PART THREE

قسمت سوم

Linear Systems

سیستم های خطی

دستگاه معادلات

غیرخطی:

$$
\begin{aligned}
f_1(x_1, x_2, \ldots, x_n) &= 0\\
f_2(x_1, x_2, \ldots, x_n) &= 0\\
&\vdots\\
f_n(x_1, x_2, \ldots, x_n) &= 0
\end{aligned}
$$

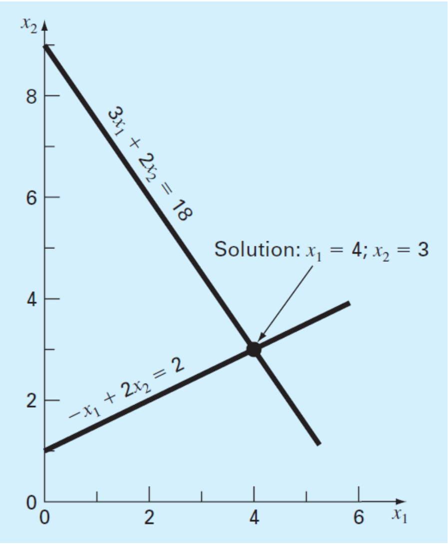
می توان معادلات غیرخطی را به فرم زیر خطی سازی نمود

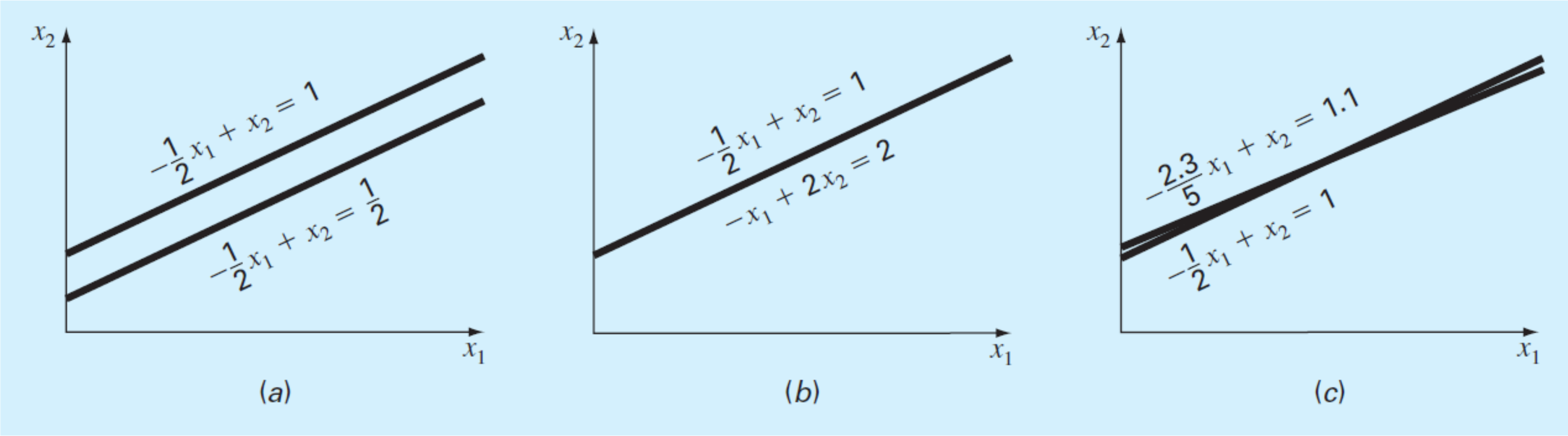
خطی:

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

مفهوم دستگاه معادلات خطی و حالات مختلف آن

$$3x_1 + 2x_2 = 18$$
$$-x_1 + 2x_2 = 2$$





a) سیستم بدون جواب

b) سیستم دارای بینهایت جواب

c) سیستم مریض (ill condition)

