

# Física Computacional - Prática 9

## Questão 2

Alex Enrique Crispim

O programa `4-poisson.py` busca resolver a Equação de Poisson

$$\nabla^2 \phi = -\rho/\epsilon_0,$$

utilizando-se dos mesmos métodos apresentados no programa `laplace.py`. A diferença está na forma da expressão final para o cálculo de  $\phi'$ . Em duas dimensões, seguindo as ideias apresentadas na resolução da questão 1, temos

$$\frac{1}{a^2} [\phi(x-a, y) + \phi(x+a, y) + \phi(x, y-a) + \phi(x, y+a) - 4\phi(x, y)] = -\rho(x, y)/\epsilon_0, \quad (1)$$

de modo que

$$\phi'(x, y) = \frac{1}{4} [\phi(x-a, y) + \phi(x+a, y) + \phi(x, y-a) + \phi(x, y+a)] + \frac{a^2 \rho(x, y)}{4\epsilon_0}. \quad (2)$$

Nas linhas 18 e 19, encontramos as expressões:

```
1 rho[ int(pX - len / 2) : int(pX + len / 2), int(pY - len / 2) : int(pY + len / 2)] =  
  rho0  
2 rho[ int(nX - len / 2) : int(nX + len / 2), int(nY - len / 2) : int(nY + len / 2)] =  
  -rho0  
3
```

onde `pX`, `pY`, `nX` e `nY` representam as coordenadas  $x$  e  $y$  das cargas positivas e negativas, respectivamente. Da forma como está escrito, podemos identificar que o array `rho` representa duas distribuições uniformes de carga na forma de um quadrado; um quadrado de carga negativa e outro de carga positiva. As linhas acima se traduzem como

$$\rho(x, y) = \begin{cases} \text{rho0}, & (x, y) \in [\text{pX} - \text{len}/2; \text{pX} + \text{len}/2] \times [\text{pY} - \text{len}/2; \text{pY} + \text{len}/2] \\ -\text{rho0}, & (x, y) \in [\text{nX} - \text{len}/2; \text{nX} + \text{len}/2] \times [\text{nY} - \text{len}/2; \text{nY} + \text{len}/2] \end{cases}. \quad (3)$$

O procedimento para a solução é o mesmo, trocando-se a equação para  $\phi'$  apenas. O resultado obtido é mostrado na figura ???. O fluxograma para o programa é mostrado na última página.

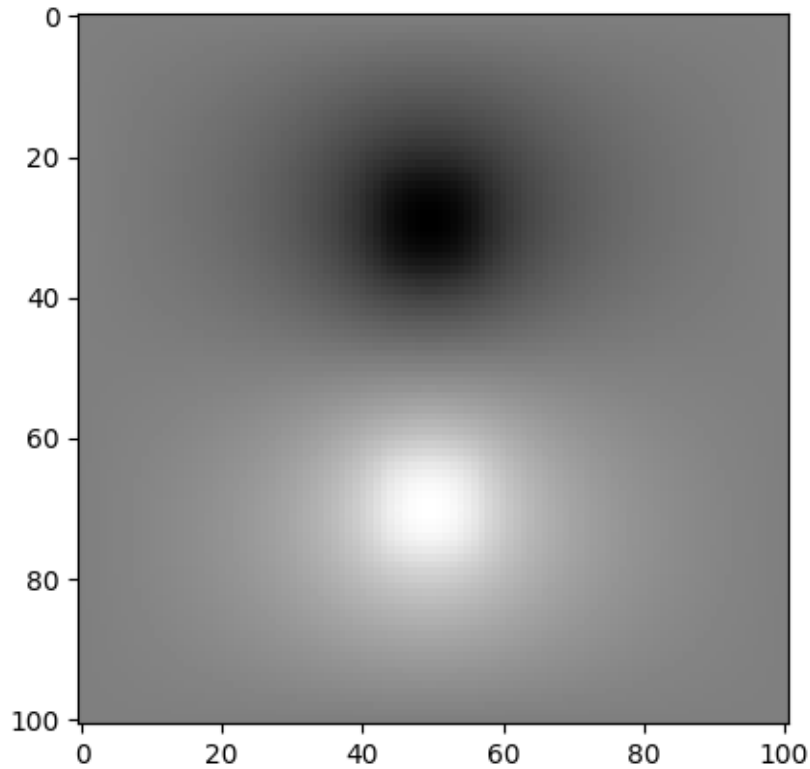


Figura 1: Solução para a equação de Poisson para densidades de carga na forma de retângulos

Figura 2: fig:solucao

Como comentários finais, o tempo de processamento do programa `4-poisson.py` é de 10 minutos e 43.358 segundos. Em contraste, o programa requerido na questão 3, a ser apresentado, fora escrito em C e seu tempo de processamento foi de 9.620 segundos!

