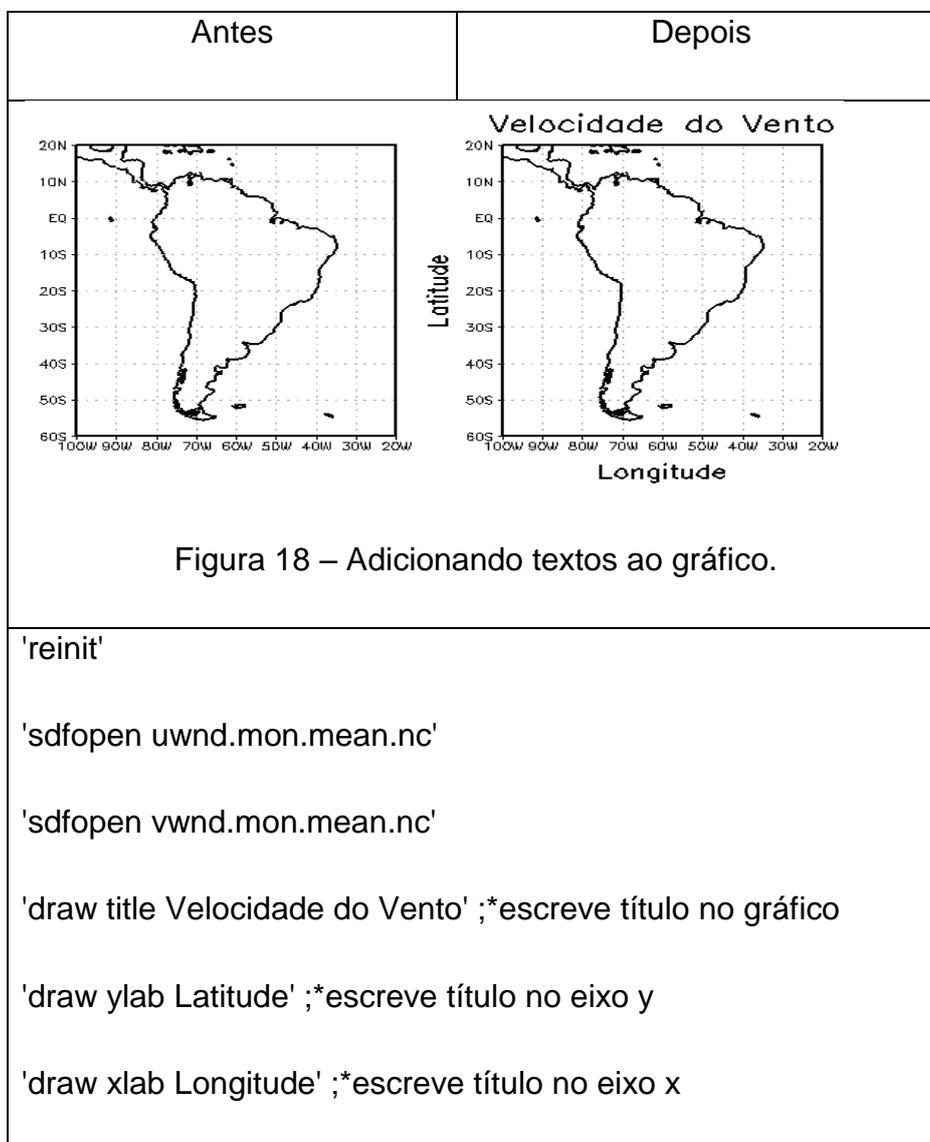
Sintaxe: draw title <texto>

### 26.2. Adicionando títulos aos eixos x e y

Sintaxe: draw xlab <texto>

Sintaxe: draw ylab <texto>

Onde <texto> é o que será escrito no gráfico.



## 27 ADICIONANDO TEXTO EM LOCAL ESPECIFICO NO GRÁFICO

**sintaxe: draw string <x> <y> <seutexto>**

Onde:

<x> e <y> são as coordenadas da página obtidas com o comando “q pos” que serão utilizadas para escrever <seutexto> no seu gráfico.

Para aumentar o tamanho do texto escrito no gráfico, basta fazer:

**sintaxe: set strsiz <valor>**

Onde <valor> pode ser 0.1, 0.2, 0.3. Isso dependera do usuário.

Existe também a opção de mudar a cor, justificar, aumentar a espessura e rotacionar o texto. Isso pode ser feito com o comando abaixo.

**sintaxe: set string <cor> <justificativa> <espessura> <rotação>**

Onde:

<cor> = veja figura de cores (**Figura 3**)

<justificativa> = l, c, r, tl, tc, bl, tc e tr

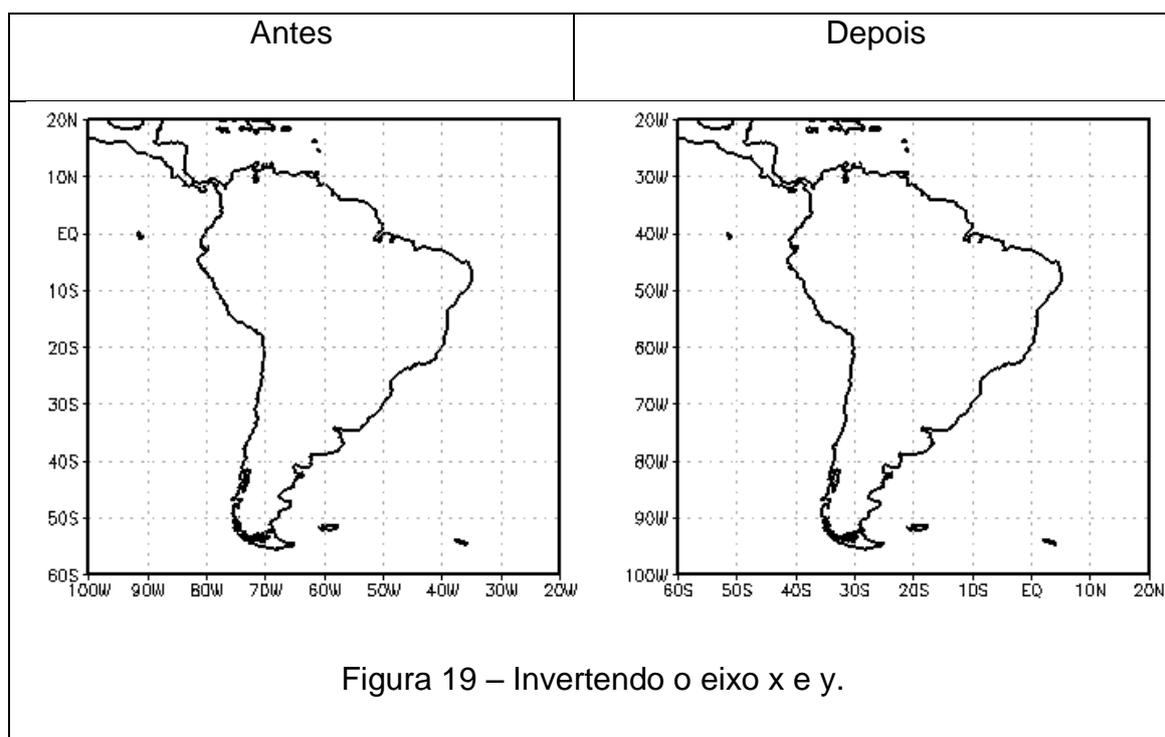
<espessura> = varia de 1 a 6

<rotação> = -90 (sentido horário) até 90 (sentido anti-horário)

## 28 INVERTENDO OS EIXOS X E Y EM UM GRÁFICO

O que é x vira y e vice-versa. Apenas os eixos são afetados e nada acontece com o gráfico.

**Sintaxe: set xyrev on** – ativa e **Sintaxe: set xyrev off** – desativa



```
'reinit'
```

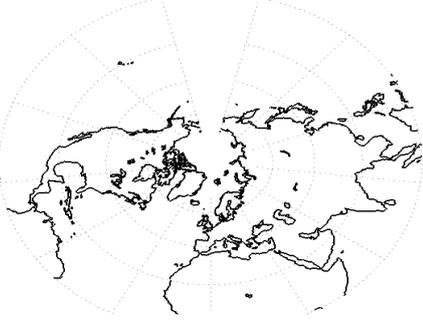
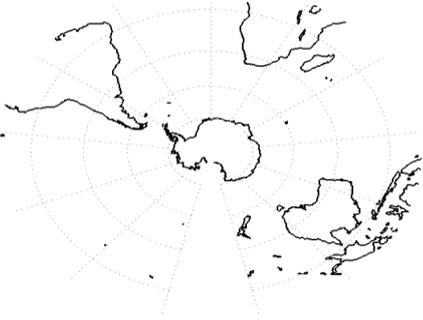
```
'sdfopen uwnd.mon.mean.nc'
```

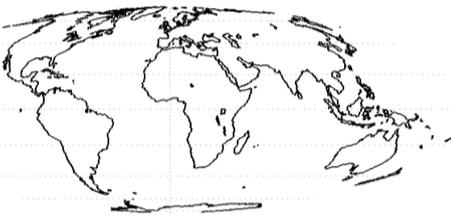
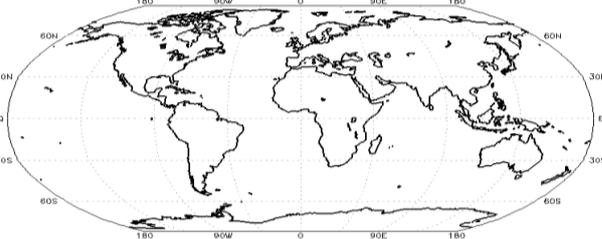
```
'sdfopen vwnd.mon.mean.nc'
```

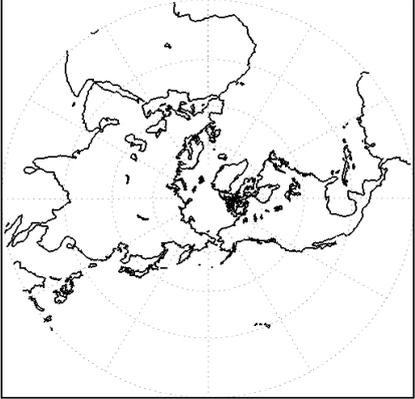
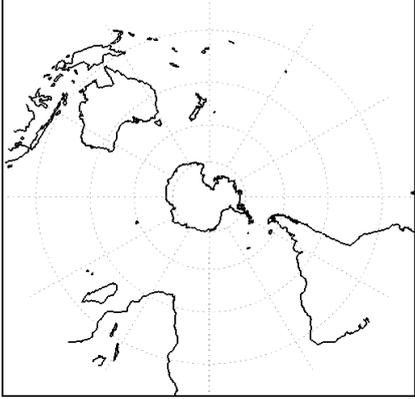
```
'set xyrev on' ;*o que é x vira y e vice-versa
```

```
'd mag(uwnd,vwnd.2)'
```

## 29 PROJEÇÕES DISPONÍVEIS NO GRADS

Lambert – HN	Lambert – HS
	
<p>Figura 20 – Projeções no GrADS. Tipo Lambert para HN e HS.</p>	
<pre>'set lat 0 90'</pre> <pre>'set lon -180 180'</pre> <pre>'set mproj lambert'</pre>	<pre>'set lat -90 0'</pre> <pre>'set lon -180 180'</pre> <pre>'set mproj lambert'</pre>

Mollweide	Robinson
	
<p>Figura 21 – Projeções no GrADS. Tipo Mollweide e Robinson.</p>	
<pre>'set lat -90 90'</pre> <pre>'set lon -180 180'</pre> <pre>'set mproj mollweide'</pre>	<pre>'set lat -90 90'</pre> <pre>'set lon -180 180'</pre> <pre>'set mproj robinson'</pre>

Estereográfica Polar – HN	Estereográfica Polar – HS
	
<p>Figura 22 – Projeções no GrADS. Tipo Estereográfica Polar para HN e HS.</p>	
<pre>'set lat 0 90' 'set mproj nps'</pre>	<pre>'set lat -90 0' 'set mproj sps'</pre>

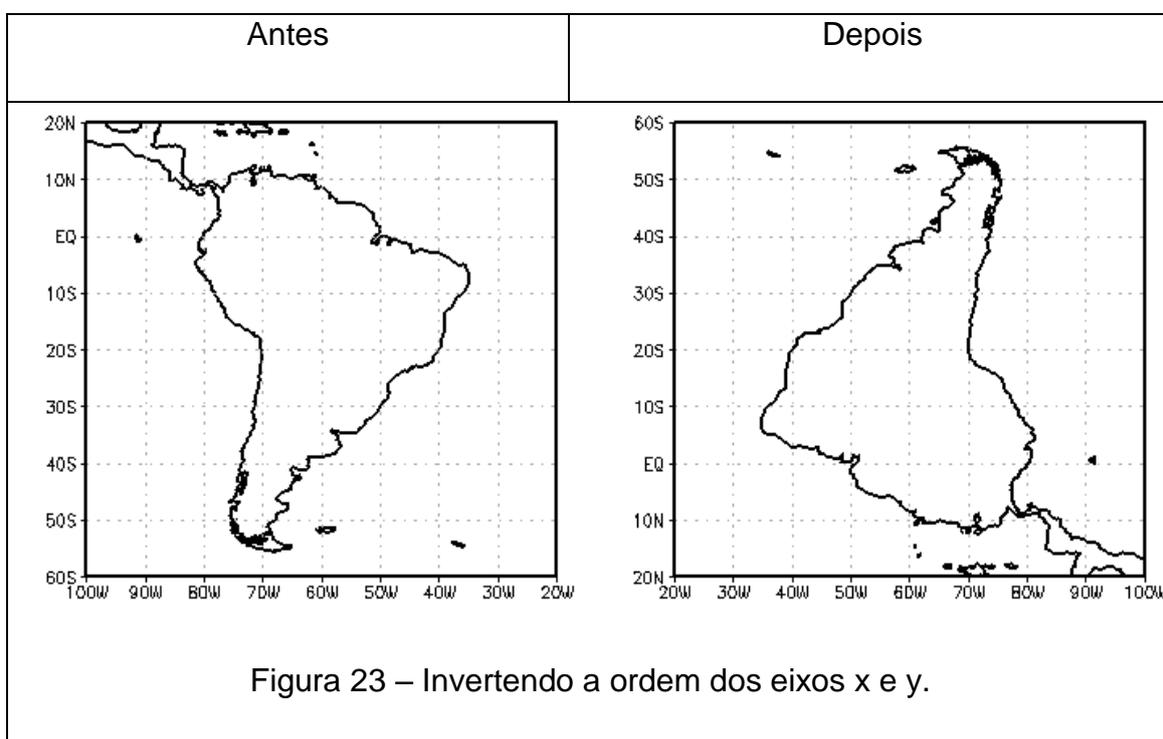
### 30 INVERTENDO A ORDEM DOS EIXOS X E/OU Y

Diferentemente do comando “set xyrev” os eixos e os mapas são modificados.

Para o eixo x: **Sintaxe: set xflip on** ou **set xflip off**

Para o eixo y: **Sintaxe: set yflip on** ou **set yflip off**

**Exemplo:** Têm-se os valores 1 2 3 4 5 6 7 para o eixo x ou y. Ao ativar este comando estes valores passam a ter a seguinte disposição 7 6 5 4 3 2 1.



```
'reinit'  
  
'sdfopen uwnd.mon.mean.nc'  
  
'sdfopen vwnd.mon.mean.nc'  
  
'set xflip on' ;*inverte os valores do eixo x  
  
'set yflip on' ;*inverte os valores do eixo y  
  
'd mag(uwnd,vwnd.2)'
```

### 31 ATRIBUINDO CARACTERÍSTICAS AO MAPA

Este comando atribui cor, estilo de linha e espessura ao contorno do mapa.

