

Introdução ao Grads Grads Script e Funções

MSc. José Leandro Campos

Sumário

- Introdução
- Primeiro Script
- Funções
- Loops no Grads
- If e else no Grads

Introdução

- Até agora aprendemos a utilizar a linha de comando para fazer cálculos e figuras;
- Muitas vezes é necessário fazer esses cálculos ou figuras mais de uma vez;
- Não é prático digitar ou fazer o “copy & paste” n vezes;
- Assim nessa aula iremos aprender a criar scripts que fazem essas figuras ou cálculos quase que automáticos.

Primeiro Script

- No OpenGrads os scripts tem uma extensão com o nome “.gs” (grads script);
- Para criar um script basta criar um arquivo ascii com extensão “.gs”;

Script exemplo1.gs

```
* ***** *
* Script criado para a aula de Meteorologia Sinótica 1 *
* 01/04/2016 *
* ***** *

*abrindo o arquivo de dados
'reinit'
'open rain.ct1'

* definindo o display
'set display color white'
'c'
'set gxout shaded'

* definindo as dimensões
'set time 30JAN2010'
'set lon 270 330'
'set lat -60 15'

* definindo outros ....
'set mpdset hires'
'set grads off'
'set grid off'

* plotando a precipitação
'set clevs 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20'
'd rain'
'cbarn'

'set gxout contour'
'set clevs 0 4 8 12 16'
'set clab off'
'set ccolor 15'
'd rain'

* Escrevendo o título e salvando a figura
'draw title CPC daily precipitation (mm/day)'
'printim rain.CPC.png'
```

- Para escrever comentários coloque um “*” antes do comentário na primeira coluna do arquivo;
- Use ‘reinit’ para tirar da memória outras variáveis que não são de interesse;
- Para rodar esse script simplesmente digite na linha de comando:
ga> exemplo1
- **Atividade 1 - Crie um script para plotar a precipitação no dia do seu aniversário!**

Funções

- Na atividade anterior, plotamos a precipitação no dia do nosso aniversário;
- Seria interessante poder plotar a precipitação qualquer dia do ano;
- Para isso, criaremos uma função!

exemplo2.gs

```
*****  
* Script criado para a aula de Meteorologia Sinótica 1 *  
* 01/04/2016 *  
*****  
  
function plot_rain(args)  
* Para passar o argumento da função para o grads, basta  
* usar o comando subwrđ dessa forma, sem as aspas  
  date = subwrđ(args,1)  
  
  'reinit'  
  'open rain.ctl'  
  
  'set display color white'  
  'c'  
  'set gxout shaded'  
  
* O argumento da função deve ser usado dessa forma, sem as  
* aspas, observe o "date".  
  'set time 'date'  
  
  'set lon 270 330'  
  'set lat -60 15'  
  
  'set mpdset hires'  
  'set grads off'  
  'set grid off'  
  
  'set clevs 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20'  
  'd rain'  
  'cbarn'  
  
  'set gxout contour'  
  'set clevs 0 4 8 12 16'  
  'set clab off'  
  'set ccolor 15'  
  'd rain'  
  
  'draw title CPC daily precipitation (mm/day)'  
  
* O argumento da função deve ser usado dessa forma, sem as  
* aspas, observe o "date"  
  'printim 'date'.rain.CPC.png'  
  
return
```

- Uma função no grads sempre tem que começar com “function nome_funcao(args)”;
- Para passar os argumentos para a função, use “var = subwrđ(args,#)”;
- Para chamar a função digite na linha de comando:
 - `ga> exemplo2 01FEB2010`
- Veja que ele plotou a precipitação na data que foi escrita após o exemplo2, a precipitação no dia 02 de fevereiro de 2010.
- **Atividade 2 – Crie uma função que plote somente o contorno da precipitação em uma dada data a ser determinada.**

Loops no Grads

- Queremos plotar a precipitação de vários dias em sequência;
- É muito trabalhoso ficar digitando na linha de comando;
- Para isso, criaremos um script, que chame a função exemplo2 e plote a precipitação para as datas que determinarmos, usando um loop.

exemplo3.gs

```
'reinit'  
  
dia_inicio = 1  
dia_final = 31  
  
dia = dia_inicio  
WHILE( dia <= dia_final )  
  'exemplo2 'dia'JAN2010'  
  dia = dia + 1  
ENDWHILE
```

Atividade 3 – Melhore os exemplos 3 e 2, fazendo com que no exemplo 3 ele abra o arquivo ctl só que comece plotando a precipitação do dia 31 ao dia 1.

- No grads o loop é construído dessa forma:
WHILE(...)
 (...)
ENDWHILE
- As condições lógicas do loop podem ser:
 - > maior
 - < menor
 - = igual
 - /= diferente
 - >= maior ou igual
 - <= menor ou igual

IF e Else no Grads

- A função exemplo2, está plotando os shadeds mais os contornos de precipitação;
- Queremos plotar somente shaded ou somente os contornos;
- Para isso utilizaremos o comando IF em nossa função.

```
function plot_rain(args)
date = subwrd(args,1)
opt = subwrd(args,2)
'reinit'
'open rain.ct1'
'set display color white'
'c'
'set time 'date'
'set lon 270 330'
'set lat -60 15'
'set mpdset hires'
'set grads off'
'set grid off'
IF( opt = 1 )
'set gxout shaded'
'set clevs 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19
ELSE IF( opt = 2 )
'set gxout contour'
'set clevs 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19
ENDIF
'd rain'
IF( opt = 1 )
'cbarn'
ENDIF
'draw title CPC daily precipitation (mm/day) \'date'
'printim 'date'.rain.CPC.ex4.png'
return
```

- As condições lógicas do “if” e do “else if”, são as mesmas do “while”;
- Para chamar essa função, basta digitar:
 - ga> exemplo4 05MAY2010 1
 - ga> exemplo4 05MAY2010 2
- **Atividade 4 – Há meses que tem 31, 30 e 28 ou 29 dias. Transforme o exemplo3 em função, use o comando if para selecionar os meses que tem 30 e 31 dias e faça o programa plotar a precipitação durante um ano inteiro.**