7600017 - Introdução à Física Computacional - 2022

Prof. Guilherme Sipahi

Primeiro Projeto

19/09/2022

Instruções

- Crie um diretório **proj1_#*usp*** em `/public/IntroFisComp22/projeto1`
- Proteja seu diretório para não ser lido por `g` e `o`
- Deixe no diretório apenas 5 arquivos de nomes `exer01.f90`, ..., `exer05.f90`
- Os códigos devem seguir rigorosamente os padrões especificados abaixo para `entrada/saída`
- **Se deixar de fazer algum exercício não inclua o arquivo correspondente**

Exercícios

1. **Precisão de números reais**

   Como a representação de números em computadores está sujeita a muitas possíveis escolhas, a representação de números reais gerou um padrão internacional conhecido como IEEE 754, revisto em 2008, que prevê a representação destes números com três diferentes precisões: simples e dupla e quádrupla. Este padrão estabelece um formato para a representação com números bem estabelecidos de bits para mantissas e expoentes de cada precisão, determinando um número bem definido de casas decimais e o intervalo de extensão dos valores.

   Neste exercício, você vai avaliar a precisão da mantissa usando uma variável auxiliar, a, que deve ser inicializada com o valor 1. O algoritmo deve implementar um *loop* para a comparação do valor da soma $1 + a$, com o número 1. Ao final de cada passo do *loop* a, deve ser dividida por 2. A precisão da mantissa será dada pelo valor de a na última iteração em que os valores de 1 e da soma divirjam e o *loop* deve ser finalizado uma vez que os valores deixem de divergir. O número de bits da representação é dado pelo número de passos executados até a obtenção da precisão.

   O programa `exer01.f90` deve implementar este algoritmo para a determinação das precisões das três representações, respeitando as seguintes diretrizes:

- **Iniciando com a precisão simples, escreva na tela uma linha que explicite qual formato está sendo avaliado (por exemplo, "PRECISAO SIMPLES"). Nas linhas consecutivas imprima os valores de a, e $1+a$ para cada iteração. Repita o processo para as precisões dupla e quádrupla, nesta ordem.**
- **Após a finalização dos três *loops*, imprima em três linhas consecutivas (uma para cada precisão), o número de bits e o valor da precisão encontrada para precisões simples, dupla e quádrupla, nesta ordem.**

Dica: Os testes das 3 precisões precisam de números definidos com o formato adequado: use `real(4)` para a precisão simples, `real(8)` para a precisão dupla e `real(16)` para a precisão quádrupla. Três *aa* diferentes devem ser inicializados. Para inicializar os valores no gfortran, use o formato `X.YeO` para precisão simples, `X.YdO` para precisão dupla e `X.Y_16` para precisão quádrupla.

2. **Erro númerico em séries - Aproximações de funções trigonométricas**

   Quase sempre, quando precisamos fazer a aproximação numérica de uma função, começamos por usar uma série de Taylor. Será que esta é uma boa ideia para a determinação das funções trigonométricas?

   Vamos usar a série de Taylor para o seno

   $$\sin(x) = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \frac{x^9}{9!} - \cdots$$

   e determinar a precisão da aproximação nos formatos de precisão simples e dupla. A precisão da aproximação será ϵ/v_p, onde ϵ é o menor termo da série que modifica o valor do resultado e v_p, o valor principal, é resultado da série truncada neste termo.

   Implemente um programa `exer02.f90` que determine a precisão para $x = 0.1, 0.2, 0.3$ e $0,4$. A saída numérica deve apresentar uma tabela com um valor de x e sua precisão em cada linha, na ordem dada anteriormente. Depois da tabela, responda a questão: você acha que as funções trigonométricas devem ser aproximadas por séries? Justifique a sua reposta.

   Você pode usar tantas linhas adicionais quanto quiser em sua resposta.

3. **Números primos**

   Escreva o programa `exer03.f90` para determinar os números primos entre 1 e M.

   Leia M a partir do terminal e escreva os resultados (um por linha) no arquivo `primos_out.dat`. Teste seu programa para $M = 100,\ 1000,\ 10000$.

   Opcional: tente otimizar seu programa para torná-lo mais rápido (você pode verificar a velocidade de processamento do programa utilizando o comando time do linux).

4. **Área e volume do tetraedro**

Escreva o programa `exer04.f90` executar as tarefas previstas no exercício.

Leia, a partir de um arquivo de entrada de nome `vet_in.dat`, quatro vetores com coordenadas as dos vértices de um tetraedro $\vec{v}_1$, $\vec{v}_2$, $\vec{v}_3$, $\vec{v}_4$ (com coordenadas x_1, y_1, z_1, x_2, y_2, z_2, etc.). Os quatro vetores devem ser lidos separadamente, com as três coordenadas de cada um em uma linha separados por espaços em branco, i.e.

```
x1  y1  z1
x2  y2  z2
 ⋮
```

(a) Determine o volume e a àrea de cada uma das faces do tetraedro.

(b) Determine quantos valores distintos de área existem no tetraedro.

(c) Coloque as áreas em ordem crescente.

Escreva o volume em uma linha, a soma das áreas da face na segunda e, em ordem crescente de valor, cada uma das áreas distintas em uma nova linha de um arquivo com o nome `tetra_out.dat`.

5. **Determinação de autovalores**

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"

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

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 o programa `exer05.f90` 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.