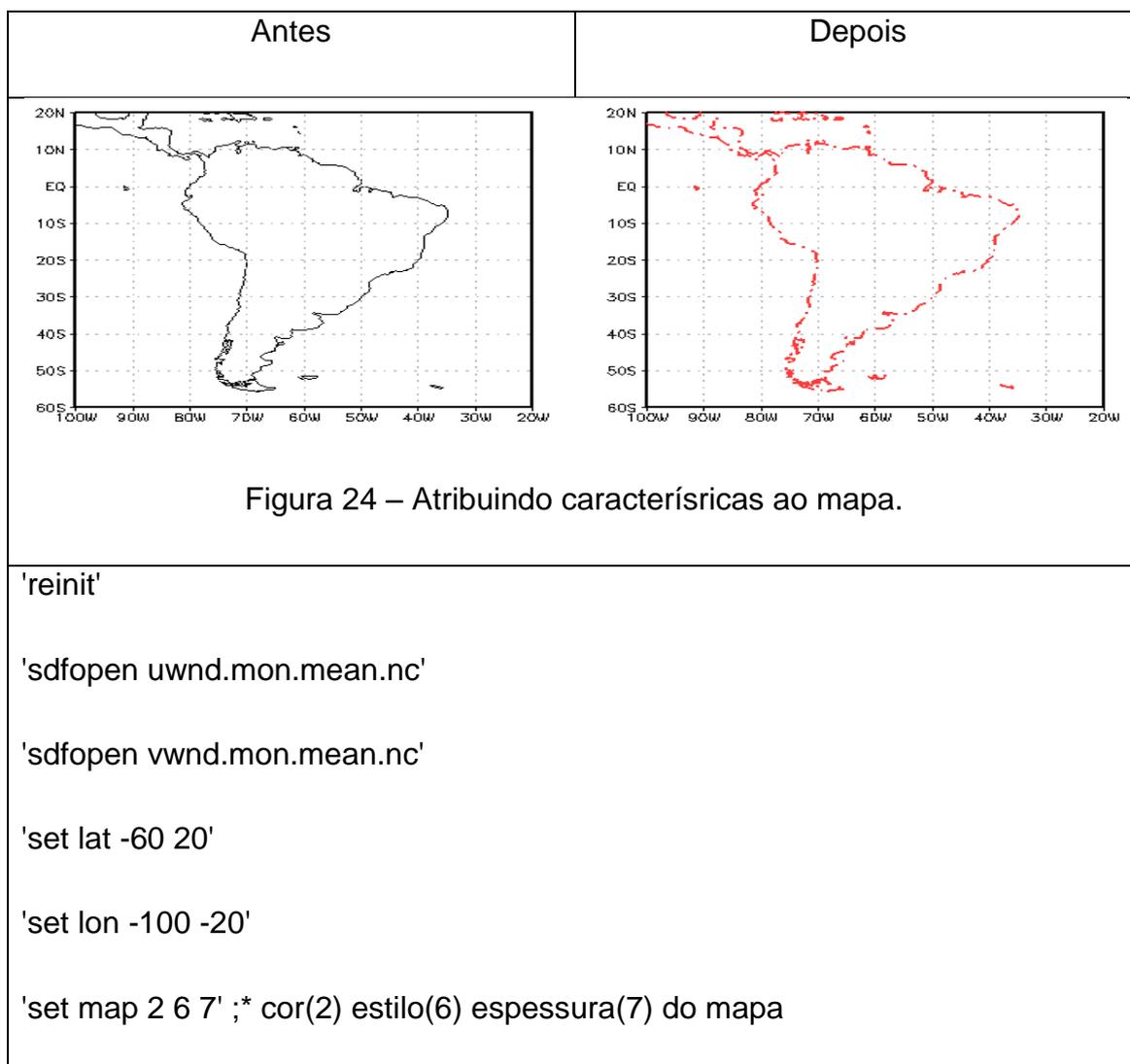
**Sintaxe: set map <cor> <estilo> <espessura>**

Onde:

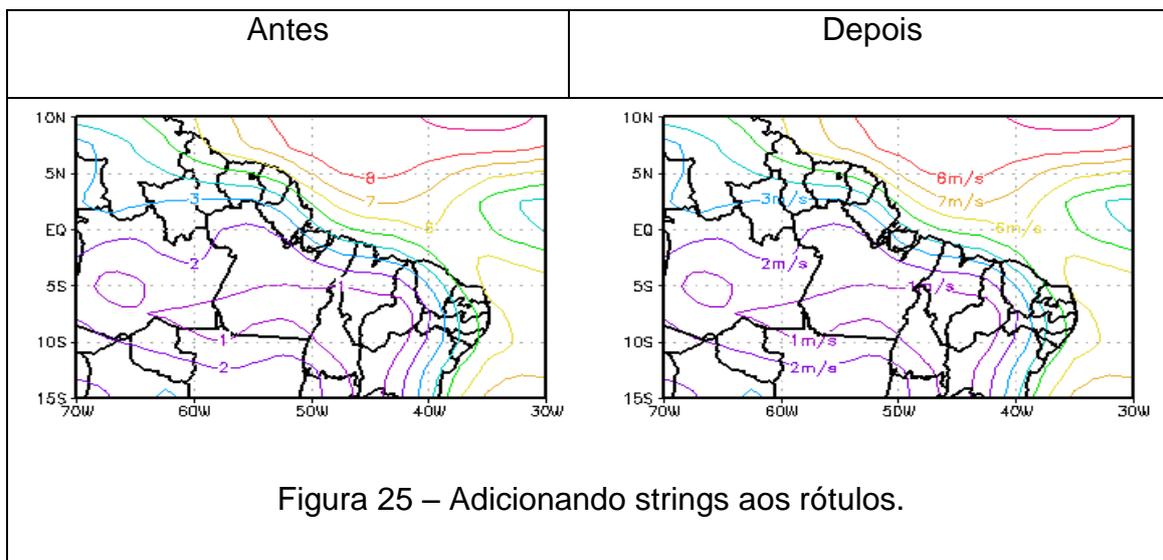
**<cor>** = veja tabela de cores (**Figura 3**)

**<estilo>** = estilo de linha (**Figura 4**)

**<espessura>** = varia de 1 a 6



## 32 ADICIONANDO STRINGS NOS RÓTULOS DOS CONTORNOS



```
'reinit'
```

```
'sdfopen uwnd.mon.mean.nc'
```

```
'sdfopen vwnd.mon.mean.nc'
```

```
'set lat -15 10'
```

```
'set lon -70 -30'
```

```
'set clab %gm/s' ;*adiciona "m/s" em cada rótulo dos contornos
```

```
'd mag(uwnd,vwnd.2)'
```

Para adicionar o símbolo de porcentagem (%) em cada rótulo dos contornos, basta fazer o comando abaixo.

```
ga->set clab %g%%
```

Para colocar dois dígitos após a casa decimal.

```
ga->set clab %.2f
```

O número 2 representa 2 dígitos após a casa decimal.

### 33 FIXANDO VALORES PARA O EIXO Y

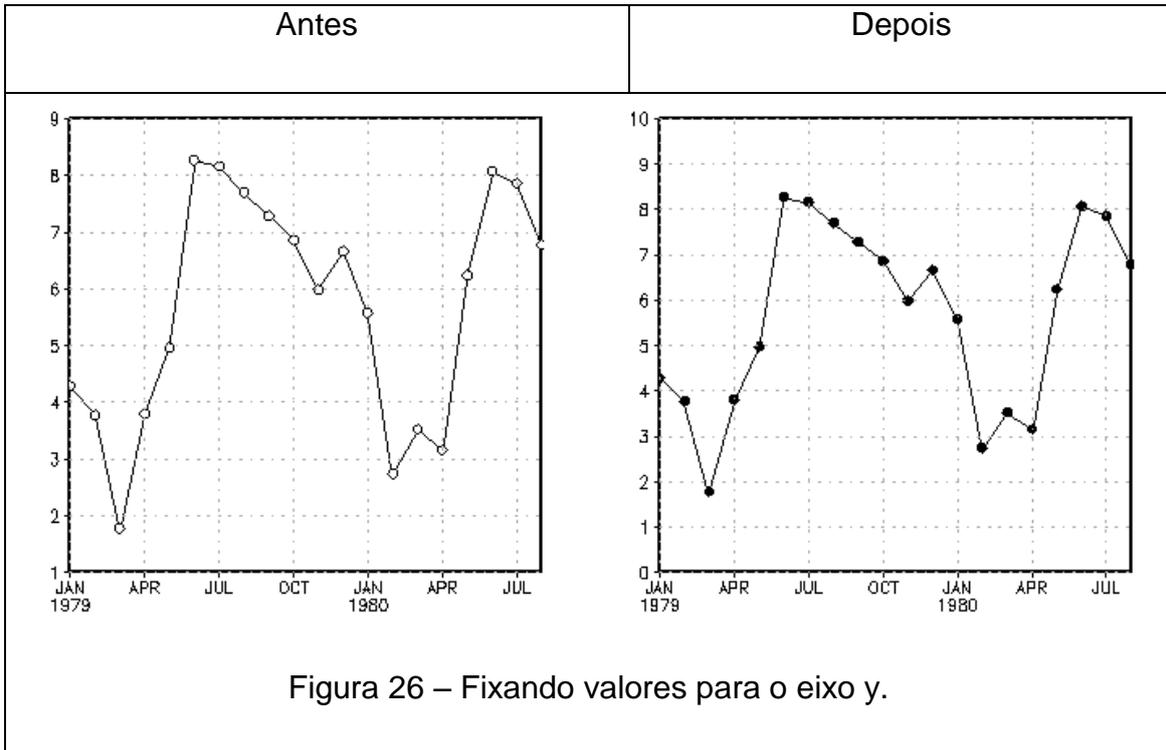


Figura 26 – Fixando valores para o eixo y.

```
'reinit'

'sdfopen uwnd.mon.mean.nc'

'sdfopen vwnd.mon.mean.nc'

'set vrange 0 10' ;*define a escala para o eixo y com valor mínimo de 0 e
máximo de 10

'set lat 0'

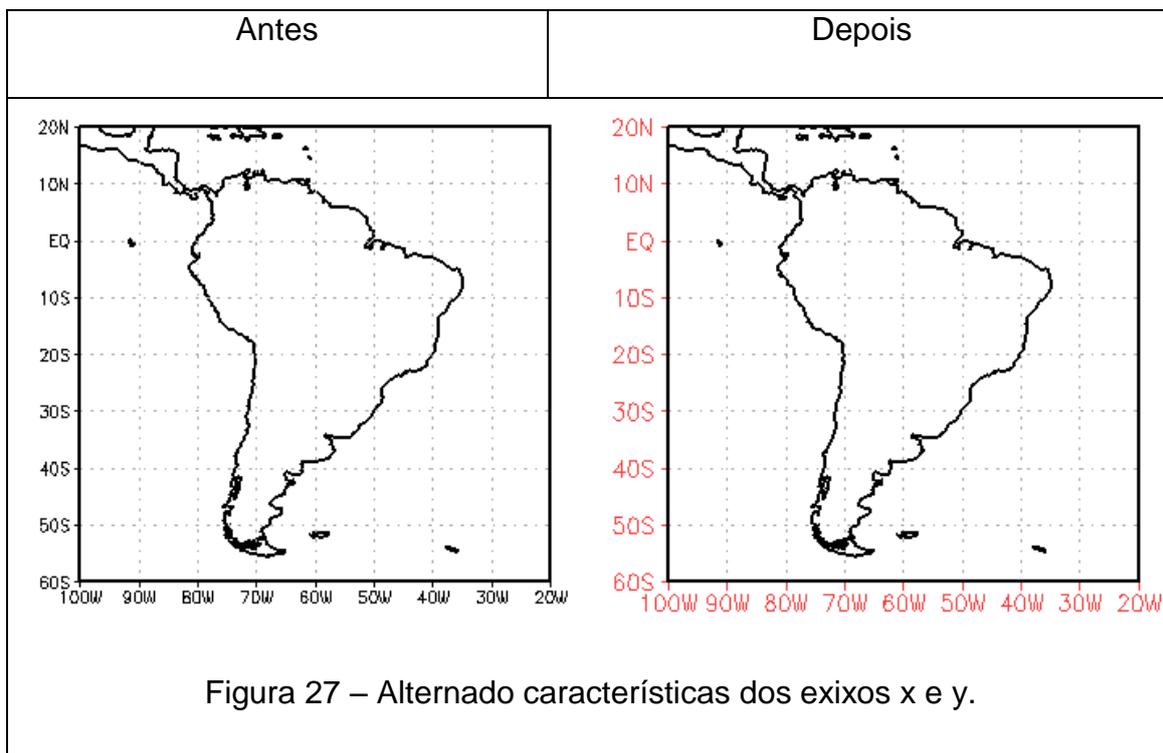
'set lon -100'

'set t 1 20'

'd mag(uwnd,vwnd.2)'
```

**Importante:** O comando anterior manipula o eixo y. Caso a mudança fosse aplicada ao eixo x, deve-se usar o comando 'set vrange2'.

### 34 ALTERANDO CARACTERÍSTICAS DOS EIXOS X E Y



'reinit'

'sdfopen uwnd.mon.mean.nc'

'sdfopen vwnd.mon.mean.nc'

'set xlopts 2 4 0.15' ;\*cor(2) espessura(4) tamanho(0.15) do eixo x

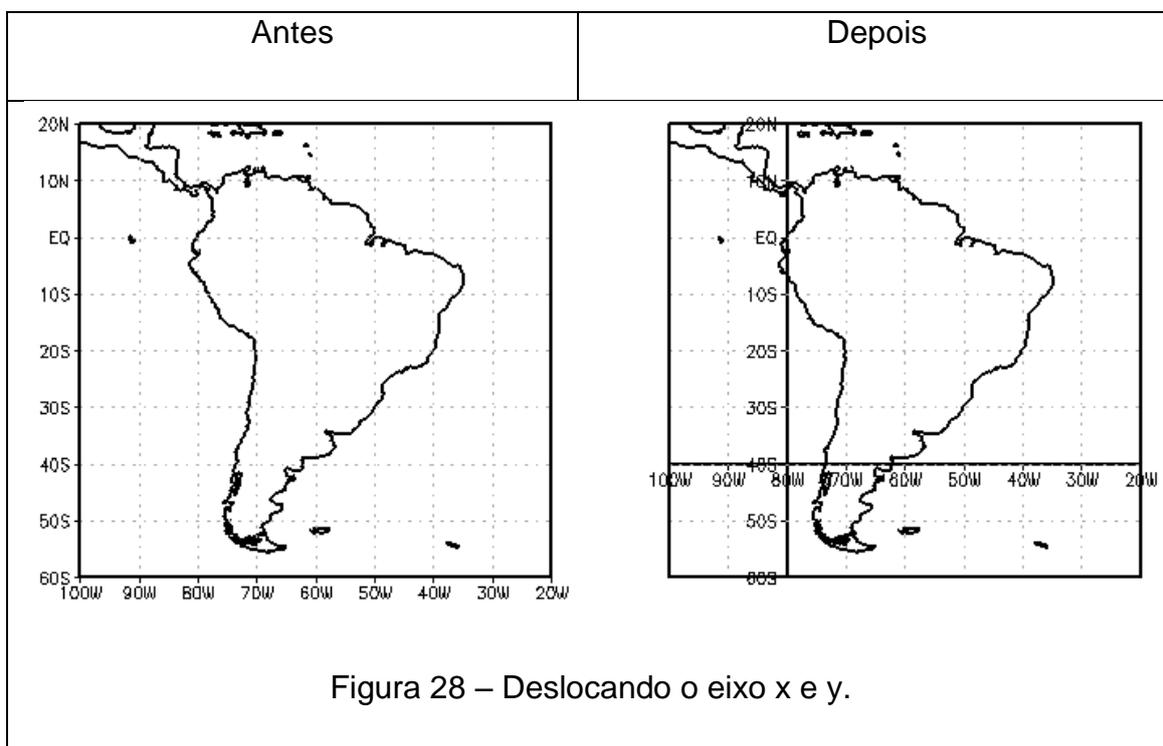
'set ylopts 2 4 0.15' ;\*cor(2) espessura(4) tamanho(0.15) do eixo y

'd mag(uwnd,vwnd.2)'

### 35 DESLOCANDO O EIXO X E/OU Y EM UM GRÁFICO

**Para o eixo x:** Sintaxe: `set xlpós <val1> <val2>`. Onde <val1> corresponde ao valor em que o eixo deve ser deslocado em relação a <val2> que pode ser b (bottom = lado inferior) ou t (top = lado superior).

