



Ministério da
**Ciência, Tecnologia
e Inovação**



sid.inpe.br/mtc-m21b/2014/04.14.18.23-PUD

INTRODUÇÃO AO NCAR COMMAND LANGUAGE (NCL)

José Guilherme Martins dos Santos

URL do documento original:

<<http://urlib.net/8JMKD3MGP5W34M/3G5LGP5>>

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^a 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^a 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/2014/04.14.18.23-PUD

INTRODUÇÃO AO NCAR COMMAND LANGUAGE (NCL)

José Guilherme Martins dos Santos

URL do documento original:

<<http://urlib.net/8JMKD3MGP5W34M/3G5LGP5>>

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

	<u>Pág.</u>
1 INTRODUÇÃO	1
2 INSTALAÇÃO DO NCL	3
3 VISÃO GERAL DO NCL	7
3.1 Formatos disponíveis de saída	7
3.2 Tipos de dados	7
3.3 Expressões numéricas	7
3.4 Variáveis	8
3.5 Loops	8
3.6 Condicionais	9
3.7 Dimensão e indexação	9
3.8 Redução de dimensão	10
3.9 Arquivos suportados	10
3.10 Imprimir o conteúdo de um arquivo	11
3.11 Importando dados short ou byte e convertendo para float	12
3.12 Convertendo formatos para netCDF	12
4 COMO OBTER AJUDA	13
5 ACENTUAÇÃO EM NCL	15
6 SUPERSCRIPTS E SUBSCRIPTS	17
7 CORES, FONTES, ESTILOS DE LINHA, PREENCHIMENTO, MARCADORES E PROJEÇÕES	19
8 GRÁFICOS	21
8.1 Gráfico de linha	22
8.2 Gráfico de barra	29
8.3 Gráfico espacial	32
9 CONSIDERAÇÕES FINAIS	39
10 LINKS IMPORTANTES	41

REFERÊNCIAS BIBLIOGRÁFICAS 43

1 INTRODUÇÃO

Dentre as ferramentas de trabalho para visualizar e manipular dados meteorológicos, o *NCAR Command Language* ou simplesmente NCL tem se mostrado uma ferramenta altamente poderosa com suas diversas funções, scripts e uma infinidade de opções disponíveis no site do [NCL](#) para manipular e visualizar dados dos mais variados tipos. E o melhor de tudo, o *software* é grátis. Além disso, a excelente lista de discussão onde os usuários estão sempre prontos a ajudar quem precisa é fantástica. Claro que existem outras ferramentas além do NCL, mas ao meu ver é sem dúvida uma das melhores que já tive acesso e portanto, aconselho quem estiver interessado em usá-lo que vá em frente que a satisfação é garantida.

Esse tutorial serve como apoio e soma-se aos existentes na internet como o manual produzido pelo Mateus Teixeira disponível no link abaixo.

http://www.ncl.ucar.edu/Document/Manuals/ncl_ptBR.pdf.

2 INSTALAÇÃO DO NCL

O NCL usado neste tutorial foi instalado no Ubuntu 32 bits, para demais distribuições Linux verificar possíveis dependências. Faça uma atualização do Linux antes de proceder com o restante dos passos.

Para realizar o download do NCL clique no link <http://www.earthsystemgrid.org/home.htm>. Em seguida, clique em **Login** que está no canto superior direito da página. Ao acessar a página com o seu usuário e senha após criar seu usuário caso você não possua, clique em **NCL: NCAR Command Language** que está logo abaixo. Com isso, serão mostradas várias versões do NCL. Para essa instalação será utilizada a versão **NCL Version 6.1.2**. Utilize a versão pré-compilada, pois dessa forma, não será necessária a instalação de pacotes adicionais.

Selecione a versão **NCL Version 6.1.2 precompiled binaries, OPeNDAP-enabled** que é suficiente para processar dados e gerar gráficos. Logo em seguida, clique em **Download Files**. Aceite o termo de uso e depois clique em **Download individual files**. Estarão disponíveis várias versões do NCL.6.1.2. E qual devo usar no meu Linux? Digite no seu terminal **gcc -v** para saber qual versão do gcc está instalada no seu Linux. Lembre-se de antes realizar uma atualização do seu sistema Linux para evitar conflitos de bibliotecas. No meu caso, a versão do gcc instalada é a 4.6.3. Sendo assim, utilizarei a versão **ncl_ncarg-6.1.2.Linux_Debian6.0_i686_gcc445.tar.gz** que é a mais próxima da versão do gcc instalada no meu computador. Só um detalhe, o **i686** quer dizer sistema de 32 bits que é o sistema utilizado nesse tutorial.

Após realizar o download do arquivo acima, crie uma pasta chamada **ncarg** em **/usr/local**. Use o **sudo mkdir ncarg** pois precisará de permissão de root. Em seguida, descompacte o arquivo fazendo:

```
tar -zxvf ncl_ncarg-6.1.2.Linux_Debian6.0_i686_gcc445.tar.gz
```

Serão criados os diretórios: **bin**, **lib** e **include**

Copie esses diretórios para **/usr/local/ncarg**. Como é um diretório fora do seu **home**, será necessário copiar com permissão de root, então proceda da seguinte forma:

```
sudo cp -r bin lib include /usr/local/ncarg/
```

Clique em <http://www.ncl.ucar.edu/Document/Graphics/hluresfile> e salve-o no seu **home**. Perceba que não tem extensão alguma nesse arquivo porque ele é um arquivo de configuração do NCL que pode alterar o tamanho da tela, tipo de fonte usada, cor e etc. Fica a critério do usuário explorar suas características. Clique no link abaixo para ler sobre a descrição desse arquivo.

<http://www.ncl.ucar.edu/Document/Graphics/hlures.shtml>.

Com o arquivo **hluresfile** no seu **home**, proceda da seguinte forma:

```
mv hluresfile .hluresfile
```

Dessa forma, ele passa a ser um arquivo oculto do sistema. Para ver se ele está oculto, digite no seu **home** o comando **ls -a**, com isso você verá algo como **.hluresfile**, tem um ponto na frente do arquivo.

O próximo passo é editar o arquivo **.bashrc** que também é um arquivo oculto do sistema porque tem um ponto na frente do arquivo, para visualizar esse arquivo digite **ls -a** no seu **home** e **adicione as linhas abaixo em vermelho caso não tenha a variável PATH no seu .bashrc**:

```
export NCARG_ROOT=/usr/local/ncarg
```

Os caminhos na variável PATH é tudo numa só linha.

```
PATH=/usr/bin:/bin:/usr/local/bin:$PATH:/usr/bin:/usr/bin/X11:  
/usr/local/bin:/lib:/usr/lib:./:/usr/sbin:/usr/local/bin:/usr/local/bin:  
$NCARG_ROOT/bin
```

```
export PATH=$NCARG_ROOT/bin:$PATH
```

```
export LD_LIBRARY_PATH=/usr/local/lib:/usr/lib:/lib
```

Não esqueça de atualizar o seu **.bashrc** digitando no seu **HOME**:

```
source .bashrc
```

Agora, deve-se digitar **ncl** no terminal do Linux para saber se o programa foi instalado corretamente, isso pode ser feito em qualquer lugar. Caso apareça o erro abaixo:

```
ncl: error while loading shared libraries: libssl.so.0.9.8: cannot open sha-
```

red object file: No such file or directory

Digite no seu terminal: **sudo apt-get install libssl0.9.8**

Ao instalar essa biblioteca, digite novamente ncl e caso apareça o erro abaixo:

bash: /usr/local/ncarg/bin/ng4ex: /bin/csh: interpretador inválido: Arquivo ou diretório não encontrado

Digite no seu terminal Linux: **sudo apt-get install csh**

Os passo acima resolvem os erros.

Vamos testar se o NCL foi instalado corretamente. Digite no terminal **ncl** e se aparecer as linhas abaixo, o ncl foi instalado corretamente.

**Copyright (C) 1995-2013 - All Rights Reserved University Corporation
for Atmospheric Research NCAR Command Language Version 6.1.2
The use of this software is governed by a License Agreement. See
<http://www.ncl.ucar.edu/> for more details.**

Para encerrar a sessão digite **quit** ou **exit**.

Outra forma de testar se o ncl foi instalado corretamente é **digitar diretamente no terminal do Linux** e não com o ncl aberto a linha abaixo:

ng4ex gsun01n

Tem que aparecer uma sequência de imagens, isso é um script teste para saber se o NCL foi instalado corretamente. Apenas clique sobre as imagens que elas vão passando. Se isso acontecer, o ncl está pronto para gerar figuras.

