universidade de aveiro

Computação Paralela — 2021/2022

29 Junho de 2022

Projeto 2: Paralelização em MPI de métodos iterativos para resolução de equações diferenciais a duas dimensões

Entrega no espaço elearning até dia 15 de Julho de 2022

Considere a equação de Poisson a duas dimensões:

$$\frac{\partial^2 V(x,y)}{\partial x^2} + \frac{\partial^2 V(x,y)}{\partial y^2} = f(x,y)$$

onde $f(x,y) = 7\,\sin(2\pi x)\cos(3\pi x)\sin(2\pi y)\cos(3\pi y)$. Nas aulas escreveu programas paralelizados com MPI que recorrem ao método de Jacobi para encontrar soluções numéricas deste tipo de equação no domínio $-1 \leq x \leq 1$ e $-1 \leq y \leq 1$. Esses programas discretizam o domínio em ambas as direções, x e y, formando uma 'grelha' de pontos onde a função $V(x,y)$ é calculada iterativamente. Para tal, as segundas derivadas são aproximadas por diferenças finitas em cada ponto da grelha usando os valores da função nos seus pontos vizinhos:

$$\frac{\partial^2 V(x,y)}{\partial x^2} \cong \frac{V(x-h,y)-2V(x,y)+V(x+h,y)}{h^2} \text{ e } \frac{\partial^2 V(x,y)}{\partial y^2} \cong \frac{V(x,y-h)-2V(x,y)+V(x,y+h)}{h^2},$$

onde h e o espaçamento uniforme entre pontos vizinhos na grelha (o espaçamento é o mesmo nas duas direções).

As coordenadas dos ponto da grelha são dadas por $x = -1 + jh$ e $y = 1 - ih$ através dos índices $j = 0,1,\ldots n_x - 1$ e $i = 0,1,\ldots n_y - 1$, com $n_x = n_y = 2/h + 1$ neste caso. A equação que resulta da substituição das segundas derivadas pelas suas aproximações, e das coordenadas x e y pelos índices j e i, respectivamente, é expressa na forma iterativa do método de Jacobi por

$$V_{i,j}{}^{(k)} = \frac{1}{4}\left[V_{i-1,j}{}^{(k-1)} + V_{i+1,j}{}^{(k-1)} + V_{i,j-1}{}^{(k-1)} + V_{i,j+1}{}^{(k-1)} - h^2 f_{i,j}\right],$$

onde k representa a iteração. A tolerância ε é definida à partida, e considera-se que o método convergiu quando $\frac{\sqrt{\sum_{i,j}\left[V_{i,j}{}^{(k)} - V_{i,j}{}^{(k-1)}\right]^2}}{\sqrt{\sum_{i,j}\left[V_{i,j}{}^{(k)}\right]^2}} < \varepsilon$.

Os programas desenvolvidos neste projeto devem usar uma decomposição do domínio bidimensional, idêntica à dos programas estudados nas aulas práticas, com os subdomínios distribuídos por duas colunas. Num relatório, descreva em traços gerais os desafios que encontrou em cada alínea, assim como as soluções que usou para os superar, fazendo os comentários que lhe pareçam relevantes. Apresente todos os resultados para uma tolerância $\varepsilon = 10^{-6}$. O programa deve continuar a permitir a escolha de um número arbitrário de pontos, mas para produzir as figuras do relatório use sempre $n_x = n_y = 100$.

a) Adapte o programa escrito nas aulas (aquele que usa uma decomposição cartesiana bidimensional do domínio, em numprocs/2 x 2 subdomínios) de forma a incorporar as seguintes condições fronteira: $V(-1,y) = (1+y)/4$, $V(x,-1) = (1+x)/4$, $V(x,1) = (3+x)/4$ e $V(1,y) = (3+y)/4$. Note que, além das condições fronteira, tem de alterar a função $f(x,y)$. Após os cálculos, o programa deve escrever o resultado num ficheiro binário recorrendo à função MPI_File_write_all(). Use um programa externo (fornecido em Matlab) para importar o resultado e fazer a representação gráfica de V em função de x e y, que deve ser incluída no relatório.

b) Considere agora outro tipo de condições fronteira. Adapte o programa para usar condições fronteira periódicas em ambas as direções, x e y. Note que, com fronteiras periódicas, os valores $V_{i,j}$ nas fronteiras do domínio não são fixados à partida, e têm de ser calculados da mesma forma que os do interior do domínio, recorrendo aos seus 4 vizinhos. Mais concretamente, os pontos de índices $(0,j)$ e (n_y-1,j) são vizinhos, assim como o são os pontos $(i,0)$ e (i,n_x-1), o que implicará comunicações adicionais 'através' das fronteiras. Note que, devido às condições fronteira periódicas, em cada linha e coluna o número de pontos é igual ao número de intervalos entre pontos, o que significa que $h_x = 2/n_x$ e $h_y = 2/n_y$. Inclua no relatório a representação gráfica da solução encontrada.

c) A aproximação por diferenças finitas usada na alínea anterior para as segundas derivadas corresponde a um estêncil de 5 pontos (o ponto (i,j) mais 4 vizinhos), que introduz um erro local da ordem de h^2. A ordem deste erro pode ser reduzida para h^4 usando um estêncil de 9 pontos, baseado em $\frac{\partial^2 V(x,y)}{\partial x^2} \cong \frac{-V(x-2h,y)+16\,V(x-h,y)-30V(x,y)+16\,V(x+h,y)-V(x+2h,y)}{12\,h^2}$ para a derivada em x, e na aproximação correspondente para a derivada em y. Este estêncil de 9 pontos usa 4 pontos adicionais a uma distância de $2h$. Quando aplicado ao *método de Jacobi ponderado* (aqui, a ponderação traz estabilidade ao método), este estêncil produz a seguinte equação iterativa:

$$V_{i,j}{}^{(k)} = \frac{w}{60}\big[16V_{i-1,j}{}^{(k-1)} + 16V_{i+1,j}{}^{(k-1)} + 16V_{i,j-1}{}^{(k-1)} + 16V_{i,j+1}{}^{(k-1)} - V_{i-2,j}{}^{(k-1)} - V_{i+2,j}{}^{(k-1)} - V_{i,j-2}{}^{(k-1)} - V_{i,j+2}{}^{(k-1)} - 12h^2 f_{i,j}\big] + (1-w)V_{i,j}{}^{(k-1)}.$$