**Para o eixo y:** Sintaxe: `set ylpós <val1> <val2>`. Onde <val1> corresponde ao valor em que o eixo deve ser deslocado em relação a <val2> que pode ser l (left = lado esquerdo) ou r (right = lado direito).



```
'reinit'
```

```
'sdfopen uwnd.mon.mean.nc'
```

```
'sdfopen vwnd.mon.mean.nc'
```

```
'set xlpós 1 b' ;*desloca o eixo x em 1 em relação a parte inferior (b)
```

```
'set ylpós 1 l' ;*desloca o eixo y em 1 em relação ao lado esquerdo (l)
```

```
'd mag(uwnd,vwnd.2)'
```

### 36 GERANDO GRÁFICO DE CORTE VERTICAL (HOVMÖLLER)

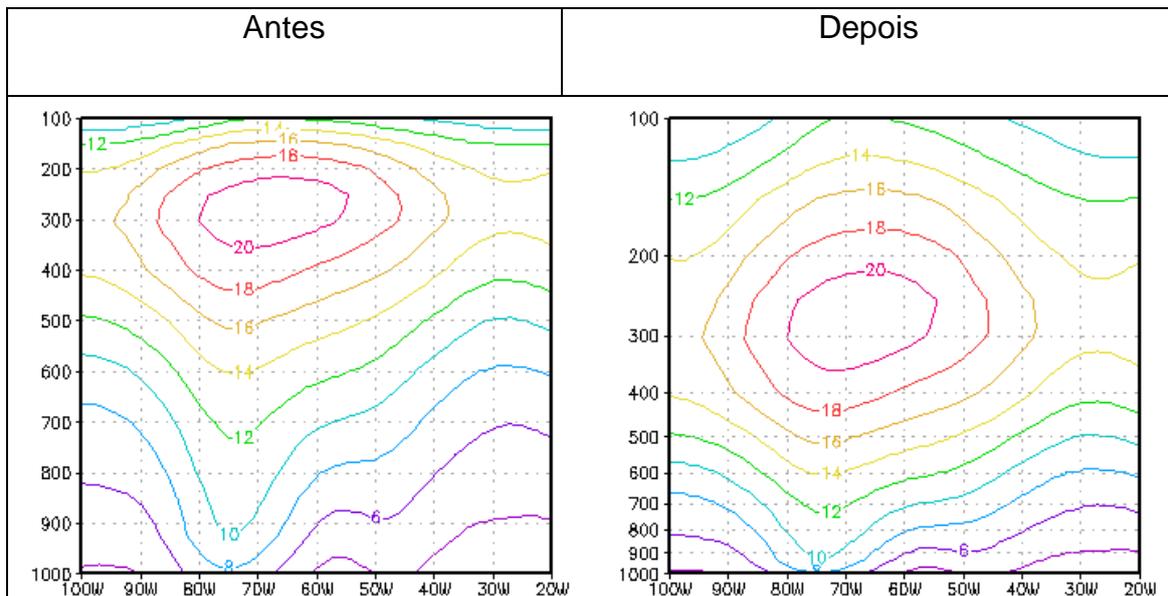


Figura 29 – Gerando gráfico Hovmöller.

```
'reinit'
```

```
'sdfopen uwnd.mon.mean.nc'
```

```
'sdfopen vwnd.mon.mean.nc'
```

```
'set lat -60'
```

```
'set lon -100 -20'
```

```
'set z 1 12'
```

```
'set zlog on' ;*ativa a escala logarítmica
```

```
'd mag(uwnd,vwnd.2)'
```

### 37 OBTENDO AS COORDENADAS X E Y DE UM PONTO

Dado um ponto (latitude e longitude) e desejamos saber os valores em ponto de grade deste ponto. Para tal, basta usar o comando "w2xy". Ele converte pontos globais em coordenadas x e y da página.

Dados: latitude = -10 (10°S) e longitude = -50 (50°W)

No terminal do GrADS digitamos o comando abaixo. Lembrando que a sequência a ser digitada é longitude (x) e depois latitude (y).

```
ga->q w2xy -50 -10
```

Serão retornados os pontos correspondentes a "x" e "y", respectivamente. Estas informações são úteis para criar marcadores em locais específicos do gráfico.

### 38 DESENHANDO UM MARCADOR

**<tipo\_marcador>** = veja figura de marcadores (**Figura 5**)

**<x>** e **<y>** = coordenadas obtidos com o comando “q pos”

**<tamanho>** = 0.1, 0.2, 0.3, ...

#### **Exemplo:**

```
ga->draw mark 5 3.34 6.67 0.15
```

Dessa forma, será criado um marcador do tipo quadrado fechado (5) na posição  $x = 3.34$  e  $y = 6.67$  com o tamanho 0.15. As coordenadas  $x = 3.34$  e  $y = 6.67$  foram obtidas com o comando “q pos”.

## 39 FUNÇÕES INTRÍNSECAS DO GRADS

### 39.1. Capturando uma linha usando a função sublin

**Sintaxe:** `sublin(string,linha)`

Onde: **string** é a string a ser armazenada

**linha** é um valor inteiro que corresponde a linha a ser armazenada

**Exemplo:**

```
'sdfopen uwnd.mon.mean.nc'
```

```
'sdfopen vwnd.mon.mean.nc'
```

```
'set lat -60 20'
```

```
'set lon -100 -20'
```

```
'd mag(uwnd,vwnd.2)'
```

```
'q dims'
```

Ao digitar 'q dims' serão mostradas as informações abaixo:

```
ga-> q dims
```

```
Default file number is: 1 → linha 1
```

```
X is varying Lon = -100 to -20 X = -39 to -7 → linha 2
```

```
Y is varying Lat = -60 to 20 Y = 13 to 45 → linha 3
```

```
Z is fixed Lev = 1000 Z = 1 → linha 4
```

```
T is fixed Time = 00Z01JAN1979 T = 1 → linha 5
```

Capturando as linhas do comando “q dims”

```
linha2 = sublin(result,2) → X is varying Lon = -100 to -20 X = -39 to -7
```

linha3 = sublin(result,3) → Y is varying Lat = -60 to 20 Y = 13 to 45

linha4 = sublin(result,4) → Z is fixed Lev = 1000 Z = 1

linha5 = sublin(result,5) → T is fixed Time = 00Z01JAN1979 T = 1

### 39.2. Capturando uma palavra usando a função subwrđ

#### Sintaxe: subwrđ(string,palavra)

Onde: **string** será a string a ser armazenada

**palavra** é um valor inteiro que corresponde a posição da string a ser armazenada

#### Exemplo:

```
'sdfopen uwnd.mon.mean.nc'
```

```
'sdfopen vwnd.mon.mean.nc'
```

```
'set lat -60 20'
```

```
'set lon -100 -20'
```

```
'd mag(uwnd.2,vwnd.3)'
```

```
'q dims'
```

Ao digitar 'q dims' serão mostradas na janela as informações abaixo:

```
ga-> q dims
```

```
Default file number is: 1 →linha 1
```

```
X is varying Lon = -100 to -20 X = -39 to -7 →linha 2
```

```
Y is varying Lat = -60 to 20 Y = 13 to 45 →linha 3
```

```
Z is fixed Lev = 1000 Z = 1 →linha 4
```

T is fixed Time = 00Z01JAN1979 T = 1 →linha 5

Capturando a linha4 do comando 'q dims'

1 2 3 4 5 6 7 8 9

linha4 = sublin(result,4) → Z is fixed Lev = 1000 Z = 1

Capturando parte de uma linha. Neste caso, a linha capturada foi a linha4.

p4 = subwrd(linha4,6) →observe que p4 armazenou da linha4 na posição 6 que corresponde a 1000. A variável "p4" poderia receber qualquer outro nome.

zz = subwrd(linha4,9) → veja que agora o valor armazenada da linha4 esta na posição 9 que corresponde a 1.

### 39.3. Capturando parte de uma palavra usando a função substr

**Sintaxe: substr(string,início,tamanho)**

Onde:

**string** é a string a ser armazenada

**início** é um valor inteiro que indica a posição onde começa a palavra a ser armazenada

**tamanho** é um valor inteiro que indica quantos caracteres terá a string a ser armazenada

**Exemplo:**

'sdfopen uwnd.mon.mean.nc'

'sdfopen vwnd.mon.mean.nc'

'set lat -60 20'

'set lon -100 -20'

'd mag(uwnd.2,vwnd.3)'

Ao digitar 'q dims' serão mostradas as informações abaixo:

ga-> q dims

Default file number is: 1 →linha 1

X is varying Lon = -100 to -20 X = -39 to -7 →linha 2

Y is varying Lat = -60 to 20 Y = 13 to 45 →linha 3

Z is fixed Lev = 1000 Z = 1 →linha 4

T is fixed Time = 00Z01JAN1979 T = 1 →linha 5

Capturando a linha5 do comando 'q dims'

1 2 3 4 5 6 7 8 9

linha5 = sublin(result,5) → T is fixed Time = 00Z01JAN1979 T = 1

Capturando parte de uma linha. Neste caso, a linha capturada foi a linha5.

x1 = subwrd(linha5,6) →observe que x1 armazenou da linha5 na posição 6 que corresponde a "00Z01JAN1979".

Serão acrescentados espaços em "00z01JAN1979" para melhor visualização da função "substr"

1 2 3 4 5 6 7 8 9 10 11 12

0 0 Z 0 1 J A N 1 9 7 9

hora = substr(x1,1,2) → isso armazenara o valor "00" pois ele começa na posição um e têm dois caracteres (00). Lembre-se, no GrADS a hora apresenta apenas dois caracteres.

dia = substr(x1,4,2) → isso armazenara o valor “01” pois ele começa na posição quatro e têm dois caracteres (01). Lembre-se, no GrADS o dia tem no máximo dois caracteres.

mes = substr(x1,6,3) → isso armazenara o valor “JAN” pois ele começa na posição seis e têm três caracteres (JAN). Lembre-se, no GrADS os meses apresentam 3 caracteres. O idioma dos meses é dado em inglês.

ano = substr(x1,9,4) → isso armazenara o valor “1979” pois ele começa na posição nove e têm quatro caracteres (1979). Lembre-se, no GrADS o ano apresenta 4 caracteres.

### **Conclusão:**

Estas funções servem para automatizar os scripts, armazenando valores, variáveis etc. Isso dependerá das necessidades do usuário.

## 40 DESENHANDO UMA CAIXA NO GRÁFICO

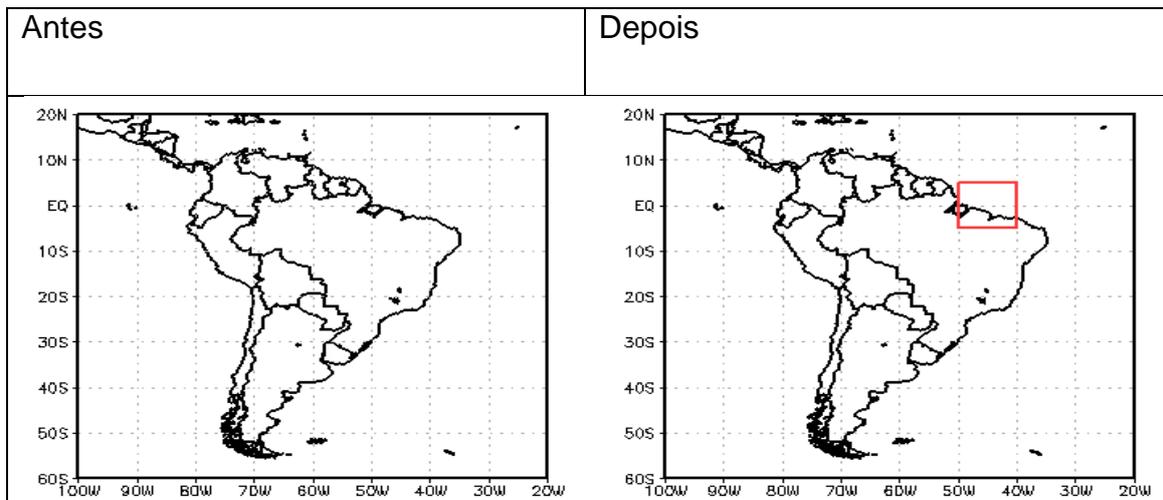


Figura 30 – Desenhando uma caixa no gráfico.

'reinit'

'sdfopen uwnd.mon.mean.nc'

'sdfopen vwnd.mon.mean.nc'

'set lat -60 20'

'set lon -100 -20'

'd mag(uwnd,vwnd.2)'

\*longitude de -50 e latitude -5

'q w2xy -50 -5' ;\*converte longitude (-50) e latitude (-5) para pontos de grade x e y

x1=subwrd(result,3) ; y1=subwrd(result,6)

\*longitude de -40 e latitude 5.

'q w2xy -40 5' ;\*converte longitude (-40) e latitude (5) para pontos de grade x e

y

```
x2=subwrd(result,3) ; y2=subwrd(result,6)
```

```
'set line 2 1 9' ;*cor(2) estilo(1) espessura(9) da caixa
```

```
'draw rec 'x1' 'y1' 'x2' 'y2' ;*desenha a caixa
```

## **41 BARRA DE CORES NO GRADS**

Você precisa ter alguns arquivos para que a barra colorida possa aparecer. São eles: `cbar.gs` (barra quadrada), `cbarn.gs` (barra com ponta nas extremidades) e `cbarc.gs` (barra no canto direito superior).

Para que a barra apareça é necessário que seu gráfico seja do tipo preenchido (`shaded`) ou quando plotamos a magnitude do vento através da opção `vector`. O comando que mostra a barra deve ser executado após a visualização da variável. Veja os exemplos abaixo.

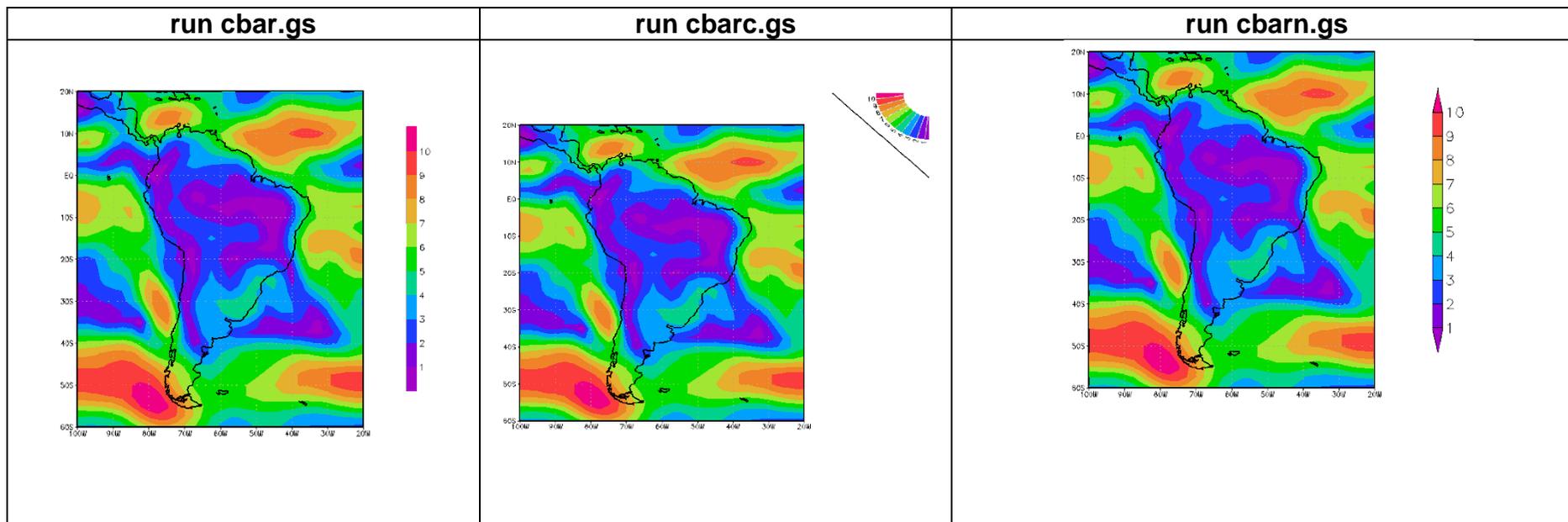


Figura 31 – Legendas do disponíveis no GrADS.