3 VISÃO GERAL DO NCL

É fortemente aconselhável que o usuário tenha sempre em mãos o tutorial em português criado pelo Mateus Teixeira (http://www.ncl.ucar.edu/Document/Manuals/ncl_ptBR.pdf) e acesso a internet para tirar dúvidas no site oficial do NCL (<http://www.ncl.ucar.edu>).

A ideia desse tópico é dar uma visão geral sobre o NCL. Por isso, dúvidas mais específicas podem ser obtidas no site <http://www.ncl.ucar.edu/>.

O NCL é uma linguagem de programação interpretada com a função de acessar, analisar e visualizar dados ambientais. Uma excelente dica é acessar o site abaixo para ser ter uma ideia do volume de informações que o NCL disponibiliza.

http://www.ncl.ucar.edu/Document/Manuals/Ref_Manual

3.1 Formatos disponíveis de saída

Há seis tipos de saídas disponíveis em NCL: **ncgm** (NCAR computer graphics meta-file), **ps** (postscript), **eps** (encapsulated postscript, contains a bounding box), **epsi** (encapsulated postscript with a bitmap preview), **pdf**, **X11 window** e **png**.

Exemplo:

```
wks = gsn_open_wks("pdf","nome_figura")
```

```
wks_2 = gsn_open_wks("ps","nome_figura")
```

3.2 Tipos de dados

Numéricos: double (64 bits), float (32 bits), long (32 ou 64 bits), integer (32 bits), short (16 bits), byte (8 bits), números complexos não são suportados.

Não-numéricos: string, character, graphic, file, logical, list.

3.3 Expressões numéricas

Operadores algébricos:

+	soma (também pode ser usado para concatenação)
-	subtração
*	multiplicação
^	exponenciação
%	módulo
#	multiplicação de matrizes
>,<	maior que, menor que

Operadores lógicos:

.lt.	menor que
.le.	menor ou igual a
.gt.	maior ou igual a
.ne.	diferente de
.eq.	igual a
.and.	e
.or.	ou
.xor.	ou exclusivo
.not.	não

3.4 Variáveis

O nome das variáveis no NCL deve começar com caracter alfanumérico e pode conter mistura de números e letras. O símbolo *underscore* "_" é permitido na criação de variáveis.

3.5 Loops

Há dois tipos de loops em NCL: o **do while** e o **do**.

do n = início,fim,incremento_opcional

[sentença(s)]

end do ; *observe que há espaço*

do while (expressão_escalar_lógica)

[sentença(s)]

end do

3.6 Condicionais

Como em outras linguagens de programação usa-se em NCL duas estruturas de **if**. São elas: **if-then** e **if-then-else**.

```
if (expressão_escalar_lógica) then
```

```
    [sentença(s)]
```

```
end if
```

```
if (expressão_escalar_lógica) then
```

```
    [sentença(s)]
```

```
else
```

```
    [sentença(s)]
```

```
end if
```

3.7 Dimensão e indexação

Há duas formas de indexar arranjos em NCL. A forma **padrão** e a **coordenada**. O índice inicia em 0 e terminam em N-1. As dimensões mais a direita variam mais rapidamente, enquanto que, as dimensões mais a esquerda variam mais lentamente. Os arranjos possuem a seguinte forma:

início:fim:incremento

Exemplo1: Supondo que a variável temperature (T) seja tridimensional, isto é, tempo, latitude e longitude. A forma padrão de selecionar determinada dimensão é feita da seguinte forma:

```
T                ; seleciona toda a matriz de dados  
T(0,:::5)        ; primeiro tempo (0), todas as latitudes (:), a cada 5 graus de longitude (::5)  
T(0,::-1,4:40)   ; primeiro tempo (0), inverte a latitude (::-1), seleciona os índices 4 a 40 de longitude (4:40)  
T(:,1,45,10:20)  ; seleciona os dois primeiros tempos, latitude fixa no índice 45, e longitude variando de 10-20
```

Exemplo2: Outra forma de selecionar o dado é por meio da indexação por coordenadas. Nesse caso, deve-se usar o símbolo {...}.

```
X = T(:, -20:20, 90:290:2)
```

Nesse caso, foram selecionados todos os tempos (:), latitude entre 20°S e 20°N e longitude de 90 a 290 variando de 2 em 2 graus.

3.8 Redução de dimensão

Ao fixar a dimensão de uma variável ocorre sua redução. Supondo que T seja do tipo T(nt,nz,ny,nx):

T1 = T(5,;,12,:) ; apenas nz e nx variam enquanto nt (5) e ny (12) estão fixos.

T2 = T(;;,;,0) ; apenas nt, nz e ny variam enquanto nx está fixo.

3.9 Arquivos suportados

O NCL é capaz de ler e manipular dados nos seguintes formatos: netCDF3 e 4, HDF4, HDF4-EOS, HDF5, HDF5-EOS, GRIB-1, GRIB-2, shapefile, *CCM History Tape* e ascii. Informações adicionais sobre esses formatos podem ser obtidas no link abaixo.

<http://www.ncl.ucar.edu/Document/Language/supported.shtml>

A abertura de um arquivo é feita usando a função **addfile**. Veja o exemplo abaixo.

$$\mathbf{x} = \mathbf{addfile}(\text{“nome_arquivo.ext”,status})$$

Onde:

x é um nome dado pelo usuário para identificar o arquivo aberto. Pode ser informado o caminho completo ou o relativo,

nome_arquivo.ext é o nome do arquivo com sua extensão (.nc, .cdf, .hdf, .hdfeos, .grb ou .grib1 e .ccm). A extensão pode ser omitida que o NCL é capaz de identificar o arquivo a ser aberto e

status que pode ser **“r”** apenas para leitura de arquivos (há suporte para todos os tipos de arquivos), **“c”** criação de arquivos (somente netCDF e HDF4) e **“w”** para leitura e edição de arquivos (somente netCDF e HDF4).

Exemplo:

$$\mathbf{a} = \mathbf{addfile}(\text{“temperatura.nc”,“r”})$$

A letra **a** receberá o conteúdo do arquivo aberto **temperatura.nc** que será lido apenas como leitura (**“r”**).

E supondo que a variável desse arquivo tenha o nome de **temp**, para manipular essa variável usa-se o símbolo `->`.

Exemplo:

```
t=a->temp
```

A partir de agora, qualquer cálculo deve ser feito com a variável **t** que contém todas as informações de **temp**. Essa será a nova variável de referência.

3.10 Imprimir o conteúdo de um arquivo

Para visualizar o conteúdo de um arquivo usa-se o comando **print**. Esse comando é bem similar ao **ncdump -h**. Dessa forma, o usuário é capaz de saber o nome da variável do arquivo, dimensões (tempo, nível vertical, latitude e longitude). Há também informações sobre o tipo de dado da variável, isto é, *float*, *interger*, *character* ou *double*. É possível extrair informações importantes com esse comando.

Não deixe de usar também o **printVarSummary** para mostrar informações sobre a variável do arquivo.

Exemplo:

```
f = addfile (“../arquivos/2000tar_1000hpa.nc”,“r”)
```

```
print(f)
```

Outra forma de explorar o arquivo pode ser feita usando o **ncl_filedump**.

Uso: **ncl_filedump arquivo.nc** ou **ncl_filedump arquivo.grb**

Isso dependerá da extensão do seu arquivo.

Exemplo:

Caso você tenha um arquivo netCDF e deseja ver o conteúdo do mesmo, proceda da seguinte forma digitando no seu terminal Linux:

```
ncl_filedump 2000tar_1000hpa.nc
```

Com isso, será mostrado um resumo do seu arquivo.

3.11 Importando dados short ou byte e convertendo para float

Antes de abrir o seu arquivo, verifique se ele é do tipo **short** (usando o **ncl_file-dump** ou **ncdump -h**), caso positivo é bem provável que deva ser feita a conversão para **float** ou **double**. Isso dependerá do seu cálculo. Os dados do tipo **short** estão compactados e necessitam ser descompactados. Para isso, usa-se a função **short2flt**.

Exemplo: O arquivo **uwnd.mon.mean.nc** é do tipo short (visto com o **ncdump -h**). Foi preciso alterar para float para realizar os cálculos de interesse.

```
u = short2flt( a->uwnd )
```

3.12 Convertendo formatos para netCDF

Há uma ferramenta chamada **ncl_convert2nc** que converte dados para netCDF. Para maiores informações acesse o site http://www.ncl.ucar.edu/Document/Tools/ncl_convert2nc.shtml.

Exemplo:

```
ncl_convert2nc chuva.grb
```

Converte o arquivo **chuva.grb** para **chuva.nc**.

