N126 - Matemática Computacional
Prática 1 - 21/3/2014

Inicie um computador com uma instalação de Octave 3.8.1

• Descarregue os ficheiros de código associados a esta prática, e guarde-os numa pasta que não exija procurar muito na árvore do diretório. Tente que a pasta não tenha símbolos "duvidosos" no nome (espaços ou caracteres especiais). Neste documento trataremos essa pasta por `[DirTrab]`. A pasta com a instalação de Octave 3.8.1 será tratada por `[path]`.

• Entre em Octave através do atalho GUI ("Graphical Users Interface"). Se necessário, terá de o executar através de `[path]/bin/octave-gui.exe` (Conselho: Será sempre melhor ter acesso a um atalho para poder abrir este ficheiro com comodidade a partir do ambiente de trabalho).

No Octave(gui) existe a "Janela de Comandos", a principal, onde serão introduzidas as instruções Octave. Existe outra janela denominada "Ambiente de Trabalho", onde se pode visualizar as variáveis guardadas na memória do Octave e, ainda, uma janela denominada "Navegador de ficheiros", onde se pode ver os ficheiros que podem ser abertos ou guardados diretamente onde está aberto.

• Use a janela do Navegador de ficheiros para visualizar os conteúdos da pasta [DirTrab]. Abra os ficheiros "*.m" nesta pasta. Os ficheiros "*.m" podem ser definições de funções ou podem ser scripts de Octave.

- Um script Octave é uma sequência de comandos que pretendemos que Octave execute.

- Uma função é um procedimento programado em Octave, um comando novo, feito por nós, e que pode ser invocado a partir da Janela de Comandos ou a partir de qualquer script.

A janela de comandos passará a mostrar o editor, onde pode ver e alterar o conteúdo dos ficheiros *.m. Não altere os ficheiros da prática. Se pretende fazer modificações, guarde os ficheiros modificados com um nome novo.

Nos ficheiros *.m, as linhas que começam por # não são interpretadas como comandos Octave, são linhas com comentários do programador que o intérprete de comandos Octave vai ignorar.

• Na janela de "Ambiente de Trabalho" abra a pasta `[Dirtrab]` para indicar ao Octave que a pasta [DirTrab] é a sua pasta de trabalho. (Se o tiver de fazer através de comandos terá de fazer `cd(‘‘[Dirtrab]’’) [RETURN]` na Janela de Comandos.)

• Escreva `addpath(‘‘[Dirtrab]’’) [RETURN]` para indicar ao Octave que procure também comandos na pasta [DirTrab], para além das pastas próprias do Octave.

• Execute o `N126Pratica1.m`. (Pode também fazê-lo escrevendo `N126Pratica1` na Janela de comandos, para executar o script do ficheiro 'N126Pratica1.m'.)

Observe o código dos diferentes scripts e funções e as explicações dadas em cada script e função. Para interpretar a sintaxe dum comando de Octave, pode abrir a documentação, situada em "[path]/share/doc/octave/octave.pdf". Também pode escrever `help [comando]` na Janela de Comandos.

Os alunos, com ajuda do Octave deverão aplicar os códigos que aprenderam nos exercícios indicados abaixo para elaborar **um relatório com a resposta a todas as questões efetuadas.**. Usará o seu número de aluno, **[NUMaluno]**, em alguns destes exercícios.

1. <u>Números inteiros e números em ponto flutuante</u>

   Observe os resultados mostrados no script "N126Pratica1".
   Indique, para variáveis de tipo "double":

   (a) **Quantos bits são reservados para o expoente?**

   (b) **Qual é o expoente máximo e o expoente mínimo dos números guardados na máquina?**

   (c) **Que números reais produzem um underflow?q**

   (d) **Que números reais representam um overflow?**

   (e) **Quantos bits são reservados para a mantissa?**

   (f) **Dê uma razão pela qual o número** $1.9999999999999999 \times 2^{1023}$ **não é representável?**

   Posteriormente, averígue como são guardados os valores $-3, 3, 4, -4, 6, -6, 8, -8$ em formato tipo 'double' e em tipo 'int8'. **Depois de comparar essa forma de guardar, responda aos seguintes itens:**

   (f) **No formato 'double' o número inteiro 3 é guardado da mesma forma que o inteiro** 3 **usado num expoente (por exemplo** 2^3**). Explique como são os dois processos de guardar inteiros e de guardar expoentes.**

   (g) **Explique como se guardam os números negativos nos dois tipos apresentados.**

   (h) **Escreva todos os bits quando representa o seu número de corpo [NUMaluno] nos dois formatos apresentados. Justifique os valores que obteve.**

   Veja a documentação sobre as instruções `flintmax`, `realmax`,`eps`, `realmin`.

   (i) **Atendendo ao que realizou nas alíneas anteriores explique os valores devolvidos por estas instruções.**

2. <u>Aritmética em base 5: Arredondamentos.</u>

   As funções `intoal` e `altoin` permitem introduzir um inteiro e uma base de modo a obter a lista de algarismos correspondentes nessa base e introduzir uma lista de algarismos e uma base de modo a obter o inteiro nessa base, respetivamente.