<pre>'reinit' 'sdfopen uwnd.mon.mean.nc' 'sdfopen vwnd.mon.mean.nc' 'set lat -60 20' 'set lon -100 -20' 'set gxout shaded' 'd mag(uwnd,vwnd.2)' 'cbar' ;*desenha a escala de cores</pre>	<pre>'reinit' 'sdfopen uwnd.mon.mean.nc' 'sdfopen vwnd.mon.mean.nc' 'set lat -60 20' 'set lon -100 -20' 'set gxout shaded' 'd mag(uwnd,vwnd.2)' 'cbarc' ;*desenha a escala de cores</pre>	<pre>'reinit' 'sdfopen uwnd.mon.mean.nc' 'sdfopen vwnd.mon.mean.nc' 'set lat -60 20' 'set lon -100 -20' 'set gxout shaded' 'd mag(uwnd,vwnd.2)' 'cbarn' ;*desenha a escala de cores</pre>
--	---	---

## 42 SALVANDO FIGURAS NO GRADS

Após ter atribuído todas as características à sua figura, digite o comando “printim” para salvar as figuras no formato “.gif”, “.png” ou “.jpg”. Este comando deve ser a última linha do seu “.gs”. Certifique-se que você tem o arquivo “gxgif” no diretório onde estão as bibliotecas do GrADS para salvar no formato “.gif”.

### Exemplo:

```
'reinit'
```

```
'sdfopen uwnd.mon.mean.nc'
```

```
'set lat -60 20'
```

```
'set lon -100 -20'
```

```
'set gxout shaded'
```

```
'd uwnd'
```

```
'cbarn'
```

```
'printim vento.gif' ;*salva a figura com o nome “vento” no formato “.gif”
```

### 43 CRIANDO UM ARQUIVO “.GS” NO GRADS

Para gerar um arquivo “.gs” abra seu editor de texto preferido e salve o arquivo com um nome qualquer com a extensão “.gs”.

**Importante: Não se esqueça de colocar os comandos do GrADS entre aspas no seu script, caso contrário seu “.gs” terá problemas e não será executado.**

**Exemplo de um arquivo .gs:**

```
'reinit' ;*reinicia o grads toda vez que executa o arquivo “.gs”
```

```
'open 250304.ctl' ;*abre o arquivo 250304.ctl
```

```
'set map 1 1 6' ;*fixa cor (1), estilo (1) e espessura (6) ao mapa
```

```
'set lat -45 -10' ;*fixa a latitude
```

```
'set lon -100 -10' ;*fixa a longitude
```

```
'set lev 925' ;*fixa o nível desejado
```

```
'set grads off' ;*desabilita a propaganda do GrADS
```

```
'set grid off' ;*desabilita a grade do GrADS
```

```
'set gxout contour' ;*seleciona o gráfico do tipo contorno ou isolinhas
```

```
'set cint 2' ;*especifica o intervalo entre as isolinhas
```

```
'set cthick 6' ;*especifica a espessura da isolinha
```

```
'd temp' ;*visualiza a variável desejada (Temperatura)
```

```
'draw title Temperatura em 925 hPa ;*escreve o título no gráfico
```

```
'printim temperatura.png' ;*salva o arquivo com o formato “.gif”
```

#### 44 EXECUTANDO ARQUIVO GS

O GrADS deve estar aberto para executar o arquivo “.gs”, para isso basta digitar:

```
ga-> run nomedoarquivo.gs ou simplesmente ga-> nomedoarquivo.gs
```

O arquivo pode ser executado sem colocar o “run” na frente do “.gs”

Pode-se executar o arquivo .gs de outra forma. Por exemplo:

```
ga->gradsnc -blc “seu-script.gs”
```

Isso abre o GrADS sem tela gráfica (b), no modo lanscape (l) via comando (c).

## 45 PROGRAMAS ADICIONAIS

### 45.1. Usando o script grib2ctl.pl

Caso seu arquivo “.grb” não possua seu respectivo “.ctl” o script “grib2ctl.pl” facilita sua criação. Abra um terminal no Linux onde estão seus dados “.grb” e use o script para criar o “.ctl”. Este arquivo pode ser baixado do seguinte endereço: <http://www.cpc.noaa.gov/products/wesley/grib2ctl.html>.

Estes comandos devem ser feitos no terminal do Linux:

1. grib2ctl.pl seu.arquivo.grib > seu.arquivo.de.saida.ctl
2. gribmap -i seu.arquivo.de.saida.ctl -0

Depois destes passos serão gerados os arquivos “.idx” e “.ctl”. Dessa forma, você poderá abrir seu arquivo “.ctl” e gerar suas figuras.

### 45.2. Usando o script lats4d.gs

O script “lats4d.gs” é muito útil quando se deseja "cortar" os dados para uma determinada data, área, nível, latitude, longitude etc. Isso ficará a critério do usuário fazer as devidas modificações.

**Exemplo 1:** Neste caso será gerado um arquivo “.grb” (-format grads\_grib) com seu respectivo “.ctl”

Com o GrADS aberto, basta digitar estes comandos:

```
ga->lats4d.gs -i arquivo.ctl ou .nc -o prefixo.saida -format grads grib -time  
01jan1989 31dec2000 -levs 1000 925 850 -lat -60 -40 -lon -100 -20 -q
```

Veja que você pode usar um arquivo “.ctl” ou “.nc” como entrada e como saída, ainda após a opção “-o” não existe mais nada apenas o nome dado pelo usuário (prefixo.saida) e dependendo do formato (opção “-format”) selecionado

(seja ele grads\_grib ou stream) a extensão será adicionada a este nome. A opção -q no final diz ao GrADS para sair ao terminar as tarefas.

Você pode selecionar apenas uma data, um nível, todo o globo, as opções dependerão de cada usuário. O seu arquivo deve conter as informações que você deseja extrair.

**Exemplo 2:** Neste caso será gerado apenas um arquivo “.bin” (-format stream). Este formato é muito utilizado quando se deseja usar o Fortran.

```
ga->lats4d.gs -i arquivoctl ou .nc -o prefixo.saida -format stream -time 01jan1989 -levs 850 -lat -60 -40 -lon -100 -20 -q
```

### 45.3. Procedimentos para instalar EOF no GrADS

\*\*\*\*\*

**REQUERIMENTO: Fortran 90 e ser root na sua máquina**

**SÓ FUNCIONA NO GrADS 1.9**

\*\*\*\*\*

- 1 - Neste endereço <http://www.atmos.ucla.edu/~munnich/Grads/EOF/> faça o download do arquivo para sua plataforma Linux.
- 2 - Crie o diretório "eof\_grads" em "/usr/local/lib/grads/" e copie o arquivo "eofudf\_linux.tar.gz" para o diretório criado.
- 3 - Ir para o diretório "eof\_grads" e descompactar o arquivo fazendo "tar -xvf eofudf\_linux.tar.gz". Com isso serão criados os arquivos: "eof.gs", "eofudf", "libcxa.so.3" e "readme.html". Copie o arquivo "eof.gs" para "/usr/local/lib/grads".
- 4 - Faça o download do código fonte (eofgrads\_src.tar.gz) que se encontra no endereço do item 1 deste tutorial para "/usr/local/lib/grads/eof\_grads".

Descompacte este arquivo fazendo "tar -xvf eofgrads\_src.tar.gz". Com isso, surge o diretório "eofgrads\_v0155". Dentro dele estão os arquivos que serão compilados.

5 - Entre no diretório "eofgrads\_v0155/src" e edite o Makefile de acordo com o compilador Fortran 90 que você usa na sua máquina. As alinhas abaixo são do arquivo "Makefile".

```
# defaults
```

```
FC=gfortran --> mude para o nome do compilador Fortran 90 que você usa
```

```
FFLAGS=-O Ma
```

```
LDFLAGS=-O
```

```
# double precision
```

```
#FFLAGS=-O -N113
```

```
#LDFLAGS=-O -N113
```

```
#Intel f90
```

```
#FC=ifc
```

```
#AIX
```

```
#FC=xlf
```

```
#FFLAGS=-O -w -axW # Intel ifc + P4 processor --> comente aqui  
colocadando #
```

```
#LDFLAGS=-O -w -axW # Intel ifc + P4 processor --> comente aqui  
colocadando #
```

```
eofudf: eofudf.o eispacks.o
```

```
$(FC) $(LDFLAGS) -o eofudf eofudf.o eispacks.o
```

```
jeofudf: jeofudf.o eispacks.o
```

```
$(FC) $(LDFLAGS) -o jeofudf jeofudf.o eispacks.o
```

```
clean:
```

```
rm -f eofudf eofudf.o *.mod eispacks.o *.d
```

```
.f.o:
```

```
$(FC) $(FFLAGS) -c $<
```

6 - Depois faça "make clean" e em seguida "make". Tudo isso e feito dentro de "src". Com estes passos, serão gerados os arquivos: "eispacks.o", "eofudf", "eofudf.o", "sizes.mod" e "udf.mod".

7 - Dentro de "/usr/local/lib/grads/" deve ter um diretório chamado "udf". Se ele não existir, faça "mkdir udf". Coloque o arquivo "eofudf" que esta em "/usr/local/lib/grads/eof\_gs/eofgrads\_v0155/src" dentro de "/usr/local/lib/grads/udf".

8 - Dentro de "/usr/local/lib/grads" crie o arquivo "udft.txt" e edite-o colando as seguintes linhas:

```
eofudf 2 6 expr value value value value char
```

```
sequential
```

```
/usr/local/lib/grads/udf/eofudf
```

```
eofudf.in
```

```
eofudf.out
```

9 - Ir para o seu home e faça "ls -a" e editar seu arquivo ".bashrc" e colar as seguintes linhas.

```
export GAUDFT=/usr/local/lib/grads/
```

```
export GAUDFT=/usr/local/lib/grads/udft.txt
```

10 - Ir para "/usr/local/lib/grads/eof\_grads" e copiar o arquivo "libcxa.so.3" para "/usr/lib/".

11 - Com estes passos você já pode gerar seus gráficos de EOF usando o GrADS.

12 - Para fazer os gráficos execute o GrADS e abra o arquivo que desejar para fazer a EOF

### **Exemplo:**

```
'gradsnc' ;*abre o grads
```

```
'sdfopen uwnd.nc' ;*abre o arquivo ".nc"
```

```
'set t 1 100' ;*fixa 100 tempos (fica ao critério do usuário)
```

```
'set lat -40 10' ;*fixa latitude
```

```
'set lon 270 330' ;*fixa longitude
```

```
'eof -neof 12 -npc 100 -tinc 1 -nper 70 -o saida uwnd' ;*calcula a EOF
```

Onde:

-neof = número de EOFs (default = 12)

-npc = número PCs (default = -neof)

-tinc = incremento para fazer o cálculo (default = 1)

-nper = 70 -> porcentagem de dados em cada ponto de grade para fazer a EOF

-o = nome do arquivo de saída (fica ao critério do usuário)

uwnd = nome da variável do arquivo (.ctl ou .nc)

13 - Se tudo ocorrer bem aparecerão as linhas abaixo.

ga-> eof -neof 12 -npc 100 -tinc 1 -nper 70 -o saida uwnd

Writing data to transfer file...

Executing eofudf binary ...

Number of EOFs: 12

Number of PCs: 12

Required % of defined data: 70

Output files prefix: saida\_

Space dim: 1089

Time dim: 100

Max: 24.56000

Min: -18.69000

Usable space points: 1089 of 1089

Usable time slices: 100 of 100

No undefs in packed data.

Computing covariance matrix...

Total variance: 13673.64

Computing eigenvectors...

Total variance: 13673.64

No. E.value % Var

1	2270.	16.602
2	1419.	10.376
3	1165.	8.521
4	944.1	6.905
5	798.9	5.843
6	703.5	5.145
7	529.9	3.875
8	506.5	3.705
9	437.5	3.200
10	387.5	2.834
11	357.3	2.613
12	349.9	2.559

Writing EOFs...

Writing PCs...

Writing time info to CTLs...

14 - O arquivo "saida\_eof.ctl" contém as PC's no espaço e o arquivo "saida\_eval.txt" estão as porcentagens das PC's que foram selecionadas.

15 - A série temporal de cada PC pode ser vista no arquivo "saida\_pc.ctf". Observe que cada nível "z" corresponde a cada PC. Para visualizar a primeira PC basta fazer no grads.

```
'set z 1' ;* se fosse a segunda, basta fazer 'set z 2' ... 'set z 12'
```

```
'set t 1 100'
```

```
'd pc'
```

#### 45.4. Instalando o regrid2 no GrADS

O regrid2 serve para interpolar os campos meteorológicos. Abaixo são mostrados os procedimentos para sua instalação.

1 – Copiar o arquivo “regrid2.tar.gz” disponível em <http://ftp.kumamoto-u.ac.jp/pub/.2/graphic/grads/sprite/udf/> para “/usr/local/lib/grads”.

2 – Dentro de “/usr/local/lib/grads” fazer “tar -zxvf regrid2.tar.gz”. Este procedimento cria o diretório “udf” em “/usr/local/lib/grads”. Dentro deste diretório encontra-se o manual do regrid2.

3 – Entrar em “udf” e ir para “regrid2”.

4 – Edite o arquivo “Makefile”. As linhas abaixo são do arquivo Makefile. Modifique as linhas em vermelho colocando o caminho onde esta o diretório grads e o tipo de compilador usado na máquina (para este exemplo esta sendo usado o gfortran).

```
SHELL=/bin/sh
```

```
#
```

```
#
```

```
#
```

**REGRID\_DIR = /usr/local/lib/grads**

SRCS= regrid2.f

OBJS= regrid2.o

#

**CFT= g77**

**LDR= g77**

**CFT= g77**

**LDR= g77**

**CFT= g77**

**LDR= g77**

LDRFLAGS=-o

#LDRFLAGS= -32 -o

EXE= \$(REGRID\_DIR)/udf/regrid2.exe

EXE= ../regrid2.exe

#FFLAGS= -c -O -32

FFLAGS= -c -O

\$(EXE): \$(OBJS)

\$(LDR) \$(LDRFLAGS) \$(EXE) \$(OBJS)

.f.o:

\$(CFT) \$(FFLAGS) \$\*.f

del\_lists:

```
-rm -f $(LISTS)
```

clean:

```
-rm -f $(OBS)
```

5 – Digitar no terminal do Linux “make clean” e depois “make -f Makefile”.

6 – Ir para seu home e editar seu arquivo “.bashrc”. Para ver este arquivo basta digitar no Linux “ls -a”. Com esta opção os arquivos ocultos (arquivos que possuem um ponto na frente do nome do arquivo) serão mostrados. Adicione a linha abaixo no seu “.bashrc”.

```
export GAUDFT=/usr/local/lib/grads/udf/regrid2
```

7 – Fazer source “.bashrc” para atualizar seu “.bashrc”.

8 – Abra o GrADS e digite “q udf”. Se tudo ocorrer bem será mostrada a linha abaixo, isso significa que o regrid foi instalado.

```
regrid2 Args: 1 8 Exec: /usr/local/lib/grads/udf/regrid2.exe
```

9 – Com isso, você será capaz de interpolar seus campos.

**Exemplo:**

```
'reinit'
```

```
'sdfopen air.nc'
```

```
'd regrid2(air,1.5,1.5,bl) ;*'air" é a variável a ser interpolada, “1.5” corresponde a resolução espacial (em longitude e latitude) que será interpolada e “bl” é o tipo de interpolação.
```

## 46 LINGUAGEM DE PROGRAMAÇÃO NO GRADS

### 46.1. Operadores

Tabela 1 – Operadores usados no GrADS

<b>Símbolo</b>	<b>Definição</b>
	Operador lógico OU
!	Unary not
%	Concatenação
!=	Diferente
>=	Maior ou igual a
<=	Menor ou igual
-	Subtração
/	Divisão
&	Operador lógico E
-	Unary minus
=	Igual
>	Maior que
<	Menor que
+	Soma
*	Multiplicação

## 46.2. Fluxo de controle

### 46.2.1. Comando while

O controle de fluxo “WHILE” tem diversas aplicações. Dependerá das necessidades do usuário empregá-lo para suas necessidades. Abaixo, são mostrados alguns exemplos.

