هیدرولیک محاسباتی

معادله دوبعدی پخش
(گام‌های جزئی)

محمدرضا هادیان

دانشگاه یزد- دانشکده مهندسی عمران

معادله دو بعدی پخش غیرماندگار

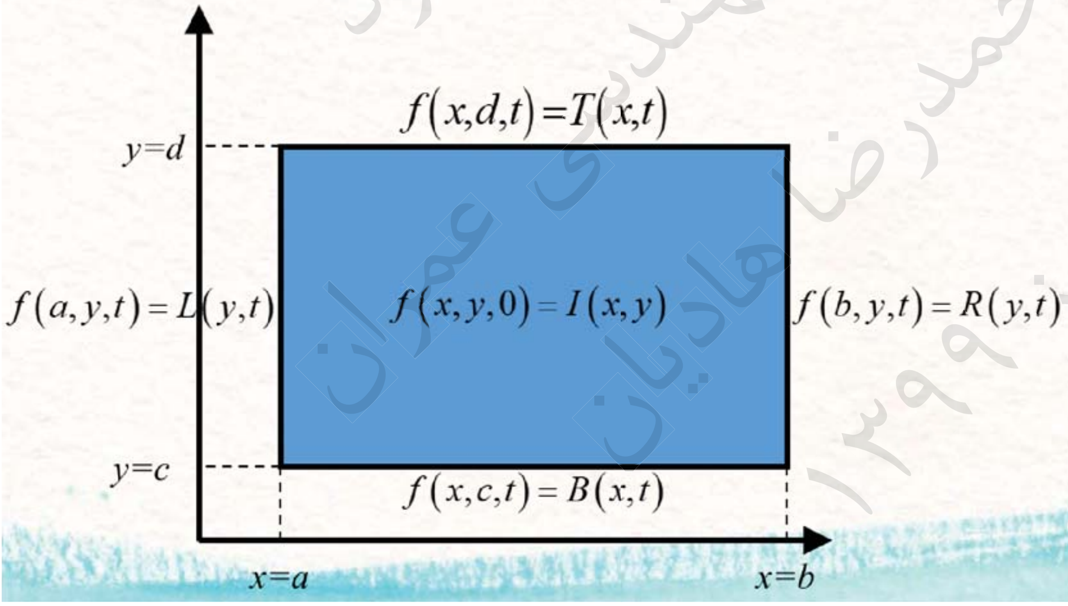
❑ معادله حاکم:

$$f_t = \alpha_x f_{xx} + \alpha_y f_{yy}$$

$$\frac{\partial f}{\partial t} = \alpha_x \frac{\partial^2 f}{\partial x^2} + \alpha_y \frac{\partial^2 f}{\partial y^2}$$

$$b \ge x \ge a$$

$$d \ge y \ge c$$



$$\Delta x = \frac{b-a}{M-1}$$

$$i = 1 \cdots M$$

$$\Delta y = \frac{d-c}{N-1}$$

$$j = 1 \cdots N$$

روش‌های گام‌های جزئی

- Fractional Steps
- Time Splitting Method
- گام زمانی به ۲ یا چند گام زمانی کوچکتر تقسیم می‌شود.

Locally One Dimensional (LOD)
روش‌های بطور موضعی یک بعدی

Alternating Direction Implicit (ADI)
روش‌های جهت‌های متناوب

روش‌های بطور موضعی یک بعدی

- گام زمانی بسته به نیاز به ۲ یا چند گام جزئی تقسیم می‌شود.

$$f_t = \alpha_x f_{xx} + \alpha_y f_{yy} \qquad \left[n\Delta t, (n+1)\Delta t\right]$$

$$\left[n\Delta t, (n+1)\Delta t\right] = \left[n\Delta t, (n+\tfrac{1}{2})\Delta t\right] \cup \left[(n+\tfrac{1}{2})\Delta t, (n+1)\Delta t\right]$$

$$\frac{1}{2} f_t = \alpha_x f_{xx} \qquad \left[n\Delta t, (n+\tfrac{1}{2})\Delta t\right]$$

$$\frac{1}{2} f_t = \alpha_y f_{yy} \qquad \left[(n+\tfrac{1}{2})\Delta t, (n+1)\Delta t\right]$$

- برای حل معادله در هر یک از گام‌های جزئی می‌توان بسته به نیاز یک روش مستقلی را بکار برد.