Esse é um dos possíveis exemplos. Consulte o link acima.

O CDO também converte para netCDF por meio do comando:

```
cdo -f nc copy chuva.grb chuva.nc
```

4 COMO OBTER AJUDA

Ao criar o seu script não entre em pânico, mas antes de qualquer coisa, reveja seu arquivo no que diz respeito a abertura de arquivos, conversão de formatos, se o caminho está certo, falta de vírgula, ponto, parênteses. Após fazer checar esses passos, tente entender o erro que é mostrado na tela para depois solicitar apoio a lista de discussão do NCL.

Faça parte da lista acessando a página:

http://www.ncl.ucar.edu/Support/email_lists.shtml

Solicitando a inscrição do seu e-mail a lista **ncl-talk@ucar.edu**.

Esse tópico tem como objetivo ensinar a buscar ajuda e solução de problemas. É impossível inserir aqui todo o conteúdo porque há muita informação. Pretendo mostrar o pouco que sei para resolver alguns problemas que surgem na hora de criar o script.

Ao acessar a página do NCL (<http://www.ncl.ucar.edu/index.shtml>) na parte superior à direita há **Support**. Os *links* que recomendo para quem está começando são:

<http://www.ncl.ucar.edu/Training>

<http://www.ncl.ucar.edu/Training/Workshops>

<http://www.ncl.ucar.edu/Training/Workshops/Scripts>

Nesses *links* há *scripts* prontos e explicação detalhada de cada um deles com os resultados e a possibilidade de realizar o *download* dos mesmos.

Outro *link* interessante é o http://www.ncl.ucar.edu/Document/Functions/list_alpha.shtml que contém os *links* abaixo:

http://www.ncl.ucar.edu/Document/Functions/list_alpha.shtml

<http://www.ncl.ucar.edu/Document/Functions>

http://www.ncl.ucar.edu/Document/Functions/list_type.shtml

Nesses *links* estão disponíveis todas as funções para manipulação e cálculo de diver-

tos parâmetros (climatologia, EOF, anomalia, correlação, dentre outros).

Um dos meus *links* preferidos é <http://www.ncl.ucar.edu/Applications/>. Nesse *link* estão todos os exemplos para geração dos mais variados tipos de gráficos e manipulação de dados.

Com essas dicas, o usuário é capaz de gerar e processar dados usando o NCL. É necessário explorar o site para buscar o que se deseja.

5 ACENTUAÇÃO EM NCL

Para inserir acentos em NCL no caso de figuras geradas em português, o Mateus Teixeira desenvolveu um esquema de acentuação que está disponível no link abaixo:

<http://mateus-teixeira.blogspot.com.br/2009/10/acentos-no-ncl.html>

Ou acessando o documento criado por ele disponível em:

http://www.ncl.ucar.edu/Document/Manuals/ncl_ptBR.pdf.

Para inserir um comentário em NCL utiliza-se o **ponto e vírgula (;)**. Isto é, tudo que estiver à direita do “;” será considerado como comentário.

Exemplo:

Vamos escrever **precipitação**, então vamos acentuar as letras **ã** e **ç**.

No corpo principal do seu script (entre o **begin** e o **end**), proceda da seguinte maneira:

atilde = “a~H-13V2F35~D~FV-2H3~” ; ã

cedil = “c~H-13F35~K~FH2~” ; ç

res@gsnCenterString = “Precipita”+cedil+atilde+“o”

Veja que é bem simples acentuar e não se preocupe que nos próximos tópicos serão mostrados exemplos reais de como acentuar em NCL. Isso foi apenas um exemplo.

6 SUPERSCRIPTS E SUBSCRIPTS

Em NCL você pode manipular *strings* de várias maneiras. Veja o link abaixo.

<http://www.ncl.ucar.edu/Applications/fcodes.shtml>

Os mais usados são: “**S**” (*superscripts*) e “**B**” (*subscripts*).

Vejamos alguns exemplos:

temperatura: $\sim\mathbf{S}\sim\sim\mathbf{N}\sim\mathbf{C} \Rightarrow \text{°C}$

velocidade: $\text{m s}\sim\mathbf{S}\sim\sim\mathbf{N}\sim \Rightarrow \text{m s}^{-1}$

radiação: $\text{W m}\sim\mathbf{S}\sim\sim\mathbf{N}\sim \Rightarrow \text{W m}^{-2}$

altura geopotencial em 200 hpa: $\text{zgeo}\sim\mathbf{B}\sim\sim\mathbf{N}\sim \Rightarrow \text{zgeo}_{200\text{hpa}}$

O $\sim\mathbf{N}\sim$ quer dizer para voltar ao modo normal, porque se deixarmos apenas o $\sim\mathbf{S}\sim$, tudo que vier após ficara no modo *superscript*.

7 CORES, FONTES, ESTILOS DE LINHA, PREENCHIMENTO, MARCADORES E PROJEÇÕES

Há um número considerável de cores disponíveis no NCL. Além dessa tabela, o usuário tem a liberdade de criar suas próprias cores. Vejam os links abaixo para visualizar as tabelas disponíveis.

http://www.ncl.ucar.edu/Document/Graphics/color_table_gallery.shtml

Há também a seleção pelo nome da cor:

http://www.ncl.ucar.edu/Document/Graphics/named_colors.shtml

Tipos de fontes: http://www.ncl.ucar.edu/Document/Graphics/font_tables.shtml

Estilos de linhas: <http://www.ncl.ucar.edu/Document/Graphics/Images/dashpatterns.png>

Padrão de preenchimento: <http://www.ncl.ucar.edu/Document/Graphics/Images/fillpatterns.png>

Tipos de marcadores: <http://www.ncl.ucar.edu/Document/Graphics/Images/markers.png>

Projeções disponíveis: http://www.ncl.ucar.edu/Document/Graphics/map_projections.shtml

8 GRÁFICOS

O objetivo deste tópico consiste em disponibilizar *scripts* prontos para que sejam alterados pelos usuários.

Cada usuário terá uma necessidade diferente na hora de gerar o seu gráfico, isto é, gráfico de linha, barra, com duas linhas, com dois eixos dentre outros. Os *scripts* serão comentados na sua maioria para facilitar o entedimento.

Dúvidas ou sugestões podem ser enviadas para o meu e-mail (jgmsantos@gmail.com) que terei o maior prazer em ajudar na medida do possível. Não sei tudo e não esqueçam de pedir ajuda na lista de discussão do NCL (ncl-talk@ucar.edu).

Os dados utilizados para gerar as figuras estão no formato netCDF e podem ser baixados em:

<https://dl.dropboxusercontent.com/u/8357888/homepage/ncl/arquivos.nc.tar.gz>.

Para descompactar digite no seu terminal `tar -zxvf arquivos.nc.tar.gz`.

A Figura 8.1 mostra alguns elementos que compõem um gráfico. As informações servem para gráficos 1D e 2D.

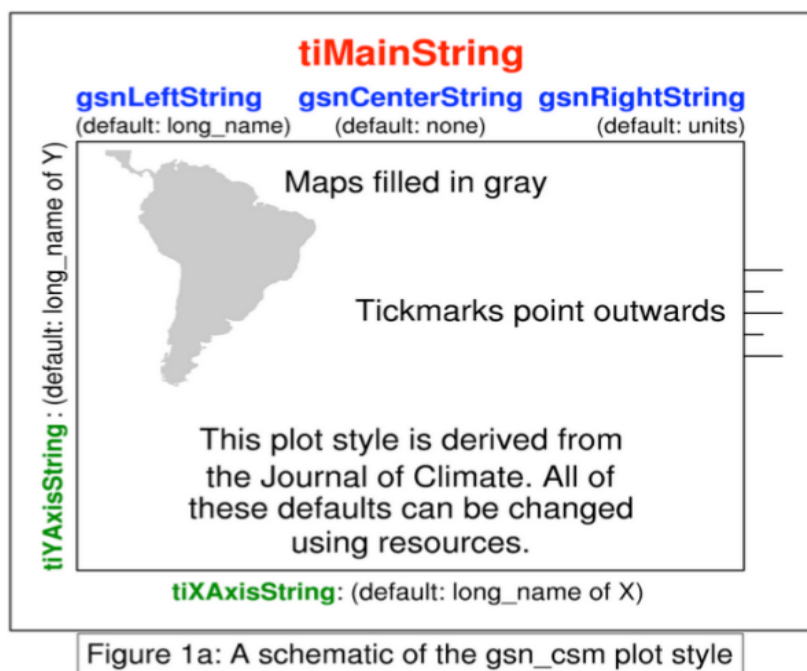


Figure 1a: A schematic of the gsn_csm plot style

Figura 8.1 - Elementos que compõem um gráfico.