**Exemplo:** Usando o "while" para varrer 10 tempos do arquivo.

```
'reinit'
```

```
'sdfopen ATSM.nc'
```

```
t1 = 1 ;* Tempo inicial que será definido pelo usuário
```

```
t2 = 10 ;* Tempo final que será definido pelo usuário
```

```
while(t1<=t2)
```

```
'set t 't1 ;*varre os tempos de t1 (1) até t2 (10)
```

```
'd anom' ;*visualiza a variável (anom)
```

```
pull enter ;*espera por um comando do usuário (PRESSIONE ENTER)
```

```
'c' ;*limpa a tela para mostrar o próximo tempo
```

```
t1 = t1 + 1
```

```
endwhile
```

### 46.2.2. Comando if

O controle de fluxo “IF” tem por finalidade substituir variáveis como também servir de condição para determinadas tarefas, dentre outras possibilidades que dependerá das necessidades do usuário. Abaixo, são mostrados alguns exemplos.

**Exemplo 1:** O comando IF usado aqui, substitui a string mes (JAN, FEB, ... , DEC) do GrADS (lembrando que os meses no GrADS estão em inglês) para valor numérico (JAN = 01, FEB = 02 , ... , DEC = 12).

```
'reinit'  
  
'sdfopen ATSM.nc'  
  
t1 = 1 ;*Tempo inicial  
  
t2 = 50 ;*Tempo final  
  
while(t1<=t2)  
  
'set t 't1 ;*varre os tempos de t1 (1) até t2 (10)  
  
'q dims' ;*fornece detalhes sobre o arquivo aberto  
  
linha5 = sublin(result,5) ;*captura a linha 5 do comando "q dims"  
  
data = subwrld(linha5,6) ;*captura a string "00Z22FEB1985" do comando "q  
dims"  
  
dia = substr(data,4,2) ;*captura o dia da string "00Z22FEB1985"  
  
mes = substr(data,6,3) ;*captura o mês da string "00Z22FEB1985"  
  
ano = substr(data,9,4) ;*captura o ano da string "00Z22FEB1985"  
  
hora = substr(data,1,2) ;*captura a hora da string "00Z22FEB1985"  
  
if(mes=JAN);mesc=01;endif  
  
if(mes=FEB);mesc=02;endif  
  
if(mes=MAR);mesc=03;endif  
  
if(mes=APR);mesc=04;endif
```

```
if(mes=MAY);mesc=05;endif
```

```
if(mes=JUN);mesc=06;endif
```

```
if(mes=JUL);mesc=07;endif
```

```
if(mes=AUG);mesc=08;endif
```

```
if(mes=SEP);mesc=09;endif
```

```
if(mes=OCT);mesc=10;endif
```

```
if(mes=NOV);mesc=11;endif
```

```
if(mes=DEC);mesc=12;endif
```

```
'd anom' ;*visualiza a variável (anom)
```

```
'draw title Anomalia da TSM no dia 'dia'/'mesc'/'ano
```

```
pull enter ;*espera por um comando do usuário (PRESSIONE ENTER)
```

```
'c' ;*limpa a tela
```

```
t1 = t1 + 1
```

```
endwhile
```

**Exemplo 2:** Neste exemplo são mostradas as aplicações dos comandos WHILE e IF. Serão abertos vários arquivos (1985 a 1990) e em seguida 5 tempos de cada arquivo aberto são visualizados e após a visualização dos tempos o mesmo é fechado para a abertura do próximo.

```
'reinit'
```

```
'set display color white'
```

```
'c'
```

```

f1 = 1985

f2 = 1990

while(f1<=f2)

'sdfopen ATSM.'f1'.nc' ;*abre a seqüência de arquivos

t1 = 1 ;* Tempo inicial

t2 = 5 ;* Tempo final

*varre os tempos de t1 (1) até t2 (5)

while(t1<=t2)

'set t 't1

'set lat -60 10'

'set lon -90 -20'

'q dims' ;*fornece detalhes do arquivo aberto

linha5 = sublin(result,5) ;*captura a linha 5 do comando "q dims"

data = subwrd(linha5,6) ;*captura a string "00Z22FEB1985" do comando "q
dims"

dia = substr(data,4,2) ;*captura o dia da string "00Z22FEB1985"

mes = substr(data,6,3) ;*captura o mês da string "00Z22FEB1985"

ano = substr(data,9,4) ;*captura o ano da string "00Z22FEB1985"

hora = substr(data,1,2) ;*captura a hora da string "00Z22FEB1985"

if(mes=JAN);mesc=01;endif

```

```
if(mes=FEB);mesc=02;endif
if(mes=MAR);mesc=03;endif
if(mes=APR);mesc=04;endif
if(mes=MAY);mesc=05;endif
if(mes=JUN);mesc=06;endif
if(mes=JUL);mesc=07;endif
if(mes=AUG);mesc=08;endif
if(mes=SEP);mesc=09;endif
if(mes=OCT);mesc=10;endif
if(mes=NOV);mesc=11;endif
if(mes=DEC);mesc=12;endif

'd anom' ;*visualiza a variável (anom)

'draw title Anomalia da TSM no dia 'dia'/'mesc'/'ano' ;*escreve o título no gráfico

pull enter ;*espera por um comando do usuário (PRESSIONE ENTER)

'c'

t1 = t1 + 1

endwhile

'reinit' ;*fecha o arquivo

'set display color white'

'c'
```

```
f1 = f1 + 1
```

```
endwhile
```

## 47 SALVANDO ARQUIVO TEXTO NO GRADS

Esta opção é útil quando se deseja salvar uma determinada série de dados para ser usada em outro pacote gráfico para visualização (SURFER, MATLAB, STATISTICA etc.) ou até mesmo para manipulação usando a Linguagem FORTRAN. Veja o exemplo abaixo.

```
'reinit'
```

```
'sdfopen ATSM.nc'
```

```
t1 = 1 ;* Tempo inicial
```

```
t2 = 10 ;* Tempo final
```

```
*Enquanto (while) t1 (1) for menor ou igual a t2 (10)
```

```
while(t1<=t2)
```

```
'set t 't1' ;*varre os tempos de t1 (1) até t2 (10)
```

```
'set lat 0' ;* fixa a latitude no Equador
```

```
'set lon -20' ;* fixa a longitude em 120W (longitudes oeste = negativas)
```

```
'q dims' ;*fornece detalhes sobre o arquivo aberto
```

```
linha5 = sublin(result,5) ;*captura a linha 5 do comando "q dims"
```

```
data = subwrd(linha5,6) ;*captura a string "00Z22FEB1985" do comando "q  
dims"
```

```
dia = substr(data,4,2) ;*captura o dia da string "00Z22FEB1985"
```

```
mes = substr(data,6,3) ;*captura o mes da string "00Z22FEB1985"
```

```
ano = substr(data,9,4) ;*captura o ano da string "00Z22FEB1985"
```

```
hora = substr(data,1,2) ;*captura a hora da string "00Z22FEB1985"
```

```
if(mes=JAN);mesc=01;endif
```

```
if(mes=FEB);mesc=02;endif
```

```
if(mes=MAR);mesc=03;endif
```

```
if(mes=APR);mesc=04;endif
```

```
if(mes=MAY);mesc=05;endif
```

```
if(mes=JUN);mesc=06;endif
```

```
if(mes=JUL);mesc=07;endif
```

```
if(mes=AUG);mesc=08;endif
```

```
if(mes=SEP);mesc=09;endif
```

```
if(mes=OCT);mesc=10;endif
```

```
if(mes=NOV);mesc=11;endif
```

```
if(mes=DEC);mesc=12;endif
```

```
'd anom' ;*visualiza a variável "anom"
```

Observe que foi selecionado apenas um ponto (set lat 0 e set lon -20). E o tempo esta variando. Logo, o resultado será uma série temporal.

```
Posições          1      2  3  4
```

```
atsm = subwrd(result,4) ;*captura a posição 4 de "Result value = 0.4"
```

```
*será escrito no arquivo de saída "val_tsm.dat":
```

```
*dia = coluna 1, mês = coluna 2, ano = coluna 3 e Anomalia da TSM = coluna 4
```

```
saida = write(val_tsm.dat,dia' 'mesc' 'ano' 'atasm)
```

```
say dia' 'mesc' 'ano' 'atasm    ;*mostra na tela o que será guardado no  
arquivo “.dat”
```

```
t1 = t1 + 1
```

```
endwhile
```

## 48 APLICAÇÕES DIVERSAS

### 48.1. Gráfico de linha de uma área

```
'reinit'
```

```
'sdfopen air.1980.nc'
```

```
'set lat 0' ;*tem que fixar uma latitude qualquer. Ela não terá nenhuma influência na figura
```

```
'set lon -60' ;*tem que fixar uma longitude qualquer. Ela não terá nenhuma influência na figura
```

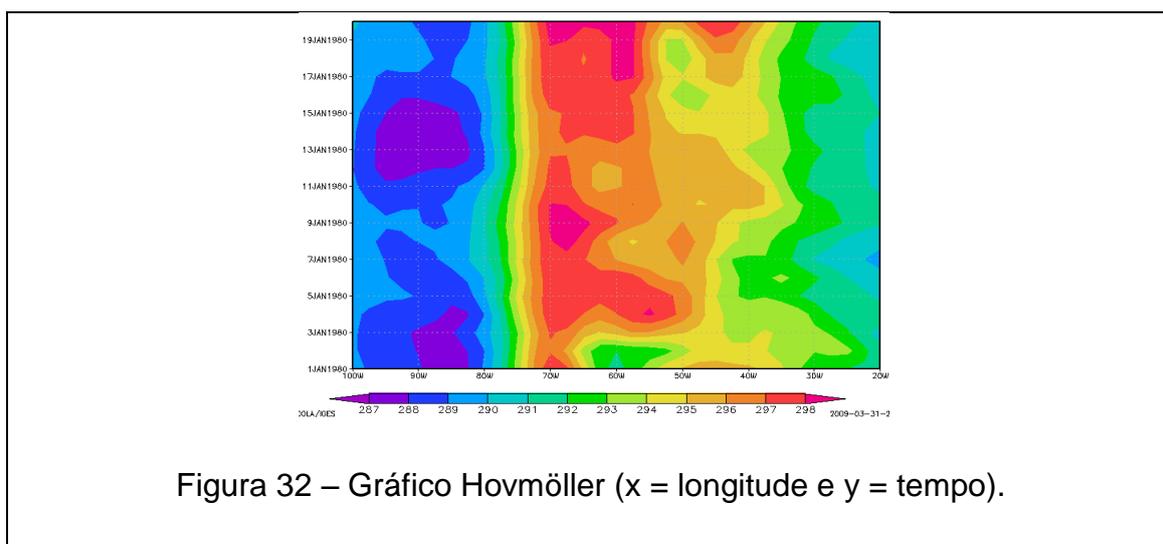
```
'set t 1 20' ;*tempo que será visualizado
```

\*é necessário fixar um ponto (lat,lon) para a função “tloop” funcionar. A função “aave” realiza o cálculo de média e retorna apenas um valor

```
'd tloop(aave(air,lon=-60,lon=-55,lat=-5,lat=-2.5))'
```

```
'printim graf_linha.png' ;*salva a figura com o nome “graf_linha” no formato “.gif”
```

### 48.2. Gráfico Hovmöller (x = longitude e y = tempo)



```
'reinit'
```

```
'sdfopen air.1980.nc'
```

```
'set lat 0' ;*tem que fixar uma latitude qualquer. Ela não terá nenhuma  
influência na figura
```

```
'set lon -100 -20' ;*faixa de longitude desejada
```

```
'set lev 925' ;*fixa o nível desejado
```

```
'set t 1 20' ;*tempo que será visualizado
```

```
'set gxout shaded' ;*gráfico do tipo preenchido
```

```
'd ave(air,lat=-20,lat=-10)' ;*faz média (ave) na latitude de -20 a -10
```

```
'cbarn' ;*mostra a escala de cores
```

```
'printim hov_xlon_ytempo.gif' ;*salva a figura no formato “.png”
```

### 48.3. Gráfico Hovmöller (x = latitude e y = altura)

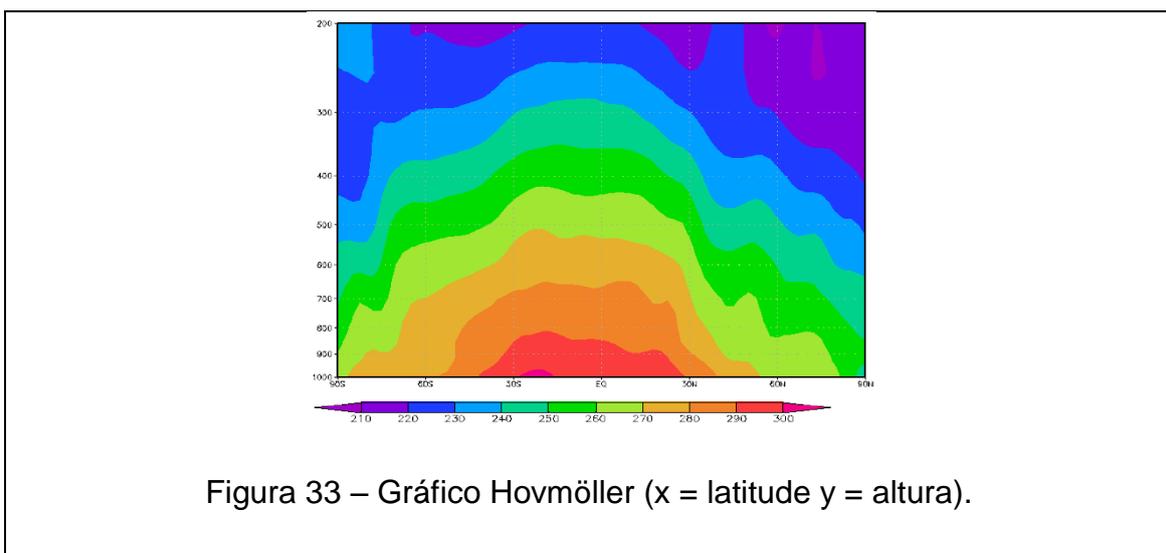


Figura 33 – Gráfico Hovmöller (x = latitude y = altura).

```
'reinit'
```

```
'set display color white'
```

```
'c'
```

```
'sdfopen air.nc'
```

```
'set t 20' ;*tempo que será feito o gráfico
```

```
'set lon -60' ;*tem que fixar uma latitude qualquer. Ela não terá nenhuma  
influência na figura
```

```
'set lev 1000 200' ;*nível desejado
```

```
'set grads off'
```

```
'set gxout shaded'
```

```
'set zlog on' ;*habilita escala logarítmica
```

```
'd ave(air,lon=-60,lon=-50)' ;*faz média na longitude
```

```
'cbarn'
```