8.1.3 Representing Linear Algebraic Equations in Matrix Form

بیان معادلات جبری خطی به فرم ماتریسی

$$a_{11}x_1 + a_{12}x_2 + a_{13}x_3 = b_1$$
$$a_{21}x_1 + a_{22}x_2 + a_{23}x_3 = b_2$$
$$a_{31}x_1 + a_{32}x_2 + a_{33}x_3 = b_3$$

$$[A]\{x\} = \{b\}$$

- ماتریس ضرائب

$$[A] = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix}$$

- بردار مقادیر ثابت

$$\{b\}^T = \lfloor b_1 \quad b_2 \quad b_3 \rfloor$$

- بردار مجهولات

$$\{x\}^T = \lfloor x_1 \quad x_2 \quad x_3 \rfloor$$

- حل دستگاه

$$[A]^{-1}[A]\{x\} = [A]^{-1}\{b\}$$
$$\{x\} = [A]^{-1}\{b\}$$

سیستم های overdetermined و underdetermined

- اگر تعداد معادلات (ردیفها) از تعداد مجهولات (ستونها) بیشتر باشد $m > n$ به آن سیستم Overdetermined و به سیستم های دارای تعداد مجهولات بیشتر از معادلات $m < n$ ، underdetermined گفته می شود.

- برازش یک منحنی مرتبه n بر روی m نقطه (m>n) یک مسئله Overdetermined است.
- بهینه سازی عددی نیز یک مسئله underdetermined است.

حل دستگاه معادلات جبری با استفاده از متلب

- استفاده از backslash

```
>> x = A\b
```

- استفاده از ماتریس معکوس

```
>> x = inv(A)*b
```

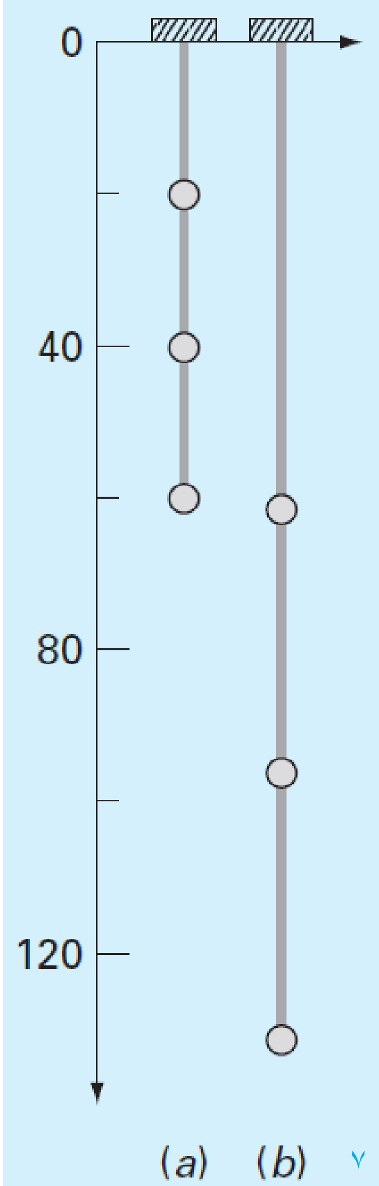
- استفاده از backslash نسبت به ماتریس معکوس برای حل دستگاه معادلات جبری بهینه تر می باشد.

مثال:

- سه نفر با مشخصات زیر توسط سه بانجی پرش می نمایند. طول هریک از این بانجی ها ۲۰ متر می باشد و ضریب فنریت آنها نیز در جدول زیر داده شده است. مطلوبست موقعیت نهایی این سه نفر در لحظه تعادل پس از پرش.

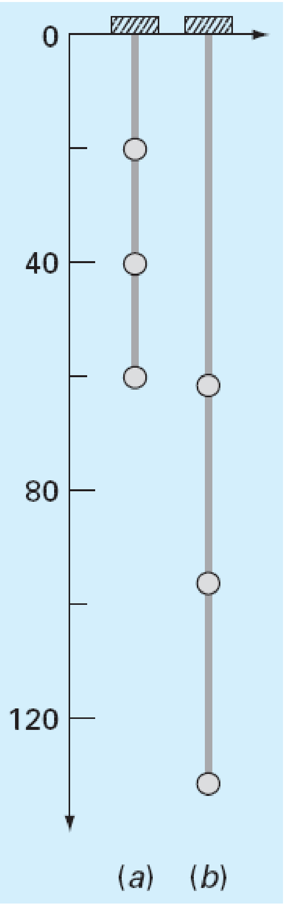
Jumper	Mass (kg)	Spring Constant (N/m)	Unstretched Cord Length (m)
Top (1)	60	50	20
Middle (2)	70	100	20
Bottom (3)	80	50	20

$$\begin{bmatrix} 150 & -100 & 0 \\ -100 & 150 & -50 \\ 0 & -50 & 50 \end{bmatrix} \begin{Bmatrix} x_1 \\ x_2 \\ x_3 \end{Bmatrix} = \begin{Bmatrix} 588.6 \\ 686.7 \\ 784.8 \end{Bmatrix}$$