روش‌های بطور موضعی یک بعدی

- با استفاده از روش FTCS برای هر دو گام جزئی:

$$\frac{1}{2}\frac{f_{i,j}^{n+\frac{1}{2}}-f_{i,j}^{n}}{\frac{\Delta t}{2}}=\alpha_x\frac{f_{i-1,j}^{n}-2f_{i,j}^{n}+f_{i+1,j}^{n}}{(\Delta x)^2}$$

$$\frac{1}{2}\frac{f_{i,j}^{n+1}-f_{i,j}^{n+\frac{1}{2}}}{\frac{\Delta t}{2}}=\alpha_y\frac{f_{i,j-1}^{n+\frac{1}{2}}-2f_{i,j}^{n+\frac{1}{2}}+f_{i,j+1}^{n+\frac{1}{2}}}{(\Delta y)^2}$$

$$f_{i,j}^{n+\frac{1}{2}}=S_x\left(f_{i-1,j}^{n}+f_{i+1,j}^{n}\right)+\left(1-2S_x\right)f_{i,j}^{n}$$

$$f_{i,j}^{n+1}=S_y\left(f_{i,j-1}^{n+\frac{1}{2}}+f_{i,j+1}^{n+\frac{1}{2}}\right)+\left(1-2S_y\right)f_{i,j}^{n+\frac{1}{2}}$$

- ثابت می کنند که حوزه پایداری و مرتبه دقت تغییری نمی کند.

روش‌های بطور موضعی یک بعدی

- با استفاده از روش BTCS برای هر دو گام جزئی:

$$\frac{1}{2}\frac{f_{i,j}^{n+\frac{1}{2}}-f_{i,j}^{n}}{\frac{\Delta t}{2}}=\alpha_x\frac{f_{i-1,j}^{n+\frac{1}{2}}-2f_{i,j}^{n+\frac{1}{2}}+f_{i+1,j}^{n+\frac{1}{2}}}{(\Delta x)^2}$$

$$\frac{1}{2}\frac{f_{i,j}^{n+1}-f_{i,j}^{n+\frac{1}{2}}}{\frac{\Delta t}{2}}=\alpha_y\frac{f_{i,j-1}^{n+1}-2f_{i,j}^{n+1}+f_{i,j+1}^{n+1}}{(\Delta y)^2}$$

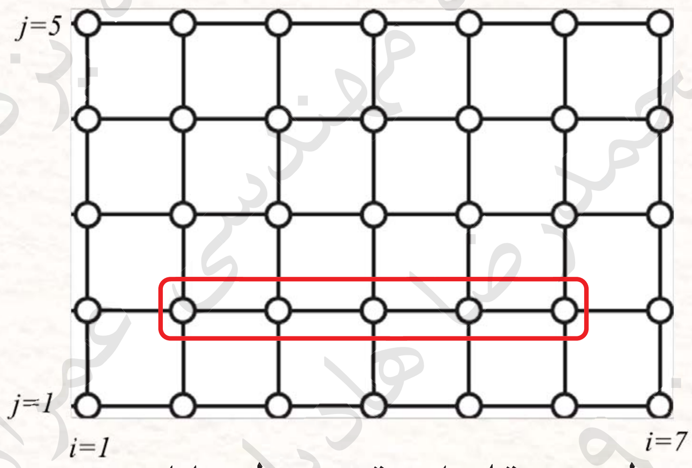
$$-S_x\left(f_{i-1,j}^{n+\frac{1}{2}}+f_{i+1,j}^{n+\frac{1}{2}}\right)+\left(1+2S_x\right)f_{i,j}^{n+\frac{1}{2}}=f_{i,j}^{n}$$

$$-S_y\left(f_{i,j-1}^{n+1}+f_{i,j+1}^{n+1}\right)+\left(1+2S_y\right)f_{i,j}^{n+1}=f_{i,j}^{n+\frac{1}{2}}$$