#### 48.4. Gráfico Hovmöller (x = tempo e y = altura)

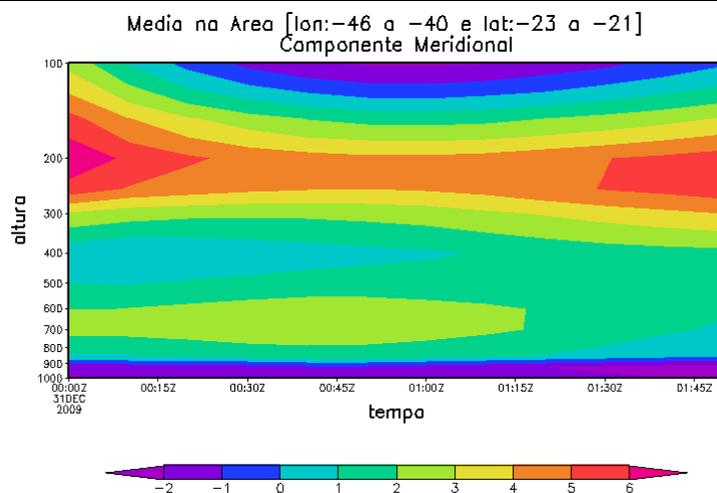


Figura 34 – Gráfico Hovmöller (x = tempo y = altura).

```
'reinit'
```

```
'set display color white'
```

```
'c'
```

```
'open se_g1.ctl'
```

```
'set gxout shaded'
```

```
'set t 1 12 '
```

```
'set z 1 12'
```

```
'define vave = aave(v,lon=-46,lon=-40,lat=-23,lat=-21)'
```

```
'set x 1'
```

```
'set y 1'
```

```
'set zlog on'
```

```
'd vave'
```

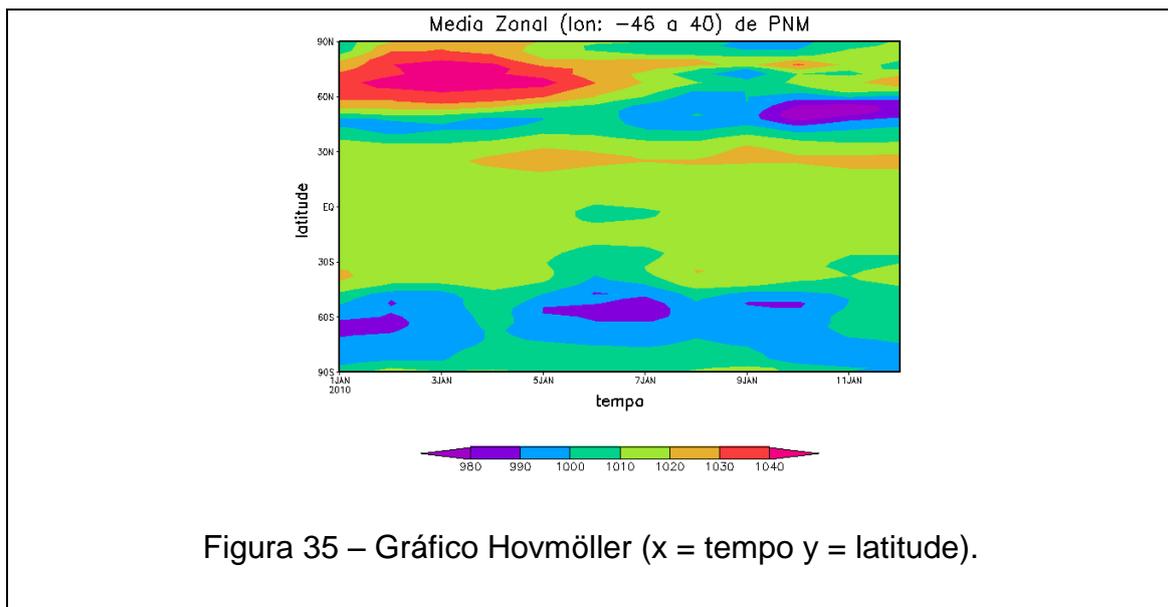
```
'cbarn.gs'
```

```
'draw ylab altura'
```

```
'draw xlab tempo'
```

```
'draw title Media na Area [lon:-46 a -40 e lat:-23 a -21] \ Componente  
Meridional'
```

#### 48.5. Gráfico Hovmöller (x = tempo e y = latitude)



```
'sdfopen slp.2010.nc'
```

```
'set gxout shaded'
```

```
'set t 1 12 '
```

```
'set x 1'
```

```
'set xyrev on'
```

```
'd (ave(slp,lon=-46,lon=-40))/100 '  
  
'cbarn.gs'  
  
'draw ylab latitude '  
  
'draw xlab tempo'  
  
'draw title Media Zonal (lon: -46 a 40) de PNM'
```

#### 48.6. Mascarando o continente ou oceano

Para plotar apenas os valores sobre o continente ou oceano, será usado o `basemap.gs` que esta disponível em <http://grads.iges.org/grads/gadoc/library.html>.

**Importante:** Os arquivos: `lpoly_mres.asc`, `lpoly_US.asc`, `lpoly_US.asc.part`, `lpoly_hires.asc`, `lpoly_lowres.asc` e `basemap.gs` devem estar no mesmo diretório onde se encontra o seu “.gs”, pois do contrário ele não funcionará.

**Exemplo 1:** Mascarando apenas o continente

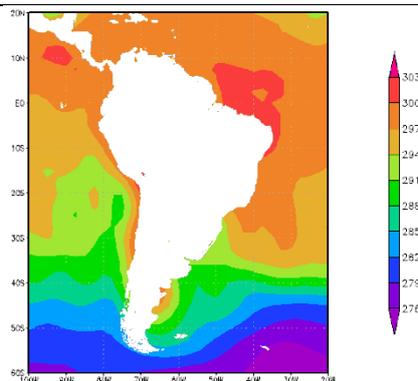


Figura 36 – Mascarando o continente.

```
'reinit'  
  
'sdfopen air.1980.nc'  
  
'set lat -60 20'  
  
'set lon -100 -20'  
  
'set t 1'  
  
'set gxout shaded'  
  
'd air'  
  
'basemap L 0 0 M'  
  
'cbarn'
```

O argumento L = diz para mascarar o continente. Se fosse para mascarar o oceano, mudar de “L” para “O”. O segundo argumento refere-se à cor do preenchimento (ver figura de cores). O terceiro argumento define uma cor para o contorno do preenchimento (figura de cores) e o quarto argumento, desenha ou não desenha os Estados e os países e pode assumir os valores: M, L ou H. Neste caso, mascaramos o continente (L). Definimos a cor branca tanto para o preenchimento (0) como para o contorno (0) e o último argumento (M) não interfere no gráfico.

## Exemplo 2: Mascarando apenas o oceano

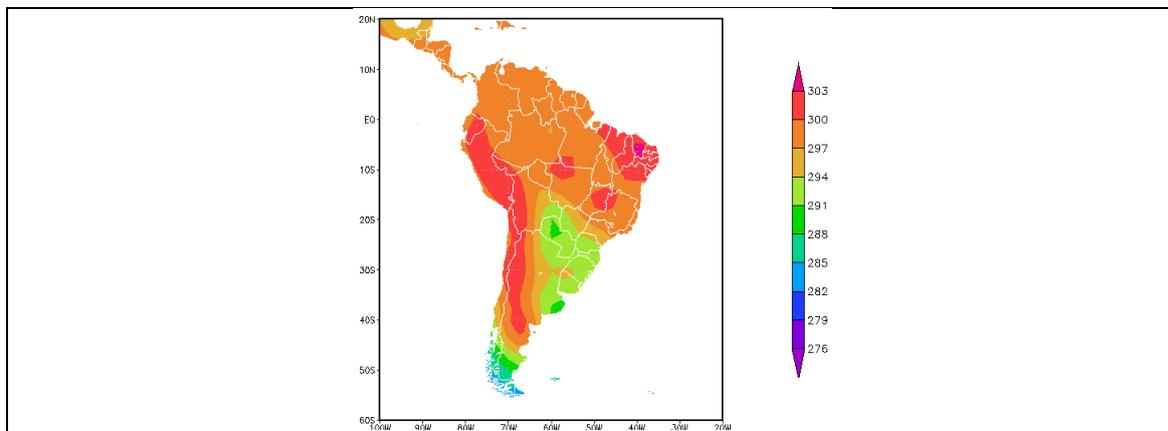


Figura 37 – Mascarando o oceano.

```
'reinit'
```

```
'set display color white'
```

```
'c'
```

```
'sdfopen air.1980.nc'
```

```
'set lat -60 20'
```

```
'set lon -100 -20'
```

```
'set grads off'
```

```
'set gxout shaded'
```

```
'set mpdset brmap' ;*divide os Estados do Brasil e o países
```

```
'd air'
```

```
'basemap O O O L'
```

```
'cbarn'
```

Foi mascarado apenas o oceano (O), seleccionou-se a cor branca para representar o oceano (0) e o contorno do continente (0).

#### 48.7. Criando escala de cores personalizada

Este “.gs” pode ser encontrado no seguinte endereço <http://kodama.fubuki.info/wiki/wiki.cgi/GrADS/script?lang=en>.

##### Exemplo:

```
'reinit'  
  
'set display color white'  
  
'c'  
  
'sdfopen air.1980.nc'  
  
'set lat -60 20'  
  
'set lon -100 -20'  
  
'set grads off'  
  
'color -gxout shaded -kind white->gray->black -var air-273.15 5 30 5'  
  
'd air-273.15'  
  
'cbarn'
```

O script color.gs cria uma escala de cores definida pelo usuário. Dentro deste script estão as cores disponíveis. Fica ao critério do usuário escolher a melhor cor para o seu gráfico. Veja que white, gray e black são cores que estão disponíveis dentro do script. A variável usada foi a temperatura do ar (air), sendo que ela foi transformada de Kelvin para Celsius. Os demais valores, 5 30 5 representam o menor valor, maior valor e o intervalo de variação, respectivamente.

## 48.8. Usando o script `clave.gs` para gerar climatologia

Este “.gs” pode ser encontrado no seguinte endereço <http://kodama.fubuki.info/wiki/wiki.cgi/GrADS/script?lang=en>.

### **Exemplo 1:** Climatologia para DEZ-JAN-FEV

```
'reinit'  
  
'set display color white'  
  
'c'  
  
'sdfopen air.nc'  
  
'set lat -60 20'  
  
'set lon -100 -20'  
  
'set grads off'  
  
'set gxout shaded'  
  
'set lev 1000'  
  
'clave air 01dec%y 28feb%ypp 1979 1986'  
  
'cbarn'
```

A climatologia da Temperatura do Ar foi feita para o nível de 1000 hPa apenas para o verão (DEZ-JAN-FEV). Observe que tem um "pp" após o "y" em "feb". Isto quer dizer que, quando se faz a climatologia onde se "quebra" o mês de ano para o outro ano, deve-se colocar o "pp" após o "y". Em outras palavras, aqui foi feito de 01dez/1979 até 28feb/1980, 01dez/1980 até 28feb/1981 ... 01dez/1986 até 28feb/1987 , veja que houve uma quebra no ano, por isso, usou-se o "pp".

## **Exemplo 2:** Climatologia para JUN-JUL-AGO

```
'reinit'
```

```
'set display color white'
```

```
'c'
```

```
'sdfopen air.nc'
```

```
'set lat -60 20'
```

```
'set lon -100 -20'
```

```
'set grads off'
```

```
'set gxout shaded'
```

```
'set lev 1000'
```

```
'clave air 01jun%y 31jul%y 1979 1986'
```

```
'cbarn'
```

Observe que não tem "pp" pois não houve quebra de ano. Observe que foi feito de 01jun/1979 até 31jul1979, 01jun/1980 até 31jul1980 ... 01jun/1986 até 31jul1986.

### **48.9. Usando o script drawbox.gs para desenhar uma caixa**

Uma forma mais simples de desenhar uma caixa sobre um mapa é usar o drawbox.gs. Este ".gs" pode ser encontrado no seguinte endereço [http://www.atmos.umd.edu/~bguan/grads/GrADS\\_Scripts.html](http://www.atmos.umd.edu/~bguan/grads/GrADS_Scripts.html).

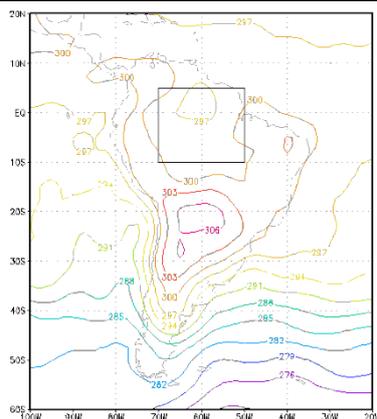


Figura 38 – Usando o drawbox.gs.

```
'reinit'

'set display color white'

'c'

'sdfopen air.nc'

'set lat -60 20'

'set lon -100 -20'

'set grads off'

'set gxout shaded'

'd air'

'drawbox -60 -50 -10 0'

'printim caixa.gif'
```

Desenha uma caixa, dado as coordenadas lon1 lon2 lat1 lat2. Onde: **lon1** representa a longitude mais oeste e **lat2** a latitude mais ao sul. Em outras

palavras, foi desenhada uma caixa que abrange as longitudes de -60 a -50 e latitudes de -10 a 0.

#### 48.10. Usando o script rmean.gs para calcular média móvel

Este “.gs” pode ser encontrado no seguinte endereço [http://www.atmos.umd.edu/~bguan/grads/GrADS\\_Scripts.html](http://www.atmos.umd.edu/~bguan/grads/GrADS_Scripts.html).

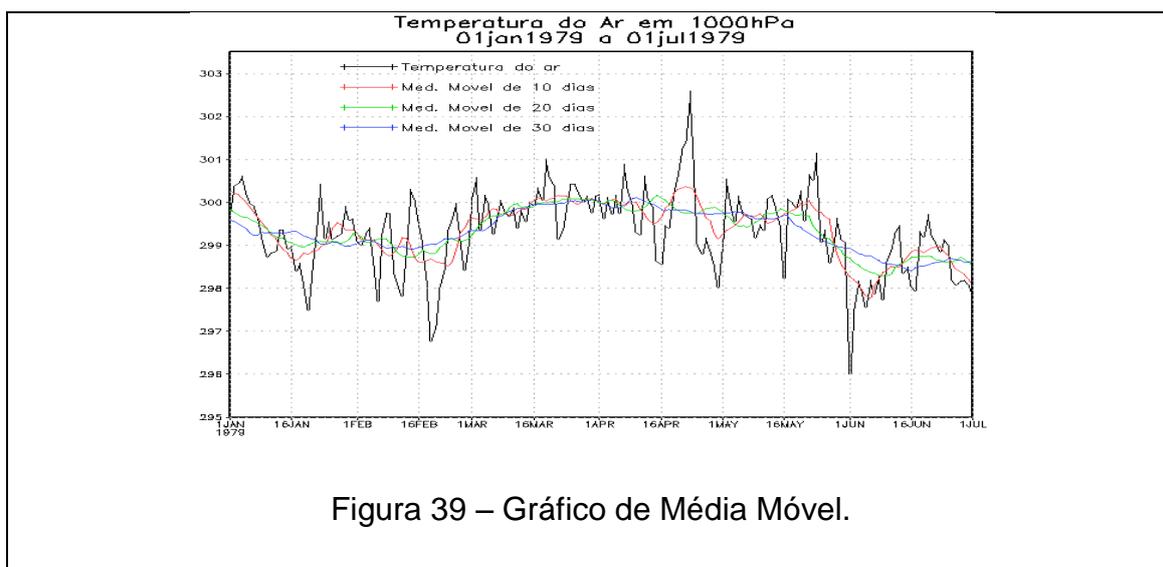


Figura 39 – Gráfico de Média Móvel.

```
'reinit'  
'sdfopen air.nc'  
'set t 1 182'  
'set lon -50'  
'set lat -10'  
'set cmark 0' *define o tipo de marcador (não desenha marcador, valor 0)  
'd air'  
'draw title Temperatura do Ar em 1000hPa \ 01jan1979 a 01jul1979'
```

```
'set ccolor 2' ;*define cor vermelha (2)

'set cmark 0'

'rmean air 10 display' ;*realiza média móvel de 10 dias

'set ccolor 3' ;*define cor verde (3)

'set cmark 0'

'rmean air 20 display' ;*realiza média móvel de 20 dias

'set ccolor 4' ;*define cor vermelha (4)

'set cmark 0'

'rmean air 30 display' ;*realiza média móvel de 30 dias

'cbar_line -x 3.31094 -y 7.46761 -c 1 2 3 4 -m 1 1 1 1 -l 1 1 1 1 -t "Temperatura
do ar" "Med. Movel de 10 dias" "Med. Movel de 20 dias" "Med. Movel de 30
dias"
```

O script `cbar_line.gs` cria uma legenda de acordo com as configurações dadas pelo usuário. Os valores `-x 3.31094` e `-y 7.46761` são as posições na página (estes valores foram obtidos usando o comando “q pos” visto no início da apostila) onde ficarão as legendas. O `-c` representa a cor atribuída a cada legenda. O `-m` é o tipo de marcador e `-t`, os textos que serão escritos na legenda.

