

Projecto de Física Computacional (R. Coelho)

Mestrado Integrado em Engenharia Física Tecnológica
Ano lectivo: 2019/20 Semestre: 1

Data de Entrega: **4 de Janeiro de 2020 (23h59)**

Instruções 1

- **Certifique-se** que tem uma pasta de trabalho **PROJECTO** no directório **do grupo a que pertence** no servidor SVN com a seguinte estrutura :

```
PROJECTO
├── bin
├── main
│   └── Projecto.cpp
├── obj
│   └── Makefile
├── src
│   ├── PIC.cpp
│   └── PIC.h
```

- **Não** altere o nome dos ficheiros com código fonte ou ficheiros que contêm as figuras a serem produzidas no Projecto.
- **Todos** os ficheiros com código fonte (*.cpp), header files (*.h) e Makefile necessários em cada problema devem ser submetidos no servidor SVN.
- **NAO** CRIE QUALQUER FICHEIRO COM **ACENTUAÇÃO** NEM FICHEIROS COM NOME QUE DIFIRA APENAS EM CARACTERES LOWERCASE/UPPERCASE e.g. Teste.cpp e test.cpp.
- **Não submeta no SVN ficheiros binários (*.o ou .exe).**

Instruções 2

- A pasta **/src** deve conter **todas as classes** (*header e source*) necessárias à resolução do Projecto. Inclui quer classes **já desenvolvidas** ao longo do semestre quer classes **novas** pedidas no Projecto.
- A pasta **/bin** serve para conter o ficheiro binário (*.exe) gerados ao executar a Makefile. *Não submeta este ficheiro binário (ou object code, a ser criado em /obj) no SVN !*
- A pasta **/main** serve para conter apenas o ficheiro Projecto.cpp (programa main para a execução do Projecto).
- Na **raíz** da pasta PROJECTO devem estar
 - A(s) Figura(s) pedidas para o Projecto.
 - Ficheiro README (apenas caso necessário) que auxilie a compreensão do trabalho feito (obstáculos encontrados que não permitam concluir o trabalho, bibliotecas desenvolvidas e como compilar/linkar).
 - **Relatório do trabalho desenvolvido** (máximo 5 páginas, ficheiro **PDF**), incluindo, entre outros, descrição dos métodos implementados para a resolução do problema, implementação das condições iniciais do problema e resultados obtidos na dinâmica temporal conforme pedido nos objectivos do trabalho.

PROBLEMA

1 – Introdução

Neste problema vamos estudar a dinâmica de um ensemble de partículas com carga eléctrica sujeitas a um campo eléctrico que resulta, de forma auto-consistente, da posição ocupada pelas próprias partículas carregadas (!). Na figura 1 este conceito é ilustrado com um exemplo.

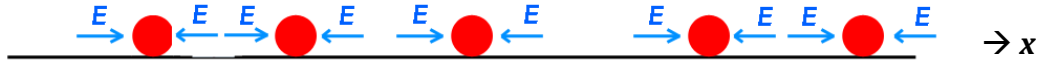


Figura 1 – Distribuição de electrões com indicação do campo eléctrico criado por cada carga.

O campo eléctrico em qualquer ponto do espaço resulta, naturalmente, da contribuição conjunta de todas as cargas eléctricas. Na geometria a uma dimensão indicada na Figura 1, resulta também que, se soubermos o campo eléctrico $\mathbf{E} = E_x(x)\mathbf{x}$ em qualquer valor de x , podemos também calcular o potencial eléctrico $\phi(x)$ através da relação $E(x) = E_x(x) = -d\phi/dx$.

A equação do movimento para uma partícula carregada é naturalmente dada por

$$\frac{d^2x}{dt^2} = -\frac{qE(x)}{m}$$

onde x é a posição actual da partícula, q a sua carga eléctrica, m a sua massa. Logo, para podermos avançar no tempo a posição da partícula, precisamos de saber o campo eléctrico nessa posição, que por sua vez depende da posição de **todas** as partículas.

Apesar de apelativo, o cálculo do campo eléctrico e potencial eléctrico acima indicado não é, do ponto de vista computacional, o mais simples/natural. Para agilizar o cálculo convém utilizar uma das equações de Maxwell (**Lei de Gauss**), que relaciona o campo eléctrico com a densidade de carga eléctrica i.e.

$$\nabla \cdot \vec{E} = \frac{\rho}{\epsilon_0} = \frac{e}{\epsilon_0} (n_{ions} - n_{electrons}) \quad (1)$$

onde ρ é a densidade de carga total (positivas e negativas de carga unitária i.e. protões e electrões), n é a densidade de partículas e ϵ_0 a permissividade do vácuo.

Para simplificar o problema, vamos assumir que os iões (protões), sendo muito mais pesados que os electrões, estão *estacionários* e têm densidade constante n_0 sendo que, em média, a densidade de electrões e iões é igual (**quasi-neutralidade**).

Para determinarmos a densidade de electrões, é habitual discretizar o nosso domínio espacial $\mathbf{x} = [x_{min}, x_{max}]$ numa grelha uniforme $x = x_{min} + j \cdot h$ onde $j = 0, 1, \dots, (x_{max} - x_{min})/h$ onde h (constante) é o espaçamento da grelha i.e. $h = x_{j+1} - x_j$.