روش‌های بطور موضعی یک بعدی

- ❑ نیم گام اول:

$$-S_x f_{i-1,j}^{n+\frac{1}{2}} + \left(1+2S_x\right) f_{i,j}^{n+\frac{1}{2}} - S_x f_{i+1,j}^{n+\frac{1}{2}} = f_{i,j}^{n}$$

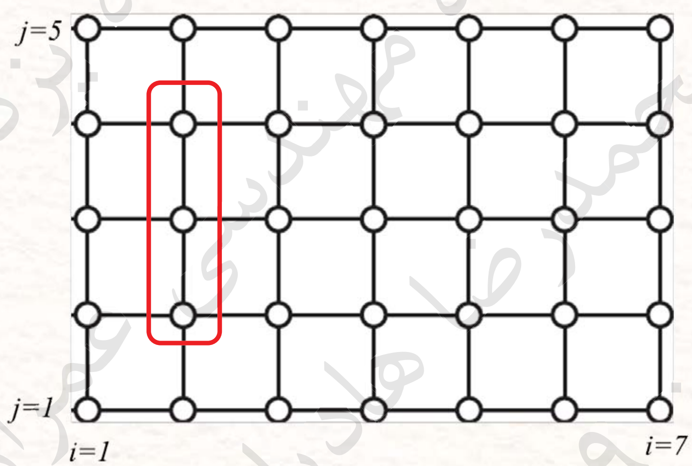


- ❑ محاسبات هر سطر مستقل از بقیه سطرها است.
- ❑ دستگاه معادلات هر سطر، ماتریس ضرائب سه قطری دارد.

روش‌های بطور موضعی یک بعدی

- ❑ نیم گام اول:

$$-S_y f_{i,j-1}^{n+1} + \left(1+2S_y\right) f_{i,j}^{n+1} - S_y f_{i,j+1}^{n+1} = f_{i,j}^{n+\frac{1}{2}}$$



- ❑ محاسبات هر ستون مستقل از بقیه ستون‌ها است.
- ❑ دستگاه معادلات هر ستون، ماتریس ضرائب سه قطری دارد.

روش‌های جهت‌های متناوب

- گام زمانی به تعداد بعدهای مسئله به گام‌های جزئی تقسیم می‌شود.

$$f_t = \alpha_x f_{xx} + \alpha_y f_{yy} \qquad \left[n\Delta t, (n+1)\Delta t\right]$$

$$\left[n\Delta t, (n+1)\Delta t\right] = \left[n\Delta t, (n+\tfrac{1}{2})\Delta t\right] \cup \left[(n+\tfrac{1}{2})\Delta t, (n+1)\Delta t\right]$$

- در هر گام جزئی یکی از جهت‌ها به عنوان جهت با محاسبات ضمنی در نظر گرفته می‌شود.

روش‌های جهت‌های متناوب

گام زمانی جزئی اول

$$\left[n\Delta t, (n+\tfrac{1}{2})\Delta t\right]$$

$$f_t\Big|_{i,j}^{n} \cong \frac{f_{i,j}^{n+\frac{1}{2}} - f_{i,j}^{n}}{\frac{\Delta t}{2}}$$

$$f_{xx}\Big|_{i,j}^{n} \cong \frac{f_{i-1,j}^{n+\frac{1}{2}} - 2 f_{i,j}^{n+\frac{1}{2}} + f_{i+1,j}^{n+\frac{1}{2}}}{(\Delta x)^2}$$

$$f_{yy}\Big|_{i,j}^{n} \cong \frac{f_{i,j-1}^{n} - 2 f_{i,j}^{n} + f_{i,j+1}^{n}}{(\Delta y)^2}$$