مثال:

```
>> K = [150 -100 0;-100 150 -50;0 -50 50]

K =
   150  -100     0
  -100   150   -50
     0   -50    50

>> mg = [588.6; 686.7; 784.8]

mg =
  588.6000
  686.7000
  784.8000
```




```
>> x = K\mg

x =
   41.2020
   55.9170
   71.6130
```

```
>> x = inv(K)*mg

x =
   41.2020
   55.9170
   71.6130
```

```
>> xi = [20;40;60];
>> xf = x+xi

xf =
   61.2020
   95.9170
  131.6130
```

9 فصل نهم

Gauss Elimination

روش حذفی گاوس
و روشهای حل مستقیم

- روش کرامر برای سیستم های معادلات کوچک که محاسبه دترمینان آنها ساده است کاربرد دارد:

$$[A]\{x\} = \{b\}$$

$$[A] = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix}$$

$$D = \begin{vmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{vmatrix}$$

$$x_1 = \frac{\begin{vmatrix} b_1 & a_{12} & a_{13} \\ b_2 & a_{22} & a_{23} \\ b_3 & a_{32} & a_{33} \end{vmatrix}}{D}$$

9.2 NAIVE GAUSS ELIMINATION

روش حذفی گاوس ساده

$$\left[\begin{array}{ccc|c} a_{11} & a_{12} & a_{13} & b_1 \\ a_{21} & a_{22} & a_{23} & b_2 \\ a_{31} & a_{32} & a_{33} & b_3 \end{array}\right]$$

$$\downarrow$$

$$\left[\begin{array}{ccc|c} a_{11} & a_{12} & a_{13} & b_1 \\ & a'_{22} & a'_{23} & b'_2 \\ & & a''_{33} & b''_3 \end{array}\right]$$

(*a*) Forward elimination

$$\downarrow$$

$$x_3 = b''_3/a''_{33}$$

$$x_2 = (b'_2 - a'_{23}x_3)/a'_{22}$$

$$x_1 = (b_1 - a_{13}x_3 - a_{12}x_2)/a_{11}$$

(*b*) Back substitution

حل دستگاه معادلات با این روش شامل دو مرحله می باشد:

۱- حذف مجهولات از معادلات و تبدیل دستگاه به یک ماتریس مثلثی (حذف مستقیم).

۲- محاسبه مجهول آخر و جایگذاری در معادلات قبلی و تکرار روند فوق (جایگذاری معکوس).

روش ساختن ماتریس بالا مثلثی (حذف مستقیم)

از معادله اول برای حذف جمله اول معادله دوم استفاده می شود. بدین منظور کلیه جملات معادله اول را در ضریب a_{21}/a_{11} ضرب می کنیم بنابراین معادله به فرم زیر تبدیل می شود.

$$a_{21}x_1 + \frac{a_{21}}{a_{11}}a_{12}x_2 + \frac{a_{21}}{a_{11}}a_{13}x_3 + \cdots + \frac{a_{21}}{a_{11}}a_{1n}x_n = \frac{a_{21}}{a_{11}}b_1$$

حال با کسر کردن این معادله از معادله دوم جمله اول حذف می شود و بقیه جملات معادله دوم به صورت زیر می باشد.

$$\left(a_{22} - \frac{a_{21}}{a_{11}}a_{12}\right)x_2 + \cdots + \left(a_{2n} - \frac{a_{21}}{a_{11}}a_{1n}\right)x_n = b_2 - \frac{a_{21}}{a_{11}}b_1$$

or

$$a'_{22}x_2 + \cdots + a'_{2n}x_n = b'_2$$

روند فوق تا ایجاد ماتریس بالا مثلثی ادامه می یابد. به این روند نرمال کردن و به معادله اول معادله محور (Pivot Equation) و به جمله اول که توسط آن بقیه جملات حذف می شود جمله محور (Pivot Element) گویند.

جایگذاری مقدار یافته شده در معادلات قبلی (جایگذاری معکوس)

- پس از اینکه ماتریس ضرایب به فرم بالا مثلثی درآمد می توان مجهول آخر را محاسبه نمود.