#### 48.11. Convertendo a Temperatura de Kelvin para Celsius

```
'reinit'
```

```
'set display color white'
```

```
'c'
```

```
'sdfopen air.nc'
```

```
'set lon -100 -20'
```

```
'set lat -60 20'
```

```
'set grads off'
```

```
'set gxout shaded'
```

```
'd air-273.15' ;*converte de K para °C
```

#### 48.12. Configurando os vetores no gráfico de vento

##### Exemplo 1: Plotando somente os vetores

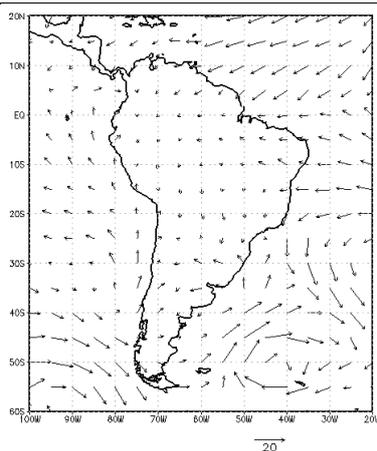


Figura 40 – Configurando os vetores do vento.

```
'reinit'  
  
'set display color white'  
  
'c'  
  
'sdfopen uwnd.1990.nc'  
  
'sdfopen vwnd.1990.nc'  
  
'set lon -100 -20'  
  
'set lat -60 20'  
  
'set grads off'  
  
'set map 1 1 6'  
  
'set gxout shaded'  
  
'd skip(uwnd,2,2);vwnd.2'  
  
'cbarn"printim skip1.gif'
```

O argumento (2,2) após uwnd aumenta ou diminui o espaçamento em x e y entre as setas. Modifique de acordo com as suas necessidades.

## Exemplo 2: Plotando os vetores e suas magnitudes

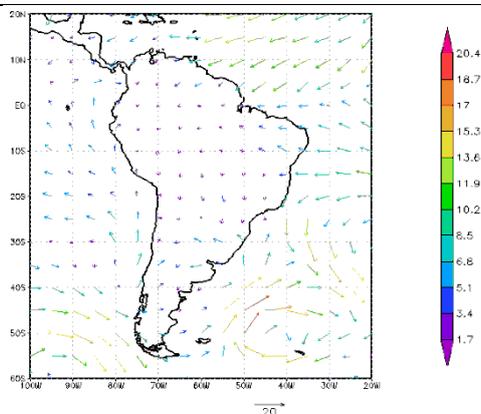


Figura 41 – Plotando a magnitude do vento.

```
'reinit'
```

```
'set display color white'
```

```
'c'
```

```
'sdfopen uwnd.1990.nc'
```

```
'sdfopen vwnd.1990.nc'
```

```
'set lon -100 -20'
```

```
'set lat -60 20'
```

```
'set grads off'
```

```
'set map 1 1 6'
```

```
'set gxout shaded'
```

```
'd skip(uwnd,2,2);vwnd.2;mag(uwnd,vwnd.2)'
```

```
'cbarn'
```

### 48.13. Calculando a vorticidade

Os cálculos da vorticidade e divergência são feitos usando diferenças finitas, por isso não existem valores nas bordas do domínio. Este cálculo está disponível em <http://www.iges.org/grads/gadoc/gradfunccdifff.html>.

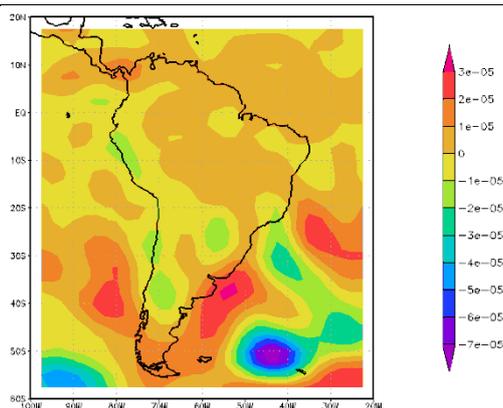


Figura 42 – Calculando a vorticidade.

```
'reinit'  
  
'set display color white'  
  
'c'  
  
'sdfopen uwnd.1990.nc'  
  
'sdfopen vwnd.1990.nc'  
  
'set lon -100 -20'  
  
'set lat -60 20'  
  
'set grads off'  
  
'set map 1 1 6'
```

```
'set gxout shaded'
```

```
'd hcurl(uwnd,vwnd.2)'
```

```
'cbarn'
```

```
'printim hcurl.gif'
```

#### 48.14. Calculando a divergência

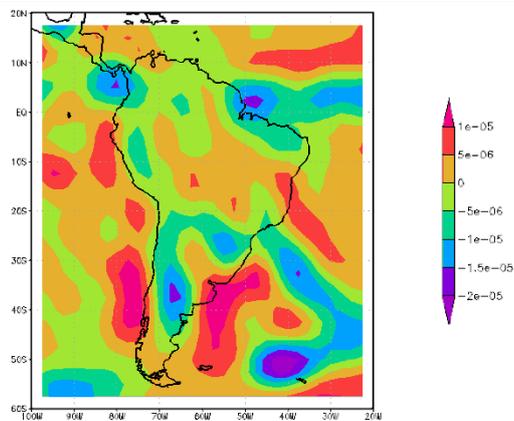


Figura 43 – Calculando a divergência.

```
'reinit'
```

```
'set display color white'
```

```
'c'
```

```
'sdfopen uwnd.1990.nc'
```

```
'sdfopen vwnd.1990.nc'
```

```
'set lon -100 -20'
```

```
'set lat -60 20'
```

```
'set grads off'  
  
'set map 1 1 6'  
  
'set gxout shaded'  
  
*valores positivos = divergência  
  
*valores negativos = convergência  
  
'd hdivg(uwnd,vwnd.2)'  
  
'cbarn'  
  
'printim hdivg.gif'
```

#### 48.15. Criando template com dados Netcdf

A criação do template para arquivos Netcdf é diferente quando comparado com os formatos conhecidos (.bin, .grb etc). **Esse procedimento só funciona no GrADS 1.9.**

Supondo que você tenha dados diários de vento a partir de 1980 até 1982 (em arquivos separados, ou seja, 3 arquivos) e com isso você deseja varrer todos os tempos para cada ano de uma única vez. Para realizar esta tarefa, basta digitar a linha abaixo que deverá ser feita com o GrADS aberto.

```
ga->sdfopen uwnd.1980.nc uwnd.%y4.nc 1095
```

#### **Explicação:**

**uwnd.1980.nc** é o primeiro arquivo aberto. Será o arquivo referencial.

**uwnd.%y4.nc** é o formato para criação do template. Neste caso estamos fazendo para ano com quatro dígitos (%y4) porque nossos dados estão

separados por ano. O nome uwnd poderia ter qualquer nome, isso fica a critério do usuário. Para obter mais informações sobre a criação de templates, visite a página Oficial do Grads (<http://grads.iges.org/grads/gadoc/templates.html>).

O valor **1095** representa o número de tempos que será varrido, no total serão três anos (365 [dias] x 3 [anos] = 1095 tempos ou dias, neste caso). Em outras palavras, a variável uwnd será varrida desde 1980 até 1982.

#### 48.16. Preenchendo regiões específicas do gráfico

O script hatch.gs (disponível em <http://kodama.fubuki.info/wiki/wiki.cgi/GrADS/script?lang=en>) preenche regiões específicas do seu mapa lat/lon. Para usá-lo basta seguir o exemplo abaixo.

##### Sintaxe:

1. hatch.gs <var> <min> <max> -angle 45 -density 0.01 -int 0.1 ou
2. hatch.gs <var> -min <min> -angle 45 -density 0.01 -int 0.1 ou
3. hatch.gs <var> -max <max> -angle 45 -density 0.01 -int 0.1

Onde:

<var> é o nome da variável do seu arquivo

<min> <max> preenche apenas os valores entre min e max

-min <min> preenche os valores acima de min

-max <max> preenche os valores abaixo de max

-angle o valor padrão é 45 e representa a inclinação do preenchimento. Os ângulos variam no sentido anti-horário

-density o valor padrão é de 0.01. Representa a densidade, ou seja, mais ou menos preenchimento

**-int** o valor padrão é 0.1 e representa o intervalo de preenchimento

**1. Exemplo:** Plotando valores entre **<min>** e **<max>**

```
'reinit'
```

```
'sdfopen uwnd.nc'
```

```
'set lat -30 10'
```

```
'set lon 280 330'
```

```
'set map 1 1 6'
```

```
'set gxout shaded'
```

```
'd msp'
```

```
'cbarn'
```

```
'hatch msp 10 15 -angle 45 -density 0.01 -int 0.1'
```

**2. Exemplo:** Preenchendo valores acima de **<min>**

```
'reinit'
```

```
'sdfopen uwnd.nc'
```

```
'set lat -30 10'
```

```
'set lon 280 330'
```

```
'set map 1 1 6'
```

```
'set gxout shaded'
```

```
'd msp'
```

```
'cbarn'
```

```
'hatch msp -min 15 -angle 45 -density 0.01 -int 0.1'
```

### **3. Exemplo:** Preenchendo valores abaixo de **<max>**

```
'reinit'
```

```
'sdfopen gui.nc'
```

```
'set lat -30 10'
```

```
'set lon 280 330'
```

```
'set map 1 1 6'
```

```
'set gxout shaded'
```

```
'd msp'
```

```
'cbarn'
```

```
'hatch msp -max 15 -angle 45 -density 0.01 -int 0.1'
```

#### **48.17. Modificando as linhas horizontais e verticais**

Para modificar as linhas horizontais e verticais do mapa, usamos o comando abaixo:

Sintaxe:

**set grid <on> ou <off> <estilo\_linha> <cor> ou <horizontal> ou <vertical>**

Onde:

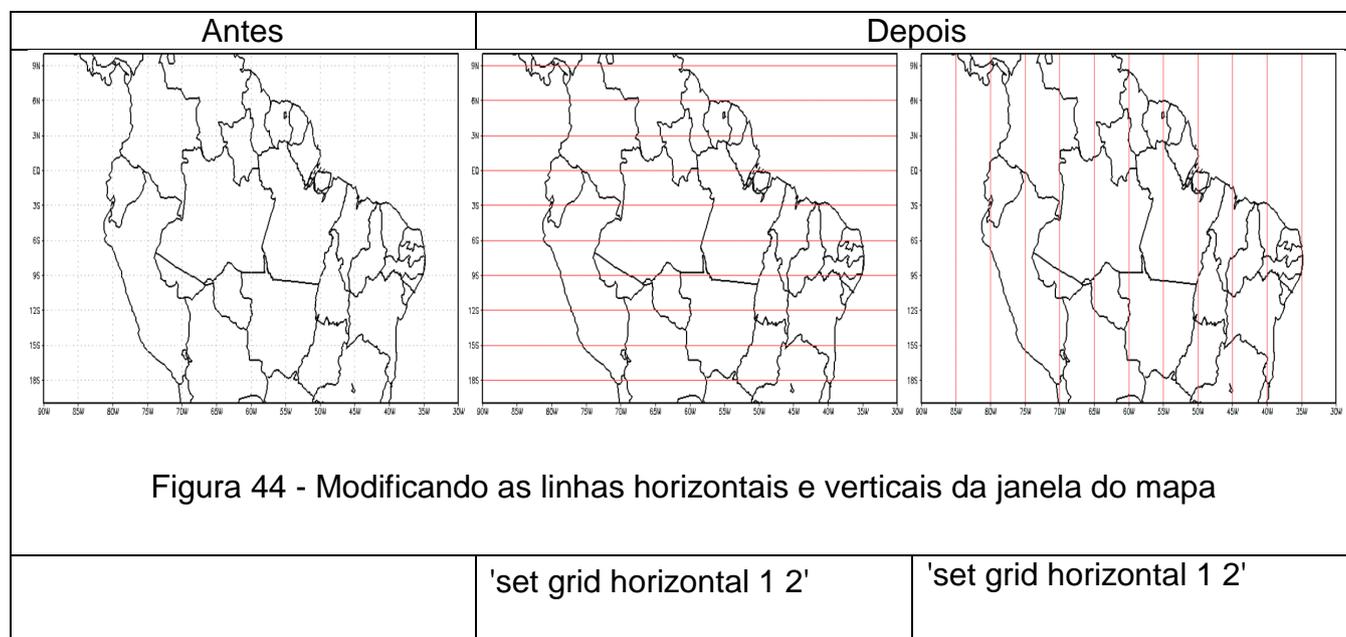
<on> habilita ou desabilita <off> as linhas

<estilo\_linha> ver estilos de linhas (Figura 4)

<cor> ver figura de cores (Figura 3)

<horizontal> fixa apenas as linha horizontais

<vertical> fixa apenas linhas horizontais



#### 48.18. Modificando a cor e espessura da borda da janela do mapa

Sintaxe: `set annot <cor> <espessura_linha>`

Onde:

<cor> ver tabela de cores (Figura 3)

<espessura\_linha> varia de 1 a 6

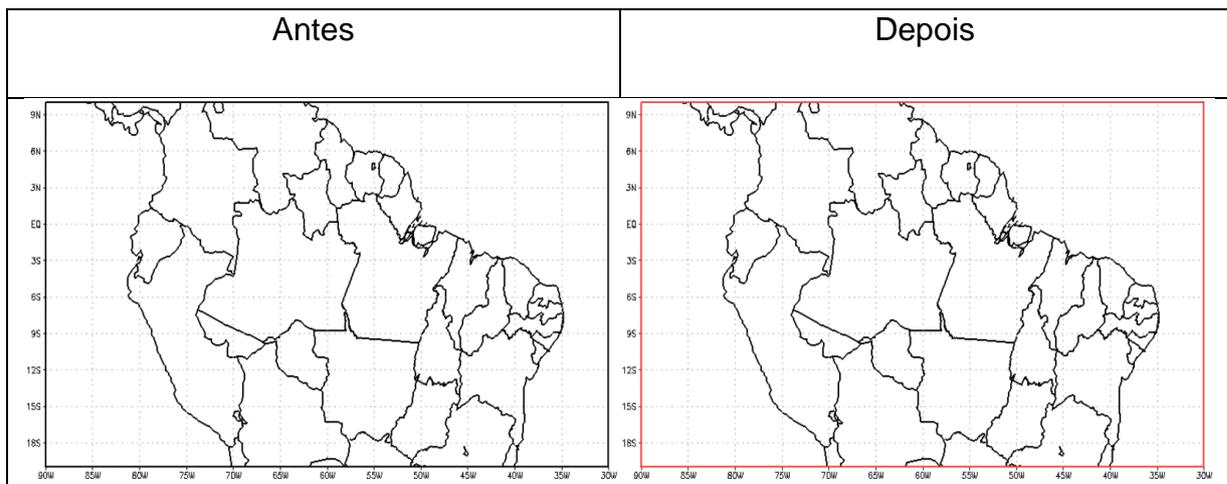


Figura 45 - Modificando a cor e espessura da borda da janela do mapa

```
'reinit'
'sdfopen air.nc'
'set annot 2 6'
'd air'
```

Neste exemplo, fixamos a cor vermelha (2) e espessura 6 para a borda da janela.

## 49 FUNÇÕES MATEMÁTICAS

### 49.1. Funções trigonométricas

**Sintaxe:** `rc = math_funcao(angulo)`

Onde:

funcao pode ser sin, cos, tan, asin, acos, atan, atan2, sinh, cosh, tanh, asinh, acosh ou atanh

**Exemplo:**

```
pi = 3.1415926 ;* valor de PI
```

```
d2r = pi/180
```

```
angd = 45 ;* ângulo a ser convertido
```

```
ang = angd * d2r ;* conversão de graus para radianos
```

```
cos = math_cos(ang) ;* realiza o cálculo
```

```
say ' COSSENO DE ' angd ' = ' cos *; mostra na tela o resultado
```

A resposta para este exemplo (cosseno de 45) será 0.707106790659975.