گام زمانی جزئی دوم

$$\left[(n+\tfrac{1}{2})\Delta t, (n+1)\Delta t\right]$$

$$f_t\Big|_{i,j}^{n+\frac{1}{2}} \cong \frac{f_{i,j}^{n+1} - f_{i,j}^{n+\frac{1}{2}}}{\frac{\Delta t}{2}}$$

$$f_{xx}\Big|_{i,j}^{n+\frac{1}{2}} \cong \frac{f_{i-1,j}^{n+\frac{1}{2}} - 2 f_{i,j}^{n+\frac{1}{2}} + f_{i+1,j}^{n+\frac{1}{2}}}{(\Delta x)^2}$$

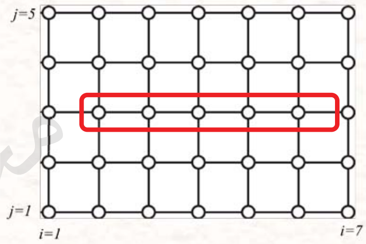
$$f_{yy}\Big|_{i,j}^{n+\frac{1}{2}} \cong \frac{f_{i,j-1}^{n+1} - 2 f_{i,j}^{n+1} + f_{i,j+1}^{n+1}}{(\Delta y)^2}$$

دانشگاه یزد دانشکده مهندسی عمران محمدرضا هادیان پاییز ۱۳۹۹

روش‌های جهت‌های متناوب

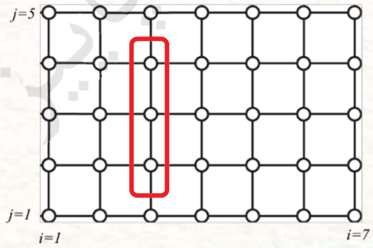
$$-\frac{S_x}{2} f_{i-1,j}^{n+\frac{1}{2}} + \left(1+S_x\right) f_{i,j}^{n+\frac{1}{2}} - \frac{S_x}{2} f_{i+1,j}^{n+\frac{1}{2}} = \frac{S_y}{2} f_{i,j-1}^{n} + \left(1-S_y\right) f_{i,j}^{n} + \frac{S_y}{2} f_{i,j+1}^{n}$$

محاسبات هر سطر مستقل از بقیه سطر ها است.



$$-\frac{S_y}{2} f_{i,j-1}^{n+1} + \left(1+S_y\right) f_{i,j}^{n+1} - \frac{S_y}{2} f_{i,j+1}^{n+1} = \frac{S_x}{2} f_{i-1,j}^{n+\frac{1}{2}} + \left(1-S_x\right) f_{i,j}^{n+\frac{1}{2}} + \frac{S_x}{2} f_{i+1,j}^{n+\frac{1}{2}}$$

محاسبات هر ستون مستقل از بقیه ستون‌ها است.



روش پایدار نامشروط و حل‌پذیر است.

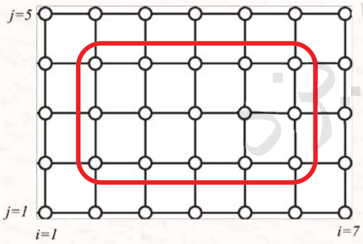
مزایای روش‌های گام‌های جزئی

- ❑ استفاده از روش‌های متفاوت و مناسب برای هر گام
- ❑ ماتریس ضرائب سه قطری در روش‌های ضمنی
- ❑ امکان استفاده از پردازش موازی

مقایسه حجم محاسبات !!!

❑ استفاده از روش ضمنی بصورت عادی:

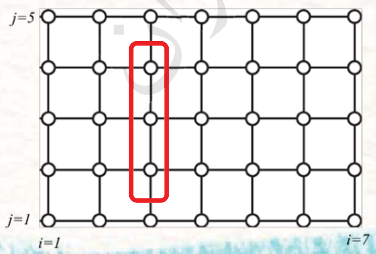
❖ یک دستگاه ۱۵ معادله‌ای با ماتریس ضرائب سه قطری بلوکی

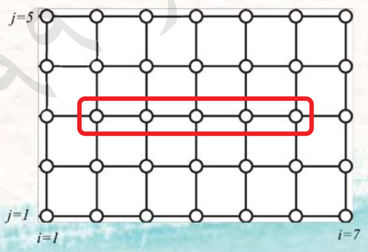


❑ روش گام‌های جزئی:

