

7600017 – Introdução à Física Computacional

Primeiro Projeto (prazo até 20/03/18)

Instruções

- Crie um diretório “PROJ1_#usp” em /home/public/FISCOMP18/PROJ1
- Proteja seu diretório para não ser lido por “g” e “o”
- Deixe no diretório apenas 5 arquivos, de nomes “exer1.f”, ..., “exer5.f”
- Os códigos devem seguir **rigorosamente** os padrões especificados abaixo para entrada/saída
- Note: se deixar de fazer algum exercício não inclua o arquivo correspondente

Exercícios

1. Vamos determinar a **precisão** de seu computador usando aritmética de ponto flutuante em precisão simples e dupla. Para isso, crie um “loop” que soma uma variável a ao número 1 e testa se, à medida que a diminui, o resultado da soma difere do número 1. O menor valor que pode ser detectado para essa variável é a chamada *precisão de máquina*. Inicie seu programa com $a = 1$ e vá dividindo o valor por 2 a cada iteração. Faça isso para variável de tipo real com precisão simples e dupla. Trabalharemos de agora em diante sempre com **precisão dupla**. Dessa forma, forneça com precisão dupla a área de um círculo de raio r . Você deve se certificar de que o valor usado para a constante π possui a precisão desejada. Após realizar testes com seu programa, elabore a versão que deverá ser despositada para correção, respeitando **rigorosamente** as especificações a seguir.

Leia, a partir do terminal, o raio r a ser considerado. Leia *apenas* essa entrada. A saída do programa deve ser dada no formato **exato** a seguir: uma linha explicativa de sua escolha, e.g. “PRECISÃO SIMPLES”, seguida de linhas com o valor de a e da soma $1 + a$, na mesma linha, após cada iteração do loop (sendo que a primeira corresponde a $a = 0.5$ e a última corresponde a $1 + a = 1$, ou seja, o *penúltimo* valor de a foi o menor detectado). Faça isto para precisão simples e, logo em seguida, para precisão dupla. (Você pode pular linhas entre os resultados, se preferir.) Logo a seguir, imprima seu resultado para a área do círculo de raio r (em precisão dupla!!) em uma linha, sendo que o resultado numérico deve ser a última palavra da linha. Seu programa deverá ser escrito em **fortran** (para compilação com **gfortran**) e deve se chamar **exer1.f**, necessariamente.

2. Como será que o computador calcula eficientemente funções, como a exponencial ou o seno? Considere a série de Taylor para $\sin(x)$ ao redor de $x = 0$. Supondo que o erro cometido pelo truncamento da série corresponda ao primeiro termo desprezado, de ordem x^N , quanto deve ser N para que $\sin(2\pi)$ seja dado em precisão dupla (que tomaremos como 10^{-15})? (para pensar antes de realizar o exercício abaixo)

Como você mudaria o programa acima para calcular o *cosse*no do ângulo x ? Implemente o cálculo da série de Taylor para o cosse no no programa **exer2.f**, que deve

ler *apenas* o ângulo x (em radianos) a partir do terminal e fornecer, como saída, três linhas, contendo: i) o valor de $\cos(x)$ com precisão de 10^{-15} , ii) o valor de N correspondente ao primeiro termo desprezado da série e iii) $\cos(x)$ calculado usando a função em precisão dupla [se tiver curiosidade, compare os resultados de $\cos(x)$ e $\text{dcos}(x)$].

- Qual o maior número inteiro (de tipo “integer”) que pode ser representado no computador? escreva um programa para verificar o valor desse número, e.g. adicionando sucessivamente 1 a um número inteiro dado e testando a consistência do resultado. (Você verá que mesmo um computador rápido pode ter dificuldade em realizar essa tarefa!) Agora, utilizando o valor encontrado, que chamaremos de N_{\max} , determine o maior número cujo fatorial pode ser representado como variável de tipo integer, que chamaremos de n_{\max} . É um número pequeno, certo? Geralmente evitamos calcular o fatorial de números maiores do que n_{\max} , ou seu logaritmo (que é uma grandeza útil em várias situações físicas), utilizando a aproximação de Stirling

$$\ln n! \approx n \ln n - n + \frac{1}{2} \ln(2\pi n).$$

Escreva um programa que forneça (no terminal) a seguinte “tabela”: para valores de n de 1 até n_{\max} (determinado a partir de N_{\max}), um por linha, forneça quatro colunas com os valores de n , $n!$, $\ln n!$ e a correspondente aproximação de Stirling, definida acima. Mais precisamente, seu programa deve **ler** um número N_{\max} (que pode ser o valor calculado acima, ou outro número menor) e dar como resultado/saída: o número n_{\max} correspondente (como última palavra da primeira linha da saída) e, a partir da linha seguinte, a tabela descrita acima. Note: na versão a ser entregue **não** inclua o cálculo de N_{\max} .