Apesar de uma pequena otimização feita (na função `swap` e como é utilizada), a diferença de tempo de processamento é claramente muito grande, mostrando a não eficiência da linguagem Python para processamentos pesados.

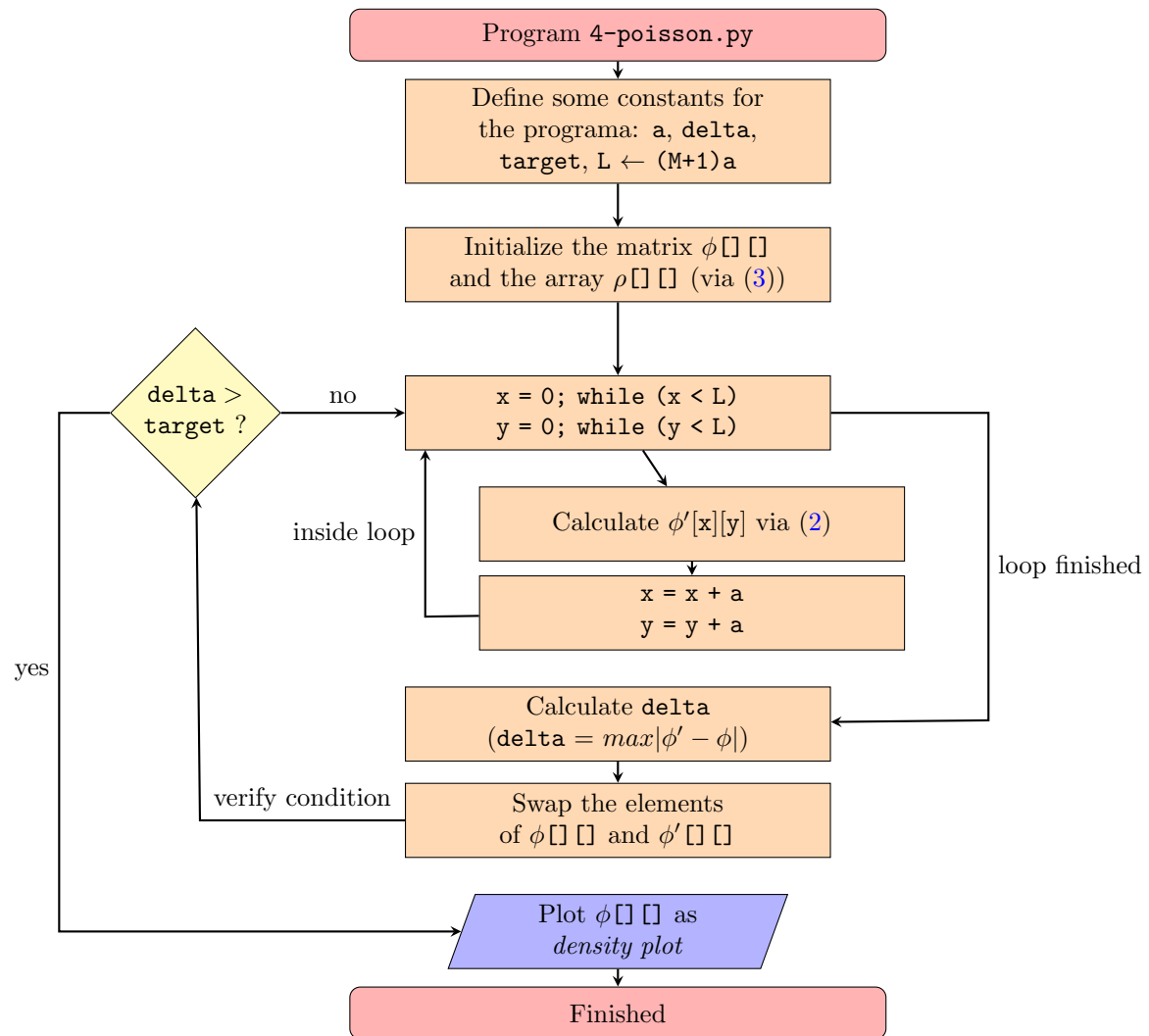


Figure 3: Flowchart of the program 4-poisson.py