❖ ۳ دستگاه ۵ معادله‌ای با ماتریس ضرائب سه قطری

❖ ۵ دستگاه ۳ معادله‌ای با ماتریس ضرائب سه قطری





پردازش موازی

❑ پردازش سریال: استفاده از یک پردازنده

❑ پردازش موازی:

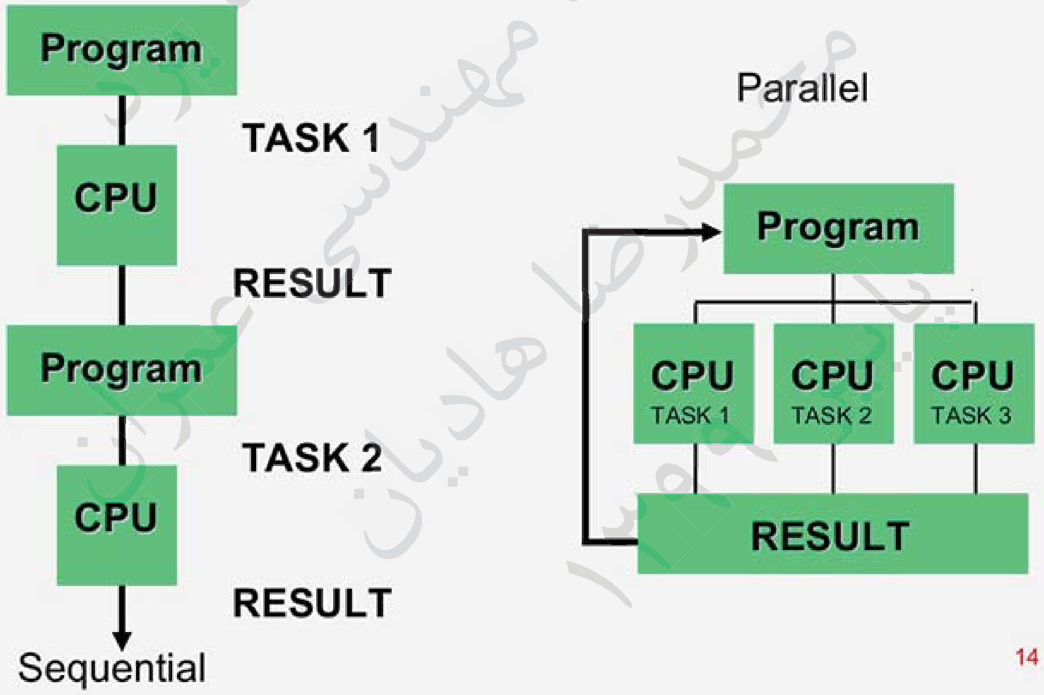
❖ کامپیوترهای چند هسته‌ای

❖ استفاده از چند کامپیوتر یک یا چند هسته‌ای (کلاستر)