### 49.2. Formatando números

**Sintaxe:** `rc = math_format(fmt,numero)`

Onde:

fmt assume o seguinte formato, por exemplo %5.2f

numero é o número a ser formatado

**Exemplo:**

```
numero = 3.1456
```

```
fmt = '%6.1f'
```

```
rc = math_format(f,v)
```

```
say ' FORMATANDO ' v ' - NOVO FORMATO USADO ' fmt ' = ' rc
```

A resposta para este exemplo será 3.1.

### 49.3. Número real no formato decimal

Arredonda o número decimal para o inteiro mais próximo.

**Sintaxe: math\_nint(numero)**

**Exemplo:**

```
numero = 3.1456
```

```
rc = math_nint(numero)
```

```
say ' INTEIRO MAIS PROXIMO DE ' numero ' = ' rc
```

A resposta para este exemplo será 3 pois se for feito arredondamento o resultado será esse. Se **numero** fosse 3.65 o resultado seria 4.

### 49.4. Obtendo a parte inteira de um número decimal

Retorna apenas a parte inteira de um número decimal.

**Sintaxe: rc = math\_int(numero)**

**Exemplo:**

```
numero = 4.7456
```

```
rc = math_int(v)
```

```
say ' PARTE INTEIRA DE ' numero ' = ' rc
```

A resposta para este exemplo será 4.

#### 49.5. Calculando o logaritmo natural

Calcula o logaritmo natural sendo que o número calculado deve ser maior que zero.

**Sintaxe:** `math_log(numero)`

**Exemplo:**

```
numero = 3.1456
```

```
rc = math_log(numero)
```

```
say ' LOG DE ' num ' = ' rc
```

A resposta para este exemplo será 0.497703496152042

#### 49.6. Calculando potência

**Sintaxe:** `rc = math_pow(numero,exp)`

**Exemplo:**

```
numero = 10
```

```
exp = 4
```

```
rc = math_pow(numero,exp)
```

```
say numero ' ELEVADO A ' exp ' = ' rc
```

A resposta para este exemplo será  $10000 = 10^4$ .

#### 49.7. Calculando raiz quadrada

**Sintaxe:** `math_sqrt(numero)`

**Exemplo:**

```
numero = 4
```

```
rc = math_sqrt(numero)
```

```
say 'RAIZ QUADRADA DE ' numero ' = ' rc
```

A resposta para este exemplo será  $\sqrt{4} = 2$ .

#### 49.8. Obtendo o valor absoluto

**Sintaxe:** `rc = math_abs(numero)`

**Exemplo:**

```
num = -3
```

```
rc = math_abs(numero)
```

```
say ' VALOR ABSOLUTO DE ' numero ' = ' rc
```

#### 49.9. Calculando exponencial

**Sintaxe:** `rc = math_exp(numero)`

**Exemplo:**

```
numero = 1
```

```
rc = math_exp(numero)
```

```
say ' VALOR DA EXPONENCIAL DE ' numero ' = ' rc
```

A resposta para este exemplo será  $e^1 = 2.71828182845905$ .

#### 49.10. Resto da divisão

Retorna o resto da divisão do numero1 pelo numero2.

**Sintaxe:** `rc = math_fmod(numero1,numero2)`

**Exemplo:**

numero1 = 6

numero2 = 2.25

`rc = math_fmod(numero1,numero2)`

`say 'RESTO DA DIVISAO DE ' numero1 ' POR ' numero2 ' = ' rc`

A resposta para este exemplo será 1.5.

**Ilustração:**

$$\begin{array}{r|l} 6 & 2.25 \\ 1.5 & \underline{2} \end{array}$$

O resto dessa divisão é 1.5, que é o resultado do exemplo acima.

#### 49.11. Parte inteira do resto da divisão

Retorna parte inteira do resto da divisão de numero1 por numero2.

**Sintaxe:** `rc = math_mod(num1,num2)`

**Exemplo:**

num1 = 6

num2 = 2.25

```
rc = math_mod(num1,num2)
```

```
say 'PARTE INTEIRA DA DIVISAO DE ' num1 ' POR ' num2 ' = ' rc
```

A resposta para este exemplo será 1.

**Ilustração:**

$$\begin{array}{r|l} 6 & 2.25 \\ 1.5 & 2 \end{array}$$

O resto dessa divisão é 1.5, porém esta função retorna apenas a parte inteira do resto, ou seja, o valor 1

**49.12. Obtendo o tamanho de uma string**

Retorna o tamanho de uma string.

**Sintaxe:** `rc = math_strlen(string)`

**Exemplo:**

```
string = grads
```

```
rc = math_strlen(string)
```

```
say 'TAMANHO DA STRING ' string ' = ' rc
```

A resposta será que “grads” tem 5 caracteres.

**49.13. Verificando o valor de uma string**

Ao fornecer uma informação seja ela uma string ou um número, se o valor retornado informar:

0 → não é um número

1 → é um inteiro

2 → não é um inteiro

**Sintaxe: rc = valnum(string)**

**Exemplo 1:**

```
string = 'grads'
```

```
rc = valnum(string)
```

```
say 'E UMA STRING --> ' string ' = ' rc
```

Neste caso, o valor retornado será 0 (zero) pois “grads” é uma string e não um número

**Exemplo 2:**

```
string = '2'
```

```
rc = valnum(string)
```

```
say 'E UM INTEIRO --> ' string ' = ' rc
```

A string vale 2 por isso será retornado o valor 1, ou seja, é um inteiro

**Exemplo 3:**

```
string = '-5.8'
```

```
rc = valnum(string)
```

```
say 'E NAO INTEIRO --> ' string ' = ' rc
```

O valor da string é “-5.8” com isso temos como resultado o valor 2, ou seja, é não inteiro.

#### 49.14. Posição de uma string

Dada uma string (uma frase neste caso) que contém mais de uma palavra, ao selecionar a posição da string que você deseja, será retornada a sua posição.

**Sintaxe: rc = wrdpos(string,posicao)**

Onde:

**string** é a string que contém mais de uma palavra

**posição** é um número inteiro e diz qual a string você deseja obter a posição

**Importante: Espaços em branco contam como caracter.**

**Exemplo:**

**Posições**                    **1**    **2**        **3**  
string            =    'UMA FRASE QUALQUER'

posicao            = 2 ;\* será capturada a string "FRASE"

rc = wrdpos(string,posicao)

say ' STRING USADA ' string '. POSICAO DE INTERESSE ' posicao '. A  
STRING COMECA EM ' rc

Veja que "UMA FRASE QUALQUER" tem 18 caracteres, conte os espaços em branco. A resposta para este exemplo será 5, pois é onde começa a string selecionada na posição 2 (FRASE).

#### 49.15. Gráfico de linha preenchido

Antes	Depois
-------	--------

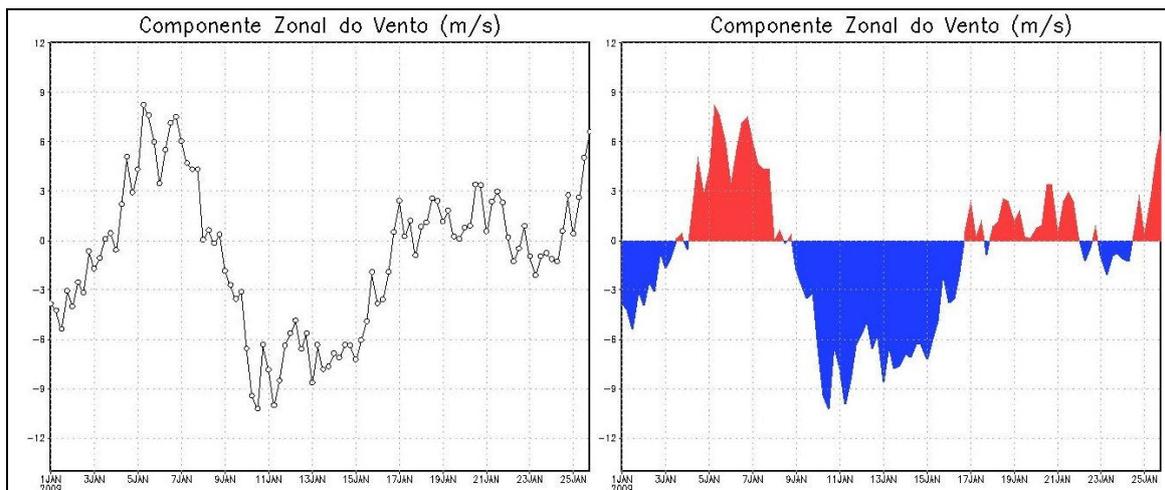


Figura 46 – Gráfico de linha preenchido.

```
'reinit'
```

```
'set display color white'
```

```
'c'
```

```
'sdfopen uwnd.2009.nc'
```

```
'set gxout linefill' ;*gráfico de linha preenchido
```

```
'set t 1 100' ;*fixa um intervalo de tempo
```

```
'set lat -10' ;*fixa uma latitude
```

```
'set lon 300' ;*fixa uma longitude
```

```
'set lfcols 2 4' ;* onde u>0 preenche com cor vermelho(2)
```

```
;* onde u<0 preenche com cor azul (4)
```

```
'd u;u*0' ;* u é a componente zonal e u*0 tem que ser uma constante
```

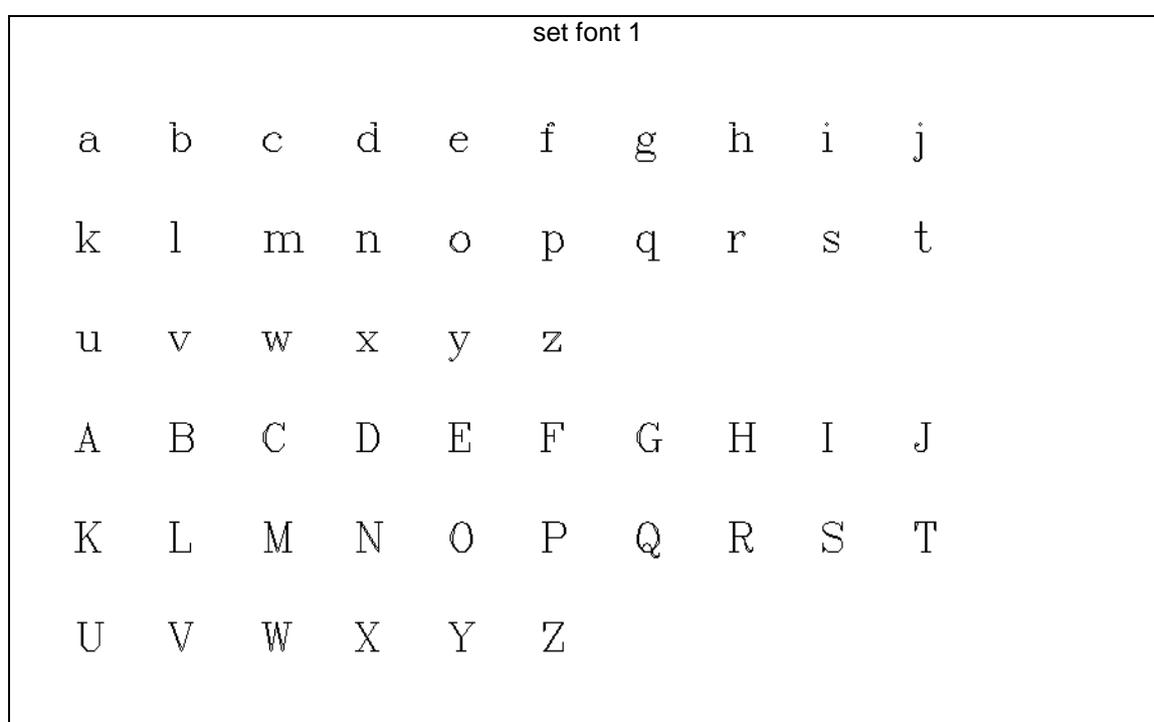
```
'draw title Componente Zonal do Vento (m/s)'
```

## 50 TIPOS DE FONTES

No GrADS são disponibilizados 5 tipos de fontes. Estas fontes correspondem a letras em itálico, negrito e símbolos matemáticos.

**Sintaxe:** set font <valor>

Onde: <valor> varia de 1 a 5. Por padrão o GrADS sempre usa a opção 1. Cabe ao usuário modificar para as demais opções disponíveis.



set font 2

*a b c d e f g h i j*  
*k l m n o p q r s t*  
*u v w x y z*  
*A B C D E F G H I J*  
*K L M N O P Q R S T*  
*U V W X Y Z*

set font 3

$\alpha \beta \gamma \delta \epsilon \zeta \eta \vartheta \iota \xi$   
 $\kappa \lambda \mu \nu \omicron \pi \epsilon \rho \sigma \tau$   
 $\upsilon \varphi \omega \phi \psi \theta$   
 $\Lambda \therefore \Gamma \Delta \Xi \heartsuit \heartsuit \diamond \clubsuit \spadesuit$   
 $\sqsupset \blacksquare \triangle \diamond \Theta \Pi \star \star \Sigma \odot$   
 $\Upsilon \blacktriangleright \Phi \chi \Psi \Omega$

set font 4

a	b	c	d	e	f	g	h	i	j
k	l	m	n	o	p	q	r	s	t
u	v	w	x	y	z				
A	B	C	D	E	F	G	H	I	J
K	L	M	N	O	P	Q	R	S	T
U	V	W	X	Y	Z				

set font 5

a	b	c	d	e	f	g	h	i	j
k	l	m	n	o	p	q	r	s	t
u	v	w	x	y	z				
A	B	C	D	E	F	G	H	I	J
K	L	M	N	O	P	Q	R	S	T
U	V	W	X	Y	Z				

## 51 LINKS INTERESSANTES SOBRE O GRADS

Não deixe de acessar a página oficial do GrADS. Estão disponíveis tutoriais, exemplos, scripts dentre outras ferramentas.

1. Endereço que contém esta apostila: <https://sites.google.com/site/jgmsantos>
2. Página oficial do GrADS: <http://www.grads.iges.org/grads>
3. [http://www.atmos.umd.edu/~bguan/grads/GrADS\\_Scripts.html](http://www.atmos.umd.edu/~bguan/grads/GrADS_Scripts.html)
4. <http://ccr.aos.wisc.edu/model/processing/>
5. <http://www.cptec.inpe.br/ManualGrADS/library.html>
6. [http://www.ifm-geomar.de/index.php?id=grads\\_tools](http://www.ifm-geomar.de/index.php?id=grads_tools)
7. <http://wind.geophys.tohoku.ac.jp/~kodamail/wiki/wiki.cgi/GrADS/script?lang=en>
8. <http://www.iges.org/grads/gadoc/gadocindex.html>
9. <http://ccr.aos.wisc.edu/model/processing/#grads>

## REFERÊNCIAS BIBLIOGRÁFICAS

NASA Advanced Information Systems Research Program. **GrADS reference card version 1.7**. Disponível em:  
<[http://grads.iges.org/grads/gadoc/reference\\_card.pdf](http://grads.iges.org/grads/gadoc/reference_card.pdf)> Acesso em: 10 jul. 2009.

NASA Advanced Information Systems Research Program. **GrADS scripting language reference card version 1.7**. Disponível em:  
<[http://grads.iges.org/grads/gadoc/reference\\_card\\_scl.pdf](http://grads.iges.org/grads/gadoc/reference_card_scl.pdf)> Acesso em: 10 jul. 2009.

GRID ANALYSIS AND DISPLAY SYSTEM (GRADS). **GrADS Documentation Index**. Disponível em: <<http://grads.iges.org/grads/gadoc/gadocindex.html>>  
Acesso em: 10 jul. 2009.

## **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 programa 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 as sequências de instruções ou códigos, expressos em uma linguagem de programação compilada ou interpretada, a ser executada por um computador para alcançar um determinado objetivo. São aceitos tanto programas fonte quanto executáveis.

### **Pré-publicações (PRE)**

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