Modifique o programa desenvolvido na alínea b) de modo a aplicar este algoritmo, com o parâmetro de ponderação $w = 15/16$. Repare que, agora, cada processo tem de receber duas linhas e duas colunas dos seus processos vizinhos. Faça a representação gráfica do resultado e compare-a com a solução obtida na alínea anterior.

Caso não tenha conseguido completar a alínea b), use as condições fronteira da alínea a) e um estêncil de 9 pontos mais compacto, que, em vez dos 4 pontos a distância $2h$, usa os 4 pontos à distância $\sqrt{2}h$ na diagonal. A este estêncil corresponde a seguinte equação iterativa:

$$V_{i,j}{}^{(k)} = \frac{1}{20}\big[4V_{i-1,j}{}^{(k-1)} + 4V_{i+1,j}{}^{(k-1)} + 4V_{i,j-1}{}^{(k-1)} + 4V_{i,j+1}{}^{(k-1)} + V_{i-1,j-1}{}^{(k-1)} + V_{i+1,j-1}{}^{(k-1)} + V_{i-1,j+1}{}^{(k-1)} + V_{i+1,j+1}{}^{(k-1)}\big] - \frac{h^2}{40}\big[f_{i-1,j} + f_{i+1,j} + 8f_{i,j} + f_{i,j-1} + f_{i,j+1}\big].$$

Repare que, para usar este estêncil compacto, cada processo terá de comunicar com 8 processos no total. Ou seja, para atualizar os pontos que se encontram nos cantos dos subdomínio também tem de comunicar com os 4 processos vizinhos nas diagonais.

d) O método de relaxação de Gauss-Seidel é uma alternativa ao método de Jacobi, que converge mais rapidamente para a solução. Trata-se de numa modificação do método de Jacobi que consiste na utilização dos valores mais recentes que estiverem disponíveis em cada momento. Ou seja, para os vizinhos cujo $V^{(k)}$ ainda não tenha sido calculado na iteração k usa-se $V^{(k-1)}$ tal como no método de Jacobi, mas quando o $V^{(k)}$ do vizinho já foi calculado na iteração atual usa-se esse o valor mais recente.

Num programa sequencial (não paralelizado) bastaria substituir as linhas

```
for (i=1; i<=myrows; i++) {
    for (j=1; j<=mycols; j++) {
        Vnew[i][j] = (Vold[i-1][j] + Vold[i+1][j]
        + Vold[i][j-1] + Vold[i][j+1] - h*h*myf[i][j])/4.0;
```

pelas linhas

```
for (i=1; i<=myrows; i++) {
    for (j=1; j<=mycols; j++) {
        Vnew[i][j] = (Vnew[i-1][j] + Vnew[i+1][j]
        + Vnew[i][j-1] + Vnew[i][j+1] - h*h*myf[i][j])/4.0;
```

para obter o método de Gauss-Seidel. Esta alteração ao programa sequencial corresponde à implementação a equação iterativa ${V_{i,j}}^{(k)} = \frac{1}{4}\left[{V_{i-1,j}}^{(k)} + {V_{i,j-1}}^{(k)} + {V_{i+1,j}}^{(k-1)} + {V_{i,j+1}}^{(k-1)} - h^2 f_{i,j}\right]$. Porém, num programa paralelizado não seria possível usar esta equação para todos os pontos. Explique no relatório qual seria o problema.

A estratégia mais simples para superar esse problema é a utilização de um esquema de atualização do tipo vermelho-preto (ou par-ímpar). Resumidamente, neste esquema cada processo executa os seguintes passos numa iteração (incluindo duas fases distintas de comunicação):

(i) Inicialmente calcula os ${V_{i,j}}^{(k)}$ de todos os pontos para os quais $i + j$ é par, usando os valores de $V^{(k-1)}$ dos 4 pontos vizinhos na iteração $k - 1$. (Note que os 4 pontos vizinhos de um ponto 'par' são todos 'ímpares', e vice-versa. Note também que para determinar corretamente a paridade de um ponto, devem de ser usados os índices i e j correspondentes à matriz global.);

(ii) Comunica aos processos vizinhos os valores atualizados dos pontos 'pares' na fronteira do subdomínio;

(iii) Calcula os valores ${V_{i,j}}^{(k)}$ com $i + j$ ímpar, usando os 4 valores dos pontos vizinhos que foram atualizados no passo (i) (e que são todos 'pares');

(iv) Finalmente, comunica os pontos 'ímpares' aos processos vizinhos, para serem usados na próxima iteração.

Partindo do programa da alínea b), converta o método de Jacobi no método de Gauss-Seidel aplicando este esquema de atualização. Tal como na alínea b), use o estêncil de 5 pontos com condições fronteira periódicas. Se não resolveu a alínea b) (apenas nesse caso) use as condições

fronteira da alínea a). Deve pesquisar os pormenores este algoritmo na bibliografia. Sugere-se que comece pela leitura do capítulo 2, secção 2.2, do livro de Jianping Zhu, *Solving Partial Differential Equations on Parallel Computers*, World Scientific, 1994.