- Escreva um programa para ordenar números. Leia os números **inteiros** N e M ($M \leq N$) um por linha a partir do terminal. Leia N números **reais** (um por linha) a partir do arquivo “ord_in.dat” e imprima, em ordem crescente, os M maiores números (um por linha) no arquivo “ord_out.dat”. Você pode supor que $N \leq 1000$ e que não haja números repetidos na lista dada.
- Uma matriz não é um número, como sabemos. Um vetor multiplicado por uma matriz resulta em um novo vetor, apontando em uma direção diferente da anterior. No entanto, para alguns vetores especiais, a multiplicação por uma matriz terá o mesmo efeito que multiplicá-los por um simples número, não alterando sua direção. Para uma dada matriz M , tais vetores especiais são chamados de seus **autovetores** \vec{u} , e os números correspondentes são os **autovalores**: para cada autovetor \vec{u} temos um autovalor λ associado à sua multiplicação por M

$$M \vec{u} = \lambda \vec{u}.$$

Os autovalores e autovetores de uma matriz possuem propriedades importantes que auxiliam no estudo de diversos problemas físicos (em particular, são essenciais para o estudo da mecânica quântica!) e portanto sua determinação é de grande importância. Ao mesmo tempo, trata-se geralmente de uma tarefa computacionalmente intensa, para a qual foram desenvolvidos vários métodos visando máxima eficiência.

Vamos utilizar um “truque”, o chamado método das potências, para determinar o mais alto valor de λ (com maior módulo) para uma dada matriz M . Uma das propriedades

dos autovetores de matrizes fisicamente importantes é que, dada a matriz M , qualquer vetor \vec{x} do espaço estudado pode ser escrito como uma combinação linear dos autovetores \vec{u}_i (correspondendo aos autovalores λ_i) de M . Ao multiplicarmos um grande número de vezes k a matriz M (de dimensão $n \times n$) por um vetor \vec{x} arbitrário teremos que apenas a componente de maior autovalor (que vamos supor que seja a de índice 1) “sobreviverá”

$$M^k \vec{x} = M^k (c_1 \vec{u}_1 + c_2 \vec{u}_2 \dots + c_n \vec{u}_n) = (c_1 \lambda_1^k \vec{u}_1 + \dots),$$

já que, quanto maior o valor de k , maior será a importância relativa do primeiro termo da soma acima. Dessa forma, podemos determinar λ_1 e \vec{u}_1 por simples multiplicações de matriz por vetor. Mais exatamente, a cada iteração do método, uma estimativa cada vez mais precisa para λ_1 é obtida de

$$\lambda_1 \approx \frac{\vec{x}_k \cdot M \vec{x}_k}{\vec{x}_k \cdot \vec{x}_k},$$

onde definimos $\vec{x}_k \equiv M^k \vec{x}$. (Note que o valor de λ_1 acima corresponde ao passo $k + 1$ do método.) Diremos que o valor encontrado para λ_1 no passo k tem precisão ϵ se a diferença entre as estimativas dos passos $k - 1$ e k não for maior do que ϵ . Elabore um programa para determinar estimativas para λ_1 e \vec{u}_1 correspondendo a uma dada precisão ϵ para λ_1 . Leia, a partir do terminal: a precisão ϵ (na primeira linha de entrada), a dimensão da matriz M (na segunda linha de entrada) e seus elementos, linha por linha (com n colunas) nas linhas seguintes. Imprima como saída: o valor de λ_1 , como última palavra da primeira linha, seguido das posições do autovetor correspondente, uma por linha. Suponha $n \leq 20$ e que M seja real e simétrica.