8.1 Gráfico de linha

Os gráficos de linha podem ser acessados nos *links* abaixo. Com esses *links* acima o usuário será capaz de criar seus próprios *scripts* e gerar suas figuras.

<http://www.ncl.ucar.edu/Applications/xy.shtml>.

http://www.ncl.ucar.edu/Applications/gsn_xy.shtml.

Entendo que para gerar um gráfico de linha são necessárias algumas informações como o título do gráfico, o título para os eixos x e y, cor da linha, se terá marcadores ou não, a espessura da linha, o estilo de linha (contínua ou tracejada) e a legenda. Espero não ter omitido nenhuma informação.

Com *script* abaixo será criada uma série temporal de temperatura em Kelvin. O dado corresponde a dois anos de dados mensais, ou seja, apresenta 24 tempos.

Para executar o *script* digite no seu terminal Linux **ncl ex01.ncl**.

```
1 load "$NCARG_ROOT/lib/ncarg/nclscripts/csm/gsn_code.ncl"
2 load "$NCARG_ROOT/lib/ncarg/nclscripts/csm/gsn_csm.ncl"
3 4 begin
5 f = addfile ("../arquivos/tair.2011.2012.nc", "r")
6 u = short2flt( f->air )
7 wks = gsn_open_wks("png", "ex01")
8 plot = gsn_csm_xy(wks, ispan(1,12,1), u(0:11, {1000}, {-2}, {300}), False)
9 end
```

Descrição do primeiro *script*:

Nas linhas 1 e 2 são lidas as bibliotecas que ficam no início do *script*. Um *script* em NCL é composto por **begin** e **end** (linhas 4 e 9, respectivamente) e as instruções e comandos ficam entre eles.

Na linha 5 corresponde a abertura do arquivo para leitura "**r**". A linha 6 é a atribuição da variável do arquivo. Veja que foi usada a função `short2flt` para descompactar o dado. Isso foi feito porque o dado é do tipo **short**. Na linha 7 está a extensão e nome do arquivo que será gerado (ex01.png). A linha 8 é a criação do *plot* propriamente dito por meio da função `gsn_csm_xy`. Observe que foram selecionados os doze primeiros meses (**0:11**). Em NCL o primeiro índice inicia em 0 e não em 1.

A função `ispan` cria um vetor com 24 posições já que nosso dado é mensal composto de dois anos, ou seja, 24 meses.

Lembrando que serão mostrados todos os tempos (:), primeiro nível que corresponde a **1000 hPa**, latitude fixa em **-2** e longitude fixa em **300**. O **False** indica que não está sendo aplicada nenhuma formatação ao gráfico.

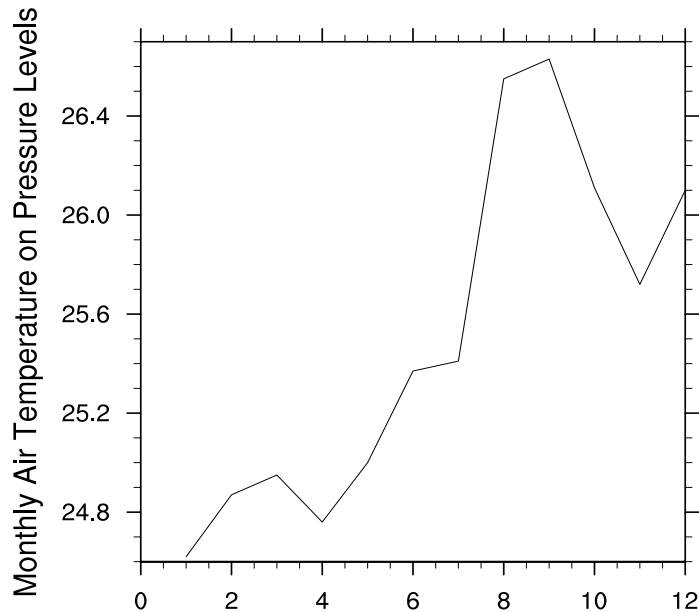


Figura 8.2 - Meu primeiro gráfico.

Observe que o gráfico não é o que desejamos, ele está bem bruto. Contudo, podemos formatar a figura habilitando o que se chama de recursos que pode conter valores lógicos *False* ou *True*. Na linha 8 do *script* acima, veja que há um *False*, como foi dito anteriormente, o gráfico não apresenta qualquer tipo de formatação.

Agora, vamos mudar esse recurso para *True* e começar a formatar a figura.

Inicialmente, vamos atribuir títulos a figura usando as funções `tiMainString`, `tiXAxisString` e `tiYAxisString`. Essas funções são responsáveis por incluir o título principal da figura e os títulos para os eixos x e y, respectivamente.

O *script* abaixo adicionará os títulos. Note que vou usar o mesmo *script* com a diferença que vou acrescentando informações para formatar o gráfico.

```
1 load "$NCARG_ROOT/lib/ncarg/nclscripts/csm/gsn_code.ncl"
2 load '$NCARG_ROOT/lib/ncarg/nclscripts/csm/gsn_csm.ncl'
```

```

3 load "$NCARG_ROOT/lib/ncarg/nclscripts/csm/contributed.ncl"
4
5 begin
6 ecirc = "e~H-13V2F35~C~FV-2H3~" ; ê = usado para acentuar o eixo x
7 f = addfile ("../arquivos/tair.2011.2012.nc", "r")
8
9 u = short2flt(f->air) ; tem que carregar o contributed.ncl para usar o short2flt
10
11 wks = gsn_open_wks("png", "ex02")
12
13 res = True ; res habilitado para formatar a figura
14 res@tiMainString = "Temperatura do Ar"
15 res@tiXAxisString = "M"+ecirc+"s"
16 res@tiYAxisString = "Temperatura (K)"
17
18 plot = gsn_csm_xy(wks, ispan(1,2,1), u(0:11, {1000}, {-2}, {300}), res)
19
20 end

```

Ao rodar o script (**ncl ex02.ncl**) será gerada a figura abaixo (Figura 8.3).

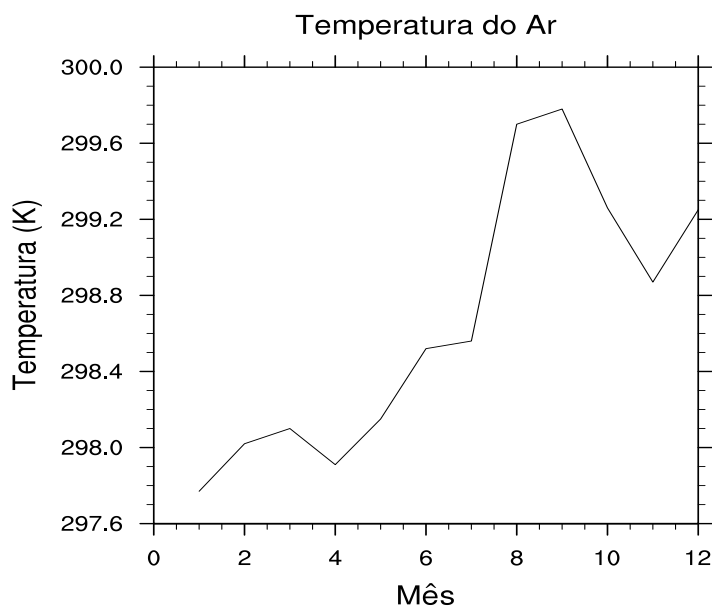


Figura 8.3 - Adicionando título ao gráfico.

Agora, vamos formatar os eixos x e y do gráfico.

```

1 load "$NCARG_ROOT/lib/ncarg/nclscripts/csm/gsn_code.ncl"
2 load "$NCARG_ROOT/lib/ncarg/nclscripts/csm/gsn_csm.ncl"
3 load "$NCARG_ROOT/lib/ncarg/nclscripts/csm/contributed.ncl"
4 ecirc = "e~H-13V2F35~C~FV-2H3~" ; ê = usado para acentuar o eixo x
5 begin
6
7 f = addfile ("../arquivos/tair.2011.2012.nc", "r")
8
9 u = short2flt(f->air)
10 u = u-273.15 ; converte de Kelvin para Celsius
11
12 wks = gsn_open_wks("png", "ex03")
13
14 res = True
15 res@tiMainString = "Temperatura em 2~S~o~N~S e 300~S~o~N~W"
16 res@tiXAxisString = "M"+ecirc+"s"
17 res@tiYAxisString = "Temperatura (~S~o~N~C)"
18 res@trXMinF = 1
19 res@tmXBMode = "Explicit"
20 res@tmXBValues = (/1,2,3,4,5,6,7,8,9,10,11,12/)

```

```

21 res@tmXBLLabels = (/“J”,“F”,“M”,“A”,“M”,“J”,“J”,“A”,“S”,“O”,“N”,“D”/)
22 res@trYMinF = 24.
23 res@trYMaxF = 27.
24
25 plot = gsn_csm_xy(wks,ispan(1,12,1),u(0:11,{1000},{-2},{300}),res)
26
27 end

```

Rode este script da seguinte forma no seu terminal Linux **ncl ex03.ncl**.

