Métodos Computacionais em Engenharia (FIS3022)
Problemas 1
DFA–FCUP 2021-22
4 de Março de 2022

1. **Precisão do computador**

   Escreva um código para calcular a precisão de trabalho do seu computador. (*Sugestão:* enquanto um número for diferente da sua metade, continue a dividi-lo ao meio.)

   (a) escreva um programa para determinar o menor número representável em precisão simples.

   (b) repita para precisão dupla.

2. **Erros de arredondamento**

   O método de Liu Hui para determinar π corresponde a considerar uma série de polígonos cada um com o dobro dos lados do anterior. A expressão é:

$$\pi \approx \frac{M \times N_{\text{lados}}}{2}$$

   onde M é o lado do polígono e N_{lados} o número de lados. Seguindo a figura o resultado é:

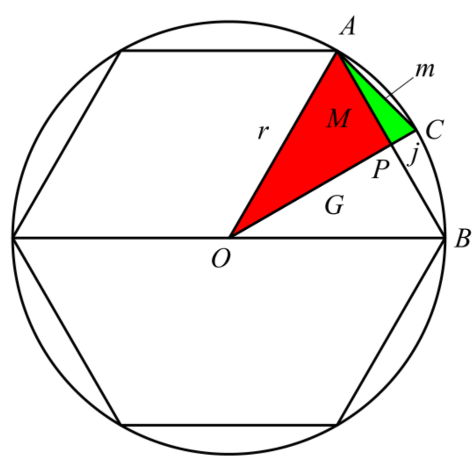

Figura 1: Geometria para o cálculo do lado de um polígono de $2N$ lados a partir do lado de um polígono com N lados. Fonte: https://en.wikipedia.org/wiki/Liu_Hui%27s_%CF%80_algorithm

$$m = \sqrt{\left(\frac{M}{2}\right)^2 + \left(r - \sqrt{r^2 - \left(\frac{M}{2}\right)^2}\right)^2}$$

   Tomando um hexágono inscrito numa circunferência de raio $r = 1$, é fácil mostrar que o lado $M_6 = 1$, que pela expressão inicial dá $\pi = 3$. Seguindo o algoritmo, $M_{12} = \sqrt{2 - \sqrt{3}} \approx 0.5176$, que conduz a $\pi = 3,1058$.

(a) escreva um programa que calcule π com 10 casas decimais, usando o algoritmo de Lui Liu. Imprima a diferença para o verdadeiro valor e veja como evolui à medida que aumenta os lados do polígono.

(b) é possível que tenha tido problemas na alínea anterior! A que se deve?

3. **Erros de arredondamento 2**

   Outra método clássico para determinar π é a fórmula de Liebnitz. Esta parte do valor trigonométrico:

$$\tan\left(\frac{\pi}{4}\right) = 1 \longrightarrow \arctan(1) = \frac{\pi}{4}$$

   Da expansão em série de Taylor para arctan vem:

$$\arctan(x) = x - \frac{x^3}{3} + \frac{x^5}{5} - \frac{x^7}{7} + \frac{x^9}{9} - \ldots,$$

   que aplicado ao argumento $x = 1$:

$$1 - \frac{1}{3} + \frac{1}{5} - \frac{1}{7} + \frac{1}{9} - \ldots = \sum_{n=0}^{\infty} \frac{(-1)^n}{2n+1} = \frac{\pi}{4}.$$

   (a) escreva um programa que faça esta soma e calcule π.

   (b) obtenha progressivamente mais casas decimais no resultado (de novo, imprima a diferença para o verdadeiro valor e veja como evolui).

   (c) repita, mas agora usando $\arctan(\pi/6)$. Compare com o caso anterior.

4. **Erros de arredondamento 3**

   A função zeta de Rienman, $\zeta(s)$, pode ser definida por:

$$\zeta(s) = \sum_{n=1}^{\infty} n^{-s} = \frac{1}{1^s} + \frac{1}{2^s} + \frac{1}{3^s} + \ldots \qquad \text{se } \sigma \equiv \mathrm{Re}(s) > 1.$$

   Verifica-se que $\zeta(2) = \frac{\pi^2}{6}$.

   (a) Escreva um programa que lhe permita calcular π. Qual a precisão que alcança usando 1000, 1000 e 100000 termos?

5. **Erros em algoritmos**

   Os erros de um algoritmo têm duas contribuições: aproximações (truncagens) e arredondamentos. Para N passos podemos escrever para o erro total:

$$\epsilon_{\text{tot}} = \epsilon_{\text{aprox}} + \epsilon_{\text{arred}} \simeq \frac{\alpha}{N^\beta} + \sqrt{N}\epsilon_m$$

   Suponhamos que a resposta exacta ao nosso problema é $\mathcal{A}$, e que a calculada ao fim de N passos (ou com N termos) é $A(N)$. Então procuramos estudar o comportamento da solução calculada para valores de N em que os erros de arredondamento ainda não sejam importantes, de modo que :

$$A(N) \simeq \mathcal{A} + \frac{\alpha}{N^\beta}.$$

De seguida repetimos o cálculo com $2N$ passos. Se o erro de arredondamento continuar a não ser importante, subtraindo as duas estimativas eliminamos $\mathcal{A}$:

$$A(2N) - A(N) \simeq \frac{\alpha}{N^\beta}\left(\frac{1}{2^\beta} - 1\right) \simeq \frac{\alpha}{N^\beta}.$$

Para ver se as hipótese anteriores estão correctas e determinar o nível de precisão para a melhor escolha de N, fazemos o gráfico de $\log_{10}|(A(N) - A(2N)/A(2N)|$ vs $\log_{10} N$. Uma rápida queda linear indica que estamos na região de convergência, e o declive dá-nos β. À medida que N cresce, o gráfico muda devido ao erro de arredondamento. Podemos assim determinar o N "ideal".

Podemos definir a função exponencial negativa como:

$$e^{-x} = 1 - x + \frac{x^2}{2!} - \frac{x^3}{3!} + \ldots = \sum_{n=0}^{\infty} \frac{(-x)^n}{n!}$$

que converge desde que $x^2 < \infty$. Podemos então calcular uma aproximação via:

$$e^{-x} \simeq \sum_{n=0}^{N} \frac{(-x)^n}{n!}. \tag{1}$$

Para ser um bom algoritmo é preciso que o primeiro termo desprezado, $(-x)^{N+1}/((N+1)!)$, e a soma de todos os termos desprezados, sejam pequenos comparados com a soma obtida.

(a) Escreva um programa que calcule a soma referida (1).

(b) Calcule a série para $x < 1$, escolhendo um N de modo a que o termo seguinte seja menos que 10^{-7} da soma até N.

(c) Examine os termos da série para $x \sim 10$ e note os cancelamentos subtractivos apreciáveis que ocorrem quando termos grandes consecutivos quase cancelam. Em particular note o que acontece perto de $n \simeq x - 1$.

(d) Veja se obtém melhores resultados usando $\exp(-x) = 1/\exp(x)$ para valores mais elevados de x. Isto elimina os cancelamentos subtractivos mas não todo os erros de arredondamento.

(e) Aumentando progressivamente x de 1 até 10, e depois de 10 até 100, determine experimentalmente quando a série perde precisão, e depois deixa de convergir.

(f) Faça gráficos do erro versus N para vários valores de x.

(*Fonte:*Computational Physics, R. Landau, M. Páez e C. Bordeianu, Wiley (2007))