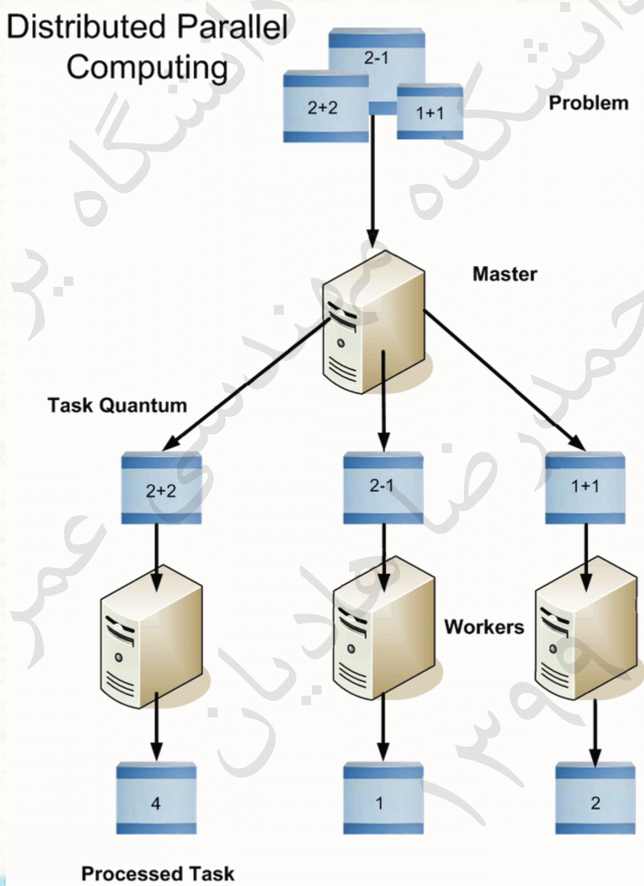
❖ استفاده از کارت گرافیک

پردازش موازی

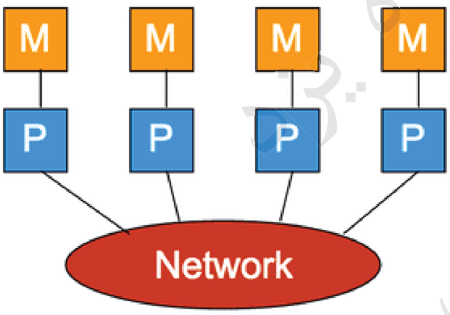
Sequential and parallel processing

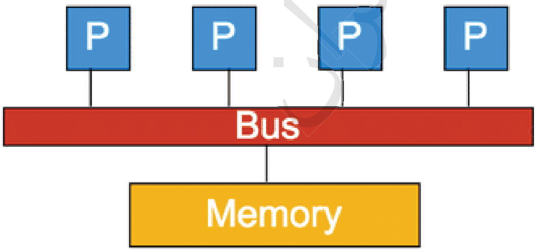


پردازش موازی



Shared vs Distributed Memory





- Distributed memory
  - Each processor has its own local memory
  - Message-passing is used to exchange data between processors
- Shared memory
  - Single address space
  - All processes have access to the pool of shared memory

Why GPU is faster than CPU ?

The GPU Devotes More Transistors to Data Processing.

[CUDA C Programming Guide Version 3.2]

برنامه ریزی موازی و سریال

❑ حل دستگاه معادلات با روش سعی و خطا

$$\begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix}$$

$$\sum_{j=1}^{n} a_{ij} x_j = b_i$$

$$\begin{cases} x_1 = \dfrac{1}{a_{11}}\left(b_1 - a_{12}x_2 + a_{13}x_3 + \cdots + a_{1n}x_n\right) \\ x_2 = \dfrac{1}{a_{22}}\left(b_2 - a_{21}x_1 + a_{23}x_3 + \cdots + a_{2n}x_n\right) \\ \vdots \\ x_n = \dfrac{1}{a_{nn}}\left(b_n - a_{n1}x_1 + a_{n2}x_2 + \cdots + a_{n(n-1)}x_{n-1}\right) \end{cases}$$

$$x_i = \frac{1}{a_{ii}}\left(b_i - \sum_{j \neq i} a_{ij} x_j\right)$$

برنامه ریزی موازی و سریال

❑ حل دستگاه معادلات با روش سعی و خطا

❖ Jacobi

$$\begin{cases} x_1^{k+1} = \dfrac{1}{a_{11}}\left(b_1 - a_{12}x_2^k + a_{13}x_3^k + \cdots + a_{1n}x_n^k\right) \\ x_2^{k+1} = \dfrac{1}{a_{22}}\left(b_2 - a_{21}x_1^k + a_{23}x_3^k + \cdots + a_{2n}x_n^k\right) \\ \vdots \\ x_n^{k+1} = \dfrac{1}{a_{nn}}\left(b_n - a_{n1}x_1^k + a_{n2}x_2^k + \cdots + a_{n(n-1)}x_{n-1}^k\right) \end{cases}$$