Na linha 19 o eixo x foi formatado de forma explícita. Há outras maneiras de formatação disponíveis [aqui](#). Na linha 20, como estamos usando apenas 12 meses, isso corresponde a posição no eixo x para cada mês e cada posição recebe uma string (linha 21). Nas linhas 22 e 23 são fixados os valores mínimo e máximo, respectivamente. O resultado desse *script* é a figura abaixo.

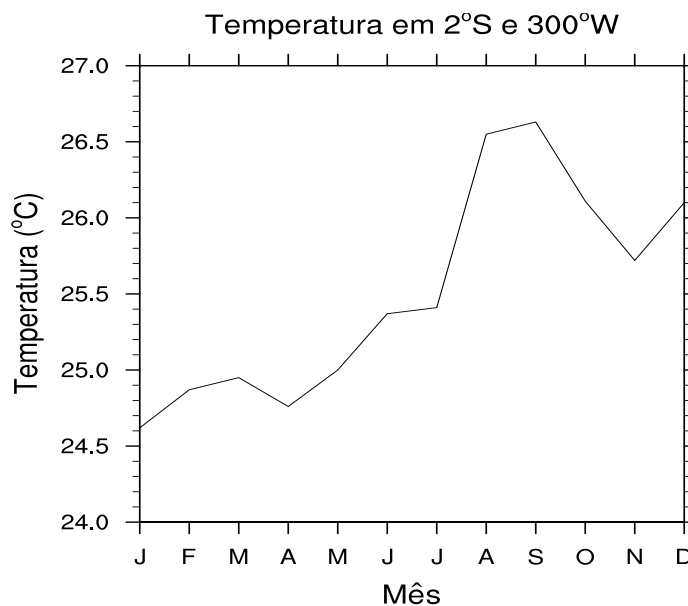


Figura 8.4 - Formatando os eixos x e y.

Após adicionar os títulos ao gráfico e formatar os eixos x e y, vamos personalizar a linha e adicionar uma legenda.

```

1 load "$NCARG_ROOT/lib/ncarg/nclscripts/csm/gsn_code.ncl"
2 load "$NCARG_ROOT/lib/ncarg/nclscripts/csm/gsn_csm.ncl"
3 load "$NCARG_ROOT/lib/ncarg/nclscripts/csm/contributed.ncl"

```



```

4 ecirc = "e~H-13V2F35~C~FV-2H3~" ; ê = usado para acentuar o eixo x
5 begin
6
7 f = addfile ("../arquivos/tair.2011.2012.nc", "r")
8
9 u = short2flt(f->air)
10 u = u-273.15
11
12 wks = gsn_open_wks("png", "ex04")
13
14 res = True
15 res@tiMainString = "Temperatura em 2~S~o~N~S e 300~S~o~N~W"
16 res@tiXAxisString = "M"+ecirc+"s"
17 res@tiYAxisString = "Temperatura (~S~o~N~C)"
18 res@trXMinF = 1
19 res@tmXBMode = "Explicit"
20 res@tmXBValues = (/1,2,3,4,5,6,7,8,9,10,11,12/)
21 res@tmXBLLabels = (/ "J", "F", "M", "A", "M", "J", "J", "A", "S", "O", "N", "D" /)
22 res@trYMinF = 24.
23 res@trYMaxF = 27.
24 res@xyLineThicknesses = 5.0
25 res@xyLineColors = "red"
26 res@xyMarkLineMode = "MarkLines"
27 res@xyMarkers = 16
28 res@xyMarkerColors = "red"
29 res@xyDashPattern = 15
30 res@xyMarkerSizeF = 0.02
31 res@pmLegendDisplayMode = "Always"
32 res@pmLegendSide = "Top"
33 res@pmLegendParallelPosF = 0.22
34 res@pmLegendOrthogonalPosF = -0.3
35 res@pmLegendWidthF = 0.15
36 res@pmLegendHeightF = 0.18
37 res@lgPerimOn = False
38 res@lgLabelFontHeightF = .02
39 res@xyExplicitLegendLabels = "Temperatura"
40

```

```
41 plot = gsn_csm_xy(wks, ispan(1,12,1), u(0:11, {1000}, {-2}, {300}), res)
42
43 end
```

Rode este script da seguinte forma no seu terminal Linux **ncl ex04.ncl**.

Nesse novo *script* adicionamos novas formatações referente à linha do gráfico. A [linha 24](#) é a espessura da linha, na [linha 25](#) é a cor da linha, na [linha 26](#) o usuário escolhe se deseja personalizar a linha, isto é, somente linha (*Lines*), linha com marcadores (*MarkLines*) ou apenas os marcadores (*Markers*), na [linha 27](#) o tipo de marcador selecionado já que escolhemos *Marklines*, na [linha 28](#) é a cor do marcador selecionado, na [linha 29](#) é o estilo de linha usado e na linha 30 está o tamanho do marcador que o usuário deseja.

As opções para personalizar a legenda vão da linha 31 a 39. A [linha 31](#) diz para mostrar a legenda, na [linha 32](#) a localização dela, na linha 33 é o deslocamento do legenda para direita (maiores valores) ou esquerda (menores valores), na linha 34 desloca para cima (maiores valores) ou para baixo (menores valores), nas linhas 35 36 referem-se a largura e altura da legenda, na linha 37 habilita (*True*) ou não (*False*) a caixa em volta da legenda, a linha 38 é o tamanho da fonte da legenda e na 39 é o nome que aparecerá na legenda. O usuário deve alterar os valores para suas necessidades.

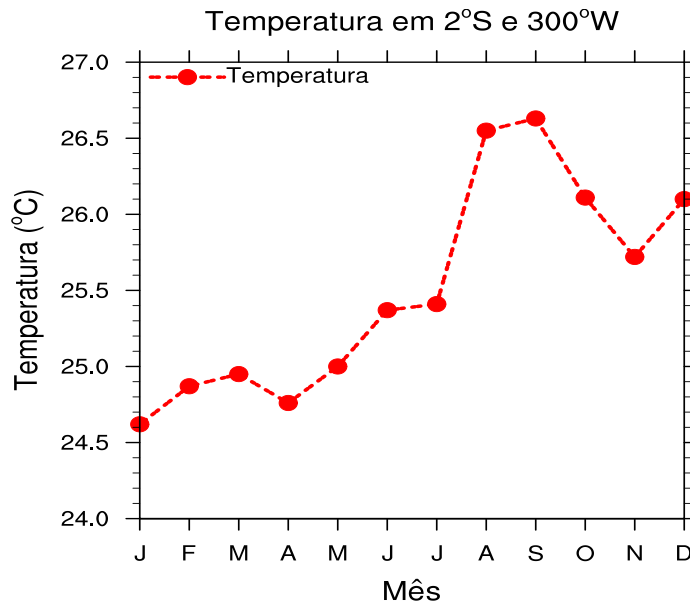


Figura 8.5 - Formatando a linha do gráfico e adicionando legenda.

Finalmente, geramos nosso primeiro gráfico de linha. É possível formatar ainda mais o gráfico, mas é impossível inserir todas as opções. O usuário deve pesquisar na página do NCL outras formatações de acordo com o seu interesse. Esses exemplos são básicos, mas já é suficiente para gerar figuras.

Não deixem de acessar os links abaixo para outras possibilidades de formatação de gráficos de linha.

<http://www.ncl.ucar.edu/Applications/xy.shtml>.

http://www.ncl.ucar.edu/Applications/gsn_xy.shtml.

