

Análise Numérica

Ficha de exercícios N°1 - Representação de Números. Mudanças de base. Matlab.

NOTA INTRODUTÓRIA: muitos dos exercícios desta ficha de exercícios e das que se lhe seguirão serão resolvidos com recurso ao sistema computacional Matlab (**Matrix Laboratory**) que os alunos usarão durante as aulas de natureza prática da disciplina de Análise Numérica num dos laboratórios de computação do Departamento de Matemática. Será feita uma breve introdução ao Matlab e os alunos familiarizar-se-ão com o sistema à medida que o forem utilizando na resolução dos exercícios propostos.

1. a) Para cada ponto x , o valor do polinómio $p(x) = 3x^4 + 7x^3 - x^2 - 4x - 4$ pode ser calculado com apenas 4 multiplicações e 4 adições / subtrações. Explique como.
- b) Generalize o método anterior (algoritmo de *Horner*) para o cálculo do valor do polinómio

$$p_n(x) = a_1x^n + a_2x^{n-1} + \dots + a_nx + a_{n+1}$$

num ponto x .

- c) Escreva uma função no Matlab **px=horner(a,x)** que é uma implementação do algoritmo anterior. O polinómio é dado na forma de um vector dos coeficientes $a = [a_1 \ a_2 \ \dots \ a_{n+1}]$ (por ordem decrescente dos graus).
 - d) Use a função **horner** para calcular o valor do polinómio $p(x) = 3x^4 + 7x^3 - x^2 - 4x - 4$ para $x = 1.5$.
 - e) A função **polyval** do Matlab pode ser usada para o mesmo efeito. No Matlab execute **polyval([3,7,-1,-4,-4],1.5)** para confirmar o resultado anterior.
2. a) Use as funções **horner** e **polyval** para encontrar as representações decimais dos números $(1000)_2$ e $(110101011)_2$ (isto é, os números cujas representações na base 2 são 1000 e 110101011).
 - b) Use a função **bin2dec** do Matlab para comprovar os resultados obtidos.
 - c) Use as funções **hex2dec** e **base2dec** do Matlab para obter a representação na base 10 dos números $(A10F9)_{16}$, $(2356504)_7$ e $(2356504)_8$ (nota: estas funções do Matlab funcionam só para números inteiros positivos no superiores a 2^{52}).
 - d) Use a função **horner** para confirmar os resultados da alínea anterior.
3. a) Escreva uma função no Matlab **bits=intdectobin(N)** para fazer a conversão de um inteiro positivo $(N)_{10}$ para a base 2. A função deve ter como parâmetro de entrada o inteiro $(N)_{10}$ e como parâmetro de saída um vector com os bits da representação binária do número N . Poderão ser úteis as funções **rem** e **floor** do Matlab.
 - b) Teste a sua função na conversão de algumas representações decimais à sua escolha e confirme que os resultados estão correctos reconvertendo as representações à base 10.
4. a) As funções **horner** e **polyval** podem ainda ser usadas para fazer a conversão da representação da parte fraccionária de um número não-inteiro positivo na base $b \geq 2$ para a base decimal. De que maneira? (Observação: recorde-se que a parte fraccionária da representação na base b corresponde às potências de expoente negativo e $b^{-k} = (1/b)^k$).

- b) Obtenha as representações decimais de $(0.1011)_2$, $(0.1101)_2$, $(0.111111101)_2$ e $(10.111111101)_2$.
5. a) Escreva uma função no Matlab **bits=fracdectobin(N)** para fazer a conversão da representação decimal de um número fraccionário para a base binária.
b) Teste a sua função na conversão dos números $(0.125)_{10}$, $(0.1)_{10}$ e $(0.1217)_{10}$.
6. a) Construa uma *script* em Matlab que calcula o maior inteiro positivo k tal que $1 + 2^{-k} > 1$.
b) Compare o valor de 2^{-k} com o valor da constante **eps** do Matlab.
7. Se um certo número x está entre o 15 e o sucessor deste, qual é o majorante do erro $|x - fl(x)|$ se o arredondamento para o mais próximo for utilizado? Justifique.
8. a) Utilize o *help* do Matlab para obter informação sobre as constantes **realmax** e **realmin**.
b) Tendo em atenção que está a trabalhar num sistema de norma IEEE754, justifique os valores de **realmax** e **realmin** obtidos.
9. a) Determine a maior potência de 10 que é representável no Matlab sem ocorrência de “overflow”.
b) Determine o maior valor inteiro de k tal $10^{-k} > 0$ no Matlab.
10. a) As funções **round**, **fix**, **ceil** e **floor** arredondam um dado número para inteiro. Utilize o *help* do Matlab para obter informação sobre estas funções.
b) Com alguns exemplos, verifique quais são os modos de arredondamento implementados naquelas funções.
11. O modo de arredondamento implementado no Matlab é, por defeito, o arredondamento “para o mais próximo”. Podemos no entanto, alterar o modo de arredondamento executando uma das seguintes instruções

```
>> system_dependent('setround',-Inf)
>> system_dependent('setround', Inf)
>> system_dependent('setround', 0)
>> system_dependent('setround', 0.5)
```

- a) No Matlab use a função **rand** para gerar um vector x de 100 números aleatórios no intervalo $[-0.5, 0.5]$ e calcule os quatro valores das somas obtidas, usando os 4 modos de arredondamento.
- b) Compare os diferentes valores da soma e indique um intervalo que contém o valor exacto da soma.