Para “povoar” a densidade electrónica nesta grelha, tomemos o exemplo de um electrão com posição e velocidade (r_i, v_i) , com $i = 0..N-1$ relativo a uma população de N -partículas. Quando a posição do electrão (r_i) é tal que $x_j < r_i < x_{j+1}$, adopta-se a seguinte estratégia:

$$n_j \rightarrow n_j + \frac{(x_{j+1} - r_i)}{(x_{j+1} - x_j)} \Big/ h \quad n_{j+1} \rightarrow n_{j+1} + \frac{(r_i - x_j)}{(x_{j+1} - x_j)} \Big/ h \quad (2)$$

Facilmente resulta que, **por partícula**, teremos um incremento na densidade “total” de $1/h$. Caso a partícula esteja exactamente entre 2 pontos da grelha, o aumento de densidade é igual em cada um desses pontos da grelha.

Facilmente se conclui também que podemos aumentar artificialmente a densidade através de uma factor de escala (constante em todos os pontos da grelha) até obter o valor desejado. No nosso problema,

porém, não será necessário pois iremos usar uma versão “normalizada” das nossas equações de movimento e Lei de Gauss.

2 – Formalismo teórico e modelo a implementar

2.1 Formalismo teórico

Tal como referido na secção anterior, iremos utilizar uma versão adimensional das equações que regem o movimento e distribuição de cargas/campo eléctrico. De facto, iremos utilizar a seguinte normalização:

$$t \rightarrow t \cdot \sqrt{\frac{\epsilon_0 m_e}{n_0 e^2}} \quad \left(\text{onde } \frac{n_0 e^2}{\epsilon_0 m_e} \text{ é o quadrado da chamada } \textit{frequência de plasma} \right)$$

$$x \rightarrow x \cdot \lambda_D \quad \left(\text{onde } \lambda_D \text{ é o chamado } \textit{comprimento de Debye} \right)$$

$$E \rightarrow E \cdot \frac{en_0 \lambda_D}{\epsilon_0}$$

$$n \rightarrow n \cdot n_0$$

Com esta normalização, as equações que regem o nosso sistema de electrões é:

$$\frac{d^2 r_i}{dt^2} = -E(r_i) \quad (3)$$

$$E(x) = -\frac{d\phi(x)}{dx} \quad (4)$$

$$\frac{d^2 \phi(x)}{dx^2} = \frac{n(x)}{n_0} - 1 \quad (5)$$

onde r_i é a posição do electrão- i .

2.2 Algoritmo de simulação

Para evoluir este sistema dinâmico, devemos então ter em conta os seguintes passos:

1. A partir de uma função de densidade de probabilidade $F(r,v)$, obter uma população aleatória de N electrões com posição e velocidade (r_i, v_i) , com $i=0..N-1$.
2. Usando a Eq. (2) (valores x_j de grelha espacial já normalizados), obter a densidade de electrões na grelha x_j com $j=0..(x_{\max}-x_{\min})/h$, sendo x_{\min} e x_{\max} dados em unidade de λ_D . Normalizar esta densidade apropriadamente de acordo com o lado direito da Eq. (5) para o potencial (**cuidado com a constante de normalização n_0** ...deve ser calculada previamente e tem única e exclusivamente que ver com o número total de partículas e a dimensão da grelha espacial i.e. a densidade média de partículas por unidade de comprimento...).
3. Depois de obtida a densidade total $(n(x)/n_0 - 1)$, resolver a equação (5).
4. Determinado o potencial em todos os pontos da grelha espacial, calcular o campo eléctrico nas posições de todas os N electrões (ver Eq.(3)).
5. Avançar no tempo a posição dos electrões de acordo com a Eq. (3).
6. Repetir o passo (2) em diante tantas vezes quantas necessárias de acordo com o tempo total de simulação da dinâmica do ensemble de electrões.

2.3 Condições iniciais, fronteira e modelos numéricos

A simulação a desenvolver terá que ter em atenção condições fronteira específicas para o problema e modelos numéricos para as operações a efectuar. Em particular:

2.3.1. Condição inicial para a posição e velocidade das partículas

- A posição inicial das N partículas é aleatória segundo uma distribuição de probabilidade uniforme em $x \in [x_{\min}, x_{\max}]$. Já a velocidade deve obedecer a uma função de densidade de probabilidade dada por

$$F(v) \propto e^{-\frac{(v-v_b)^2}{2}} + e^{-\frac{(v+v_b)^2}{2}} \quad (6)$$

sendo v_b dada.

2.3.2. Condição fronteira para a posição e velocidade das partículas

- A simulação da dinâmica ocorre numa “caixa” a uma dimensão (eixo dos x), de limites x_{\min} e x_{\max} . ($L = x_{\max} - x_{\min}$)

- Assumem-se condições fronteira periódicas para a posição da partícula e.g. se $r_i > x_{\max}$ então $r_i \rightarrow r_i - L$. Quanto à velocidade, não sofre alteração nesta mudança de posição.

