Física Computacional

Tarefa 7 - Questão 6

Alex Enrique Crispim

Podemos estender o método de Runge-Kutta descrito nas questões 2 e 3 para mais dimensões fazendo-se a troca dos K_i por arrays $\mathbf{K}_i$, acompanhando a troca de $x' = f$ por

$$\frac{\mathrm{d}\mathbf{x}}{\mathrm{d}t} = \mathbf{f}(t, \mathbf{x}_i).$$

Para o método RK4, mas equações têm basicamente a mesma forma, trocando-se os escalares por vetores;

$$\mathbf{x}_{i+1} = \mathbf{x}_i + \frac{1}{6}[\mathbf{K}_1 + 2\mathbf{K}_2 + 2\mathbf{K}_3 + \mathbf{K}_4], \tag{1}$$

$$\begin{cases} \mathbf{K}_1 = h\mathbf{f}(t_i, \mathbf{x}_i), \\ \mathbf{K}_2 = h\mathbf{f}(t_i + h/2, \mathbf{x}_i + \mathbf{K}_1/2), \\ \mathbf{K}_3 = h\mathbf{f}(t_i + h/2, \mathbf{x}_i + \mathbf{K}_2/2), \\ \mathbf{K}_4 = h\mathbf{f}(t_i + h, \mathbf{x}_i + \mathbf{K}_3). \end{cases}$$

Em certas linguagem (geralmente as que suporta POO), a operação de produto de um array por um escalar e soma de arrays é definida da mesma forma como soma de vetores, de tal forma que podemos escrever as equações acima de forma muito natual. Um exemplo fora feito em *Julia* na pasta *question 6*. Como o trabalho não é tão formal, acredito não ter problema de expressar minha opinião a respeito de um primeiro contato com Julia: eu não gostei da linguagem. Algumas poucas coisas que gostei, tem-se implementadas forma melhor ou, no mínimo, quase equivalente em python. Os testes de performance que se encontram no site da linguagem me são duvidosos, após os primeiros usos da mesma. A sintaxe é incomoda em alguns pontos. Por exemplo, a contagem dos índices dos arrays começa em 1. Uma das vantagens de começar a contar a partir do 0 é o fato de pode se pensar como contando de 1, mas utilizando sempre intervalos abertos à direita e fechados à esquerda. A linguagem Julia tirou isso. Além de que a sintaxe de muitas coisas como a função `append!()` ou `push!()` para arrays, é, no mínimo, incomoda.