8.2 Gráfico de barra

Para gerar gráficos de barra podem ser usadas as mesmas funções e opções utilizadas para o gráfico de linha. A diferença está em habilitar a função *gsnXYBarChart* para *True*. Vejamos o exemplo abaixo.

```
load "$NCARG_ROOT/lib/ncarg/nclscripts/csm/gsn_code.ncl"
load "$NCARG_ROOT/lib/ncarg/nclscripts/csm/gsn_csm.ncl"
load "$NCARG_ROOT/lib/ncarg/nclscripts/csm/contributed.ncl"
begin
f = addfile ("../arquivos/tair.2011.2012.nc","r")
```

```

u = short2fft(f->air)
u = u - 273.15
wks = gsn_open_wks("png","ex01")
res = True
res@gsnXYBarChart = True ; habilita gráfico de barras
plot = gsn_csm_xy(wks, ispan(1,12,1), u(0:11, {1000}, {-2}, {300}), res)
end

```

Ao executar o script acima será gerada a figura abaixo sem formatação.

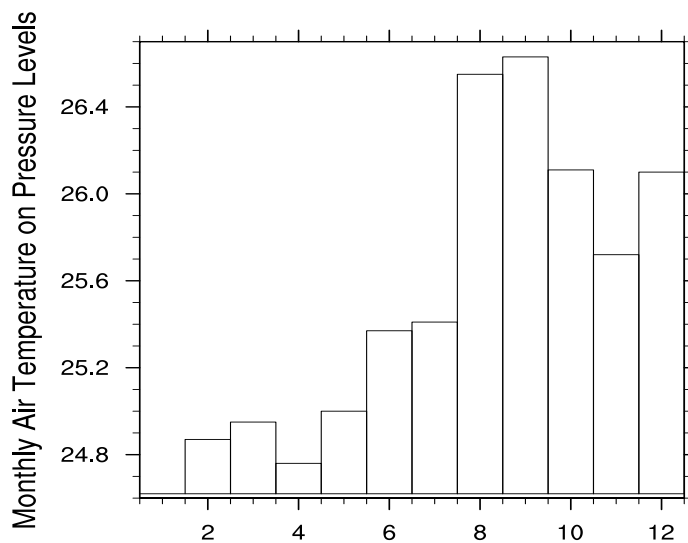


Figura 8.6 - Primeiro gráfico de barras.

Essa gráfico necessita de formatação, vejamos o exemplo abaixo para saber como ficará o gráfico com as devidas formatações.

```

load "$NCARG_ROOT/lib/ncarg/nclscripts/csm/gsn_code.ncl"
load "$NCARG_ROOT/lib/ncarg/nclscripts/csm/gsn_csm.ncl"
load "$NCARG_ROOT/lib/ncarg/nclscripts/csm/contributed.ncl"
begin
ecirc = "e~H-13V2F35~C~FV-2H3~" ; ê
f = addfile("../arquivos/tair.2011.2012.nc", "r")
u = short2fft(f->air)
u = u - 273.15
wks = gsn_open_wks("png","ex02")

```

```

res = True
res@gsnXYBarChart = True barras
res@trYMaxF = 28. ; máximo valor de y
res@trYMinF = 22. ; mínimo valor de y
res@gsnXYBarChartColors = "navy" ; cor da barra
res@gsnXYBarChartBarWidth = 0.60 ; espaçamento entre as barras
res@tmXBMode = "Explicit"
res@tmXBValues = (/1,2,3,4,5,6,7,8,9,10,11,12/)
res@tmXBLLabels = (/“J”,“F”,“M”,“A”,“M”,“J”,“J”,“A”,“S”,“O”,“N”,“D”/)
res@tiMainString = “Temperatura em 2~S~o~N~S e 300~S~o~N~W”
res@tiXAxisString = “M”+ecirc+“s”
res@tiYAxisString = “Temperatura (~S~o~N~C)”
plot = gsn_csm_xy(wks, ispan(1,12,1), u(0:11, {1000}, {-2}, {300}), res)
end

```

O resultado é a figura abaixo.

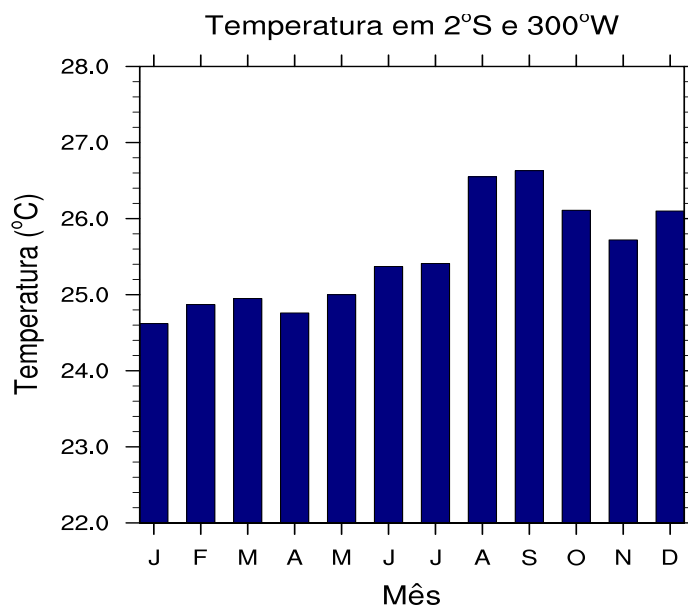


Figura 8.7 - Formatando gráfico de barras.

Agora sim, o gráfico tem outro aspecto.

Vários exemplos de gráficos de barras podem ser encontrados no link abaixo.

<http://www.ncl.ucar.edu/Applications/bar.shtml>.

Navegue pela página e veja qual é a melhor opção para você.

8.3 Gráfico espacial

Outro tipo de gráfico que será visto aqui é o do tipo lat/lon. A ideia é a mesma dos tópicos anteriores, isto é, mostrar exemplos prontos para que o usuário possa posteriormente pesquisar e criar seus próprios *scripts*.

Aqui vai uma dica, antes de abrir seu arquivo via NCL procure olhar como é a estrutura dele. Por exemplo, o usuário pode usar o `ncl_filedump` para isso.

Exemplo: Verificar a estrutura do arquivo `tair.2011.2012.nc` via `ncl_filedump`.

No seu terminal Linux digite: `ncl_filedump tair.2011.2012.nc`.

Na linha 32 terá a seguinte informação:

short air (time, level, lat, lon)

Isso quer dizer que a variável `air` é do tipo **short** com quatro dimensões (tempo, nível vertical, latitude e longitude), ou seja, ela precisa ser descompactada para visualizar corretamente os valores usando a função `short2flt`. Caso fosse *float* ou *double*, não haveria necessidade de descompactação.

Vamos ao nosso primeiro gráfico lat/lon.

```
load "$NCARG_ROOT/lib/ncarg/nclscripts/csm/gsn_code.ncl"
load "$NCARG_ROOT/lib/ncarg/nclscripts/csm/gsn_csm.ncl"
begin
f = addfile ("../arquivos/tair.2011.2012.nc","r")
t = f->air
wks = gsn_open_wks("png","ex01")
; plota o primeiro tempo (0), primeiro nível vertical (1000 hPa), todas as latitudes
(:) e longitudes (:)
plot = gsn_csm_contour_map_ce(wks,t(0,{1000},:,:),False)
end
```

O resultado é a figura abaixo.

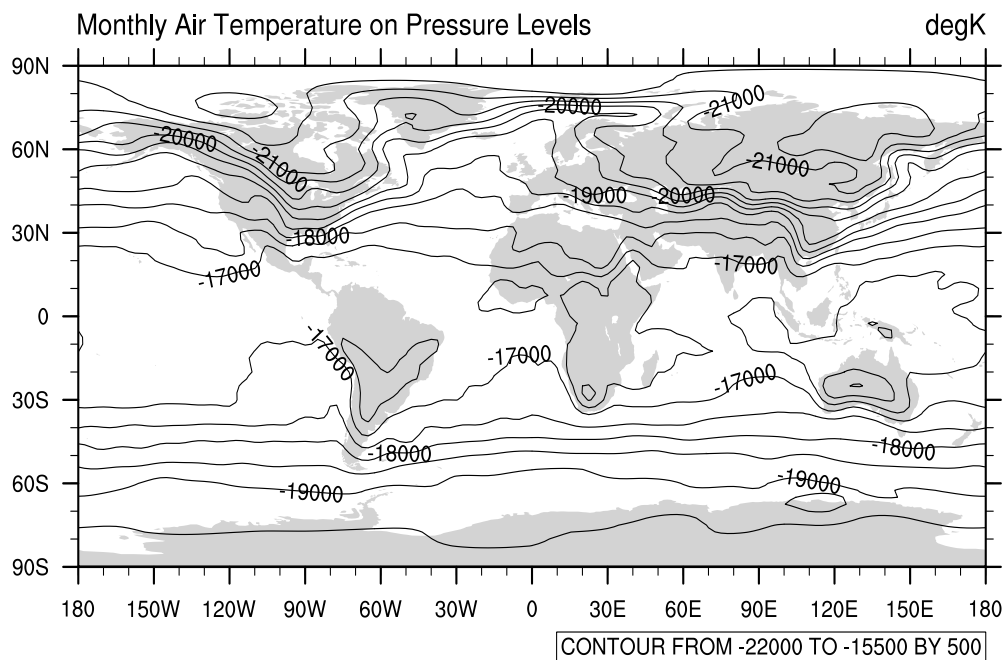


Figura 8.8 - Meu primeiro gráfico lat/lon.