2.3.3. Condição fronteira para o potencial

- O potencial eléctrico também obedece a condição fronteira periódica i.e. $\phi(x_{\min}) = \phi(x_{\max})$. Por outro lado, sabendo que o campo eléctrico é o gradiente do potencial, o potencial está sempre definido a menos de uma constante, logo podemos assumir que $\phi(x_{\min}) = 0$.

2.3.4. Métodos numéricos a utilizar

- Para o cálculo da posição inicial das partículas **apenas** poderá usar distribuições de probabilidade *Uniform(0,1)*. No final deverão ter o número de partículas pedido seguindo uma certa estatística de velocidade e de posição espacial.

- Para o cálculo da densidade deve usar o método especificado na Eq. (2).

- Para o cálculo do potencial deverá resolver a Eq. (5) com a classe EqSolver após discretização da equação diferencial usando o **método de diferenças finitas para BVPs**.

- Para o cálculo **do campo eléctrico** (a partir do potencial...) em qualquer valor de $x \in [x_{\min}, x_{\max}]$, deverá utilizar a classe **Spline3Interpolator** com a adição de um método que calcule a derivada da função interpoladora num ponto arbitrário x. Não é necessário nem aconselhável usar a classe *Derivator* (!).

- Para evoluir no tempo a equação diferencial ordinária Eq. (3) deverá utilizar, com a classe **ODESolver**, o **método de Runge-Kutta 4** para ODEs de 1ª ordem (como sabe uma ODE de 2ª ordem pode ser escrita como um sistema de 2 ODEs...).

3 – Objectivos do trabalho

1º

Implementar uma classe **PIC** de modo a podermos evoluir no tempo a posição e velocidade de um ensemble de partículas de dimensão $N=100,000$. Para tal, deverá utilizar o programa *main* disponibilizado e que, conforme observa, invoca os seguintes métodos:

- Construtor da classe PIC. A declaração deste construtor obedece à seguinte sintaxe

```
PIC(vector<double> velocity, int Npart=1000, double xmin=0.0, double xmax=1.0, int ngrid=100)
```

com

vector<double> velocity: vector com a velocidade dos 2 beams (positivo e negativo)

int Npart=1000: número total de partículas

double xmin=0.0: início do domínio espacial que contém as partículas

double xmax=1.0: fim do domínio espacial que contém as partículas

int ngrid=100: numero de pontos da grelha espacial onde calcular a densidade e potencial.

- Gerador da distribuição inicial das partículas, com seguinte declaração:

```
void FdistV(vector<double> veloc, bool save_plot, bool show_plot=false)
```

com opção de criação e gravação das **figuras** com função de densidade de probabilidade $F(v)$ “analítica” e com o **histograma** equivalente e com opção de visualização das respectivas figuras.

- Criação e visualização do **espaço de fases** de todas as partículas (o **espaço de fases representa-se com pontos num sistema de coordenadas posição-velocidade, um por cada partícula**), com seguinte declaração:

```
void Plot_Phase_Space(bool save_plot, bool show_plot=false)
```

com as mesmas opções de gravação/visualização anteriormente indicados.

- Cálculo do perfil de densidade na grelha espacial, com seguinte declaração:

```
void Density(bool save_plot, bool show_plot=false)
```

com as mesmas opções de gravação/visualização anteriormente indicados.

- Cálculo do potencial na grelha espacial, com seguinte declaração:

```
void Poisson(bool save_plot, bool show_plot=false)
```

com as mesmas opções de gravação/visualização anteriormente indicados.

- Avanço no tempo de todas as partículas, com seguinte declaração:

```
void TimeStep(double dt)
```

Os data members da classe, privados contém, entre outros, o vector de velocidades (2) a usar na PDF $F(v)$, o número total de partículas, coordenadas limites (x_{\min}, x_{\max}) que definem o domínio de simulação, número de pontos da grelha espacial, ponteiros com coordenadas da grelha espacial, densidade electrónica e potencial eléctrico. Naturalmente, deverão ter também objectos do tipo TFormula, ODEPoint e ODEsolver.

Caso I - Simular a dinâmica de 100,000 partículas com os seguintes parâmetros:

- $V_b = 5$;
- $X_{\min}=0$, $X_{\max}=50$
- Resolução espacial $dx=0.05$
- Resolução temporal $dt=0.04$
- Tempo de simulação: **60** (unidades normalizadas)

→ Obtenha figuras ilustrativas com o *espaço de fases*, potencial e densidade ($n(x)/n_0 - 1$) para diversos instantes da simulação. Outros meios adicionais de visualização da dinâmica serão valorizados.

→ O que pode concluir ou como pode caracterizar a dinâmica obtida e.g. comprimento de onda das oscilações, formação de vórtices no espaço de fases (evidência de um *regime instável*),...

Caso II - Simular a dinâmica de 100,000 partículas variando a dimensão máxima do domínio espacial (x_{\max}) de modo a obter, qualitativamente, uma dinâmica mais “estável” (ausência de vórtices no espaço de fases).

→ Determine aproximadamente para que valor de x_{\max} é que ocorre uma transição de regime (estável para instável).