$$x_i^{k+1} = \frac{1}{a_{ii}}\left(b_i - \sum_{j \neq i} a_{ij} x_j^k\right)$$

دانشگاه یزد
دانشکده مهندسی عمران
محمدرضا هادیان
پاییز ۱۳۹۹

برنامه ریزی موازی و سریال

- حل دستگاه معادلات با روش سعی و خطا
  - Gauss-Seidel ❖

$$
\begin{cases}
x_1^{k+1} = \dfrac{1}{a_{11}}\left(b_1 - a_{12}x_2^k + a_{13}x_3^k + \cdots + a_{1n}x_n^k\right) \\
x_2^{k+1} = \dfrac{1}{a_{22}}\left(b_2 - a_{21}x_1^{k+1} + a_{23}x_3^k + \cdots + a_{2n}x_n^k\right) \\
\vdots \\
x_n^{k+1} = \dfrac{1}{a_{nn}}\left(b_n - a_{n1}x_1^{k+1} + a_{n2}x_2^{k+1} + \cdots + a_{n(n-1)}x_{n-1}^{k+1}\right)
\end{cases}
$$

$$
x_i^{k+1} = \frac{1}{a_{ii}}\left(b_i - \sum_{j<i} a_{ij}x_j^{k+1} - \sum_{j>i} a_{ij}x_j^k\right)
$$

برنامه ریزی موازی و سریال

- حل دستگاه معادلات با روش سعی و خطا

$$
\begin{cases}
x_1^{k+1} = \dfrac{1}{a_{11}}\left(b_1 - a_{12}x_2^k + a_{13}x_3^k + \cdots + a_{1n}x_n^k\right) \\
x_2^{k+1} = \dfrac{1}{a_{22}}\left(b_2 - a_{21}x_1^{k+1} + a_{23}x_3^k + \cdots + a_{2n}x_n^k\right) \\
\vdots \\
x_n^{k+1} = \dfrac{1}{a_{nn}}\left(b_n - a_{n1}x_1^{k+1} + a_{n2}x_2^{k+1} + \cdots + a_{n(n-1)}x_{n-1}^{k+1}\right)
\end{cases}
$$

$$
x_i^{k+1} = \frac{1}{a_{ii}}\left(b_i - \sum_{j<i} a_{ij}x_j^{k+1} - \sum_{j>i} a_{ij}x_j^k\right)
$$

Gauss-Seidel

$$
\begin{cases}
x_1^{k+1} = \dfrac{1}{a_{11}}\left(b_1 - a_{12}x_2^k + a_{13}x_3^k + \cdots + a_{1n}x_n^k\right) \\
x_2^{k+1} = \dfrac{1}{a_{22}}\left(b_2 - a_{21}x_1^k + a_{23}x_3^k + \cdots + a_{2n}x_n^k\right) \\
\vdots \\
x_n^{k+1} = \dfrac{1}{a_{nn}}\left(b_n - a_{n1}x_1^k + a_{n2}x_2^k + \cdots + a_{n(n-1)}x_{n-1}^k\right)
\end{cases}
$$

$$
x_i^{k+1} = \frac{1}{a_{ii}}\left(b_i - \sum_{j\neq i} a_{ij}x_j^k\right)
$$

Jacobi

برنامه ریزی موازی و سریال

- ❑ حل دستگاه معادلات با روش سعی و خطا
  - ❖ Jacobi
    - ✓ تعداد گام‌های تکرار بیشتر
    - ✓ مناسب برای پردازش موازی
  - ❖ Gauss-Seidel
    - ✓ تعداد گام‌های تکرار کمتر
    - ✓ مناسب برای پردازش سریال

برنامه ریزی موازی و سریال

- ❑ برای اجرای برنامه بصورت موازی، باید الگوریتم مورد استفاده قابلیت موازی‌سازی را داشته باشد.
- ❑ پیاده سازی در کد کامپیوتری:
  - ❖ MPI
  - ❖ OpenMP
  - ❖ CUDA
- ❑ کتابخانه‌های آماده:
  - ❖ PETSc، Lis، hypre، ScaLAPACK، PBLAS
  - ❖ MATLAB Parallel Computing Toolbox
  - ❖ Intel Math Kernel Library (MKL)

دانشگاه یزد
دانشکده مهندسی عمران
محمدرضا هادیان
پاییز ۱۳۹۹

Any Question?