Observe que os valores de temperatura (K) estão diferentes, isso acontece por a variável **air** está compactada, para descompactar, usamos a função **short2flt**. Veja o script abaixo que realiza esse tarefa.

```
load "$NCARG_ROOT/lib/ncarg/nclscripts/csm/gsn_code.ncl"
load "$NCARG_ROOT/lib/ncarg/nclscripts/csm/gsn_csm.ncl"
load "$NCARG_ROOT/lib/ncarg/nclscripts/csm/contributed.ncl"
begin f = addfile ("../arquivos/tair.2011.2012.nc", "r")
t = short2flt( f->air )
wks = gsn_open_wks("eps", "ex02")
plot = gsn_csm_contour_map_ce(wks, t(0, {1000}, :, :), False)
end
```

O resultado é a figura abaixo.

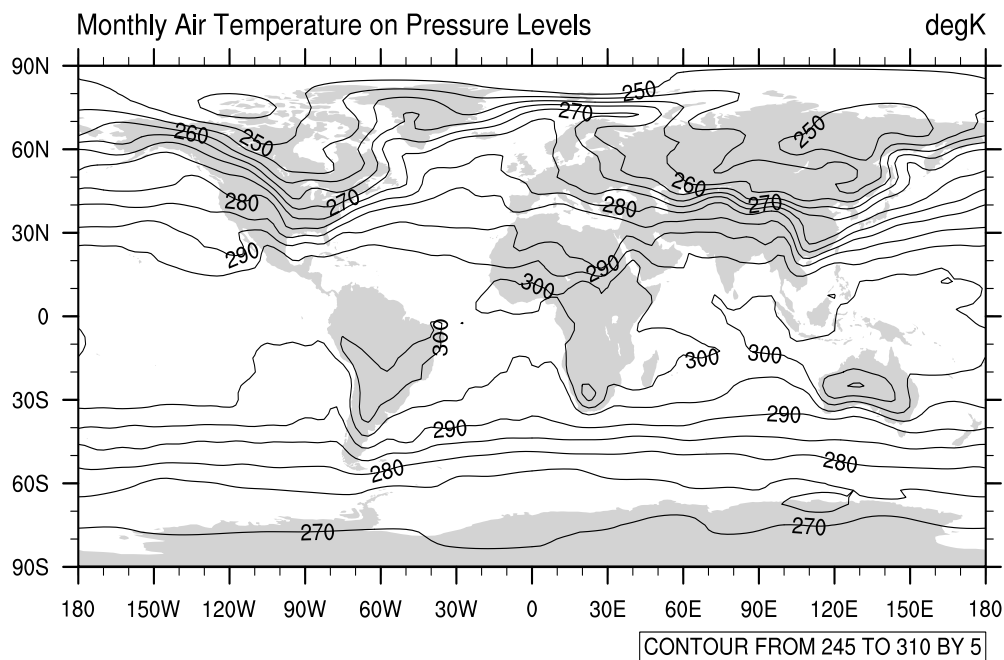


Figura 8.9 - Variável descompactada.

Agora sim! Os valores estão na ordem de grandeza de uma temperatura (K).

No script abaixo, vamos incluir títulos como também alterar o intervalo dos contornos.

```
load "$NCARG_ROOT/lib/ncarg/nclscripts/csm/gsn_code.ncl"
load "$NCARG_ROOT/lib/ncarg/nclscripts/csm/gsn_csm.ncl"
load "$NCARG_ROOT/lib/ncarg/nclscripts/csm/contributed.ncl"
begin
f = addfile("../arquivos/tair.2011.2012.nc","r")
t = short2flt( f->air )
t = t - 273.15
wks = gsn_open_wks("png","ex03")
res = True
res@tiMainString = "Temperatura em 1000 hPa" ; título principal da figura
res@gsnLeftString = "Fonte: NCEP/R2" ; título do lado esquerdo
res@gsnCenterString = "Janeiro" ; título central
res@gsnRightString = "~S~o~N~C"; título do lado direito
res@cnLevelSelectionMode = "ManualLevels" ; fixa os valores manualmente
res@cnMinLevelValF = -40. ; fixa o mínimo valor do contorno
```



```

res@cnMaxLevelValF = 40. ; fixa o máximo valor do contorno
res@cnLevelSpacingF = 10. ; fixa o espaçamento do contorno
res@cnInfoLabelOn = False ; desabilita informações do contorno
plot = gsn_csm_contour_map_ce(wks,t(0,{1000},:,:),res)
end

```

O resultado é a figura abaixo.

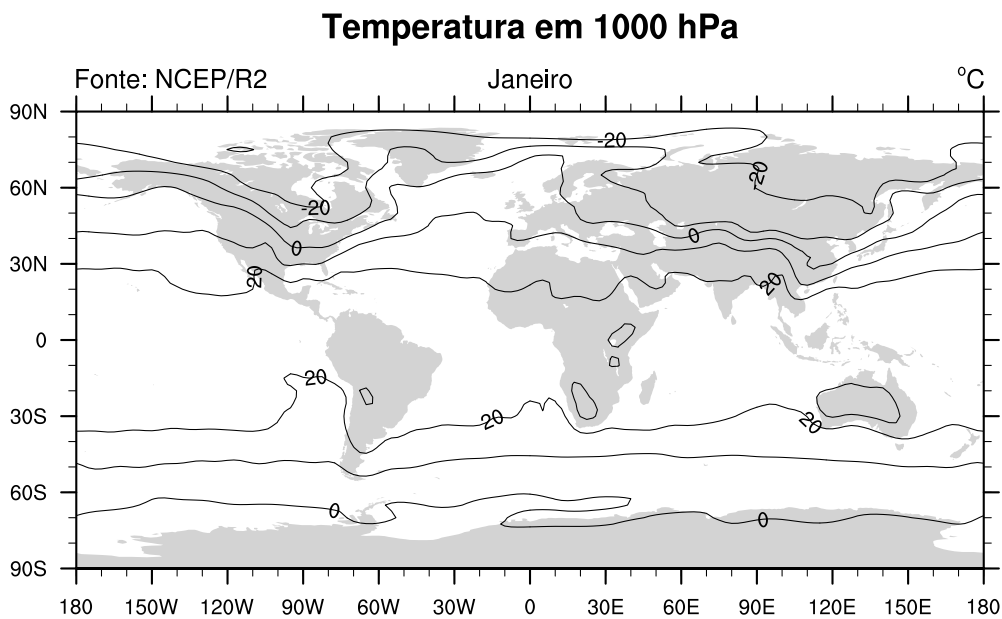


Figura 8.10 - Adicionando títulos e fixando valores máximo e mínimo.

Além do gráfico de contorno, às vezes, o usuário deseja visualizar o gráfico de forma preenchida ou *shaded*. O *script* abaixo é um bom exemplo de como realizar essa tarefa.

```

load "$NCARG_ROOT/lib/ncarg/nclscripts/csm/gsn_code.ncl"
load "$NCARG_ROOT/lib/ncarg/nclscripts/csm/gsn_csm.ncl"
load "$NCARG_ROOT/lib/ncarg/nclscripts/csm/contributed.ncl"
begin
f = addfile ("../arquivos/tair.2011.2012.nc","r")
t = short2flt( f->air )
t = t - 273.15
wks = gsn_open_wks("png","ex04")
; muda as cores de preenchimento. Veja a tabela de cores.

```

```

gsn_define_colormap(wks,"BlAqGrYeOrRe")
res = True
res@tiMainString = "Temperatura em 1000 hPa"
res@gsnLeftString = "Fonte: NCEP/R2"
res@gsnCenterString = "Janeiro"
res@gsnRightString = "~S~o~N~C"
res@cnLinesOn = False ; desabilita o gráfico de contorno
res@cnFillOn = True ; habilita o preenchimento do gráfico (shaded)
res@lbOrientation = "Horizontal" ; orientação da legenda (Horizontal ou Vertical)
plot = gsn_csm_contour_map_ce(wks,t(0,{1000},,:),res)
end

```

O resultado é a figura abaixo.