   A função `arred` permite introduzir um número real (em ponto flutuante, tipo 'double') e devolve como resultado outro real (tipo 'double') depois de fazermos um arredondamento com p algarismos em base b ao valor introduzido.

   Execute o seguinte script:

   ```
   p=2;b=2;x=linspace(1,23.5,10);xa=arred(x,p,b);disp(x);disp(xa)
   ```

   Neste script são fixados $p = 2$ algarismos e a base $b = 2$. É criada uma sequência homogénea de 10 números desde 1 até 23.5, $x = [1, 3.5, 6, 8.5, ..., 23.5]$. Posteriormente, com a função 'arred' cria a sequência arredondada com p algarismos em base $b = 2$. No fim, são mostradas as duas sequências, a original e a aproximada.

   Justifique os valores arredondados obtidos.

   Modifique o script para criar uma sequência homogénea de 3 números, entre 1 e `[NUMaluno]`. Escreva estes valores e os arredondamentos em base 5 com 2 algarismos. Justifique os três valores determinados como arredondamento.

3. Aritmética em base 7: Precisão e erro propagado.

Imagine uma máquina que usa base 7 para guardar números, com mantissas de 2 algarismos.

Qualquer número x será guardado na memória com a versão arredondada x^*. **Indique um majorante do erro relativo $\delta(a, a^*)$ que existe entre qualquer número a e o seu arredondamento a^* nesta máquina.**

Se queremos calcular uma função $f(x)$, não conseguimos usar x mas sim x^*. No entanto a máquina não pode calcular $f(x^*)$ porque não consegue representar este número. O melhor que seria capaz de fazer era obter o valor arredondado $f^*(x^*)$ do autêntico $f(x^*)$.

Portanto nesta máquina, se queremos fazer o gráfico da função $\sin(x)$ no intervalo [-3,3], criamos uma lista de 5000 valores `x=linspace(-3,3,5000)`. Esta lista entra na memória como arredondada `xa=arred(x,2,7)`. O cálculo de $\sin(x)$ nesta lista arredondada exigiria determinar `fxa=sin(xa)`, e o melhor que a máquina consegue fazer é dar os valores arredondados `afxa=arred(fxa,2,7)`. Mostremos o gráfico de $\sin(x)$ no intervalo $[-3, 3]$, segundo esta máquina (gráfico dos valores `afxa` associados a cada $x \in [-3, 3]$):

```
p=2; b=7; x=linspace(-3,3,5000); xa=arred(x,p,b); fx=sin(x);
fxa=sin(xa); afx=arred(fx,p,b); afxa=arred(fxa,p,b); plot(x,afxa)
```

O gráfico (ver janela gráfica) é similar ao gráfico de $f(x) = \sin x$. Mas para visualizar os dois gráficos ao mesmo tempo, execute depois de fechar a primeira janela gráfica:

```
hold on; plot(x,fx)
```

Parece que os erros entre $f(x)$ e o valor $f^*(x^*)$ são mais grosseiros para valores grandes de x. No entanto, observemos o erro relativo, e não o erro absoluto:

```
delta=(fx-afxa)./fx; plot(x,delta)
```

Vemos que o erro relativo $\delta(f(x), f^*(x^*))$ está dentro duns limites aceitáveis, para $x \in [-3, 3]$. Faça o mesmo estudo para o intervalo $[-4, 4]$. Observe o gráfico de $f^*(x^*)$, similar ao que tínhamos, e observe o gráfico do erro relativo, bastante diferente do que tínhamos.

Para $x = 3.141$, determine com Octave quem são $f(x)$, x^*, $f(x^*)$, $f^*(x^*)$. Determine os erros relativos $\delta(x, x^*)$, $\delta(f(x^*), f^*(x^*))$. Estabeleça se existe relação entre estes erros relativos e os valores $p = 2$, $b = 7$ da nossa máquina.

Calcule o número de condição $\kappa = \left|\frac{a \cdot f'(a)}{f(a)}\right|$ da função $f(x) = \sin x$ no ponto $a = 3.141$. Prove com ajuda do desenvolvimento de Taylor de grau 1 em δ, que para qualquer δ próximo de 0, se verifica:

$$\delta(f(a), f(a \cdot (1 + \delta))) \simeq |\delta| \cdot \left|\frac{a \cdot f'(a)}{f(a)}\right|$$

Estabeleça o erro relativo $\delta(f(x), f^*(x^*))$ em $x = 3.141$ e mostre que é próximo a 311%. Como justifica que este erro possa ser tão elevado, sendo a máquina dada por $p = 2$, $b = 7$?

Problema final: Veja a representação no computador da operação

$$\frac{2^{2^{10}}}{2^{2^{10}}}$$

e a de

$$2^{(2^{10}-2^{10})}$$

que deveria ser a mesma. Porque razão é que não dão o mesmo valor?