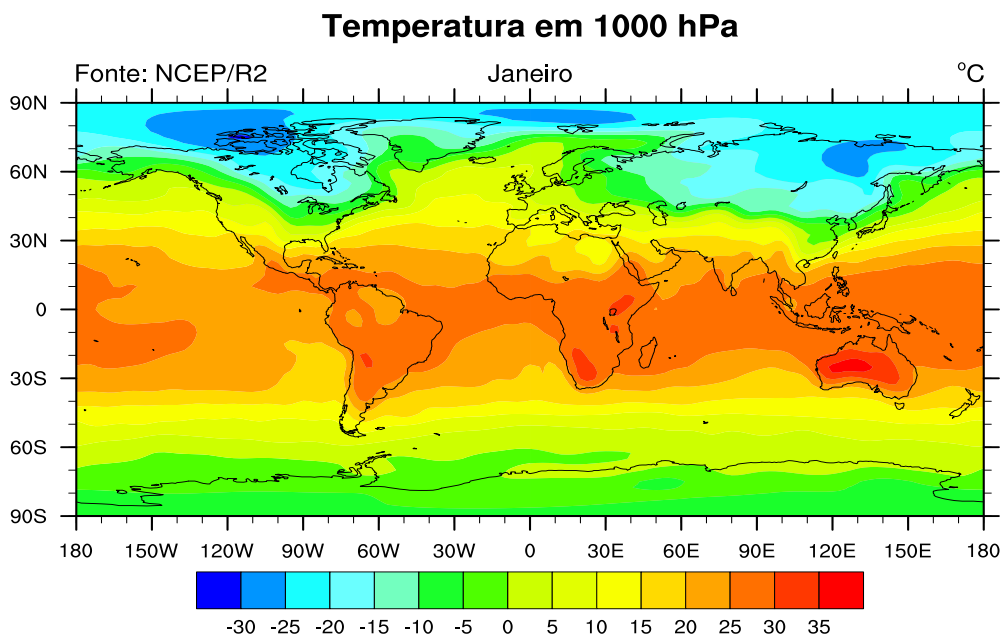


Figura 8.11 - Gráfico preenchido (*shaded*).

Outro exemplo é a geração de mapas do tipo vetor do vento (*vector*). O exemplo abaixo mostra uma figura sobre a América do Sul no nível de 200 hPa para o mês de dezembro.

```

load "$NCARG_ROOT/lib/ncarg/nclscripts/csm/gsn_code.ncl"
load "$NCARG_ROOT/lib/ncarg/nclscripts/csm/gsn_csm.ncl"

```

```

load "$NCARG_ROOT/lib/ncarg/nclscripts/csm/contributed.ncl"
begin
a = addfile("../arquivos/u.2011.2012.nc","r")
b = addfile("../arquivos/v.2011.2012.nc","r")
u = short2fft( a->uwnd )
v = short2fft( b->vwnd )
vel = sqrt( u^2+v^2 )
copy_VarCoords(u,vel)
wks = gsn_open_wks("png","ex05")
gsn_define_colormap(wks,"BlAqGrYeOrReVi200")
res = True
res@tiMainString = "Vetor velocidade do vento em 200 hPa"
res@gsnLeftString = "NCEP/R2"
res@gsnCenterString = "Dezembro"
res@gsnRightString = "m s~S~-1"
res@vpXF = 0.1 ; posição x onde será desenhada a figura
res@vpYF = 0.85 ; posição y onde será desenhada a figura
res@gsnSpreadColors = True ; use full colormap
res@gsnSpreadColorEnd = 193 ; last color to use
res@gsnSpreadColorStart = 6 ; first color to use
res@gsnMajorLonSpacing = 10 ; variação da escala do eixo x
res@gsnMajorLatSpacing = 10 ; variação da escala do eixo y
res@vcRefMagnitudeF = 15.0 ; magntiude de vetor
res@vcRefLengthF = 0.050 ; comprimento do vetor
res@vcGlyphStyle = "CurlyVector" ; efeito de curvatura nos vetores
res@vcMinDistanceF = 0.01 ; espessura dos vetores
res@vcRefAnnoOrthogonalPosF = -1.0 ; deslocamento da legenda da velocidade
res@mpMinLonF = -90. ; longitude oeste
res@mpMaxLonF = -30. ; longitude leste
res@mpMinLatF = -60. ; latitude sul
res@mpMaxLatF = 20. ; latitude norte
res@mpOutlineBoundarySets = "National" ; mostra divisao dos países
res@mpDataSetName = "Earth..4"
res@mpDataBaseVersion = "MediumRes"
res@mpOutlineOn = True
res@mpOutlineSpecifiers = (/ "Brazil:states" /) ; divisão do estados brasileiros
res@mpFillOn = False ; mapa com fundo branco

```

```

res@lbOrientation = "Vertical" ; orientação da legenda (Horizontal ou Vertical)
res@vcLevelSelectionMode = "ExplicitLevels" ; fixa o nível explicitamente
res@vcLevels = ispan(10,40,2) ; níveis de velocidade da legenda
plot=gsn_csm_vector_scalar_map_ce(wks,u(11,{200},:,:),v(11,{200},:,:),vel(11,{200},:,:),res)
end

```

O resultado é a figura abaixo.

Vetor velocidade do vento em 200 hPa

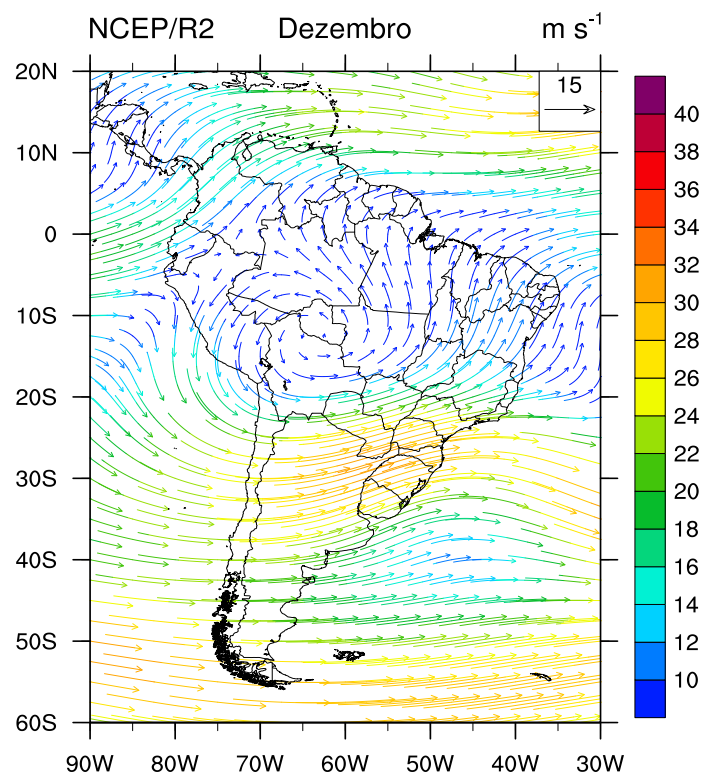


Figura 8.12 - Gráfico de vetor.

9 CONSIDERAÇÕES FINAIS

Esse tutorial sobre NCL veio em boa hora porque há pouco ou quase nada de material disponível na língua portuguesa. No Brasil esse *software* ainda tem um público muito pequeno que o utiliza e acredito que com a sua difusão, em breve teremos bastante usuários usando essa ferramenta tão poderosa.

É sempre bom ter algo no nosso idioma, mas às vezes, temos que entrar em modo *hard* para aprender algo novo, em outro idioma e ainda mais sozinho. Espero que vocês gostem desse material.

É sempre bom ter o retorno de quem usa qualquer material, pois desta forma pode-se melhorá-lo cada vez mais. Sinta-se à vontade para entrar em contato com o autor.

10 LINKS IMPORTANTES

Site que contém esse tutorial: <https://sites.google.com/site/jgmsantos/>.

Site oficial: <http://www.ncl.ucar.edu/>.

Tutorial em português: http://www.ncl.ucar.edu/Document/Manuals/ncl_ptBR.pdf.

Manual para gráficos: http://www.ncl.ucar.edu/Document/Manuals/graphics_man.pdf.

Manual oficial do NCL: http://www.ncl.ucar.edu/Document/Manuals/language_man.pdf.

REFERÊNCIAS BIBLIOGRÁFICAS

The NCAR Command Language (Version 6.1.2) [Software]. (2013). Boulder, Colorado: UCAR/NCAR/CISL/VETS. <http://dx.doi.org/10.5065/D6WD3XH5>