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + a_{13}x_3 + \cdots + a_{1n}x_n &= b_1 \\ a'_{22}x_2 + a'_{23}x_3 + \cdots + a'_{2n}x_n &= b'_2 \\ a''_{33}x_3 + \cdots + a''_{3n}x_n &= b''_3 \\ \ddots \qquad\qquad &\ \ \vdots \\ a_{nn}^{(n-1)}x_n &= b_n^{(n-1)} \end{aligned}$$

$$x_n = \frac{b_n^{(n-1)}}{a_{nn}^{(n-1)}}$$

- سپس می توان بقیه مجهولات را به ترتیب از آخر به اول محاسبه کرد:

$$x_i = \frac{b_i^{(i-1)} - \sum_{j=i+1}^{n} a_{ij}^{(i-1)} x_j}{a_{ii}^{(i-1)}} \qquad \text{for } i = n-1, n-2, \ldots, 1$$

- یکی از مشکلات روش حذفی گاوس زمانی است که جمله ای که می بایست با آن بقیه جملات را حذف نمود صفر و یا نزدیک صفر باشد. بنابراین در حین عملیات حذف تقسیم بر صفر اتفاق می افتد و یا خطای گرد کردن افزایش می یابد.
- به همین دلیل بهتر است قبل از انجام عملیات بالا مثلثی و نرمال کردن ابتدا بزرگترین جمله از نظر اندازه در هر ردیف یافته شود و آن جمله برای حذف انتخاب گردد. به این روش محورگیری جزیی (Partial Pivoting) گویند.
- می توان بزرگترین جمله در هر ردیف و هر ستون را یافت و برای حذف انتخاب نمود که به این روش محورگیری کلی (Complete Pivoting) گویند.
- محورگیری علاوه بر مزایای فوق برای سیستم های مریض (ill condition) نیز سودمند است زیرا باعث کاهش خطای گردکردن می شود.
- در عمل از محورگیری جزیی استفاده می شود زیرا محورگیری کلی باعث جابجایی مجهولات و پیچیده شدن برنامه می شود.

مثال

- دستگاه معادلات زیر را در نظر بگیرید.

$$0.0003x_1 + 3.0000x_2 = 2.0001$$
$$1.0000x_1 + 1.0000x_2 = 1.0000$$

- حل دقیق این دستگاه به شکل زیر است:

$$x_1 = 1/3 \text{ and } x_2 = 2/3.$$

$$\begin{aligned} 0.0003x_1 + 3.0000x_2 &= 2.0001 \\ 1.0000x_1 + 1.0000x_2 &= 1.0000 \end{aligned} \xrightarrow{\times\ 1/(0.0003)} \begin{aligned} x_1 + 10{,}000x_2 &= 6667 \\ -9999x_2 &= -6666 \end{aligned}$$

$$x_2 = 2/3 \Longrightarrow x_1 = \frac{\boxed{2.0001 - 3(2/3)}}{0.0003}$$

تفاضل دو عدد نزدیک به هم

Significant Figures	x_2	x_1	Absolute Value of Percent Relative Error for x_1
3	0.667	−3.33	1099
4	0.6667	0.0000	100
5	0.66667	0.30000	10
6	0.666667	0.330000	1
7	0.6666667	0.3330000	0.1

بررسی تاثیر استفاده از روش محورگیری جزئی در حذف مجهولات

- در صورتیکه معادلات جابجا شوند و سپس روند حذف مجهولات انجام شود:

$$1.0000x_1 + 1.0000x_2 = 1.0000$$
$$0.0003x_1 + 3.0000x_2 = 2.0001$$

$$x_2 = 2/3 \Rightarrow \qquad x_1 = \frac{\boxed{1 - (2/3)}}{1}$$

تفاضل دو عدد متفاوت

Significant Figures	x_2	x_1	Absolute Value of Percent Relative Error for x_1
3	0.667	0.333	0.1
4	0.6667	0.3333	0.01
5	0.66667	0.33333	0.001
6	0.666667	0.333333	0.0001
7	0.6666667	0.3333333	0.0000

- پس از آنکه ماتریس ضرایب به فرم مثلثی درآمد می توان با ضرب درایه های روی قطر اصلی دترمینان را محاسبه نمود.

$$D = a_{11}a_{22}a_{33}\cdots a_{nn}$$

$$D = \begin{vmatrix} a_{11} & a_{12} & a_{13} \\ 0 & a_{22} & a_{23} \\ 0 & 0 & a_{33} \end{vmatrix}$$

$$D = a_{11}\begin{vmatrix} a_{22} & a_{23} \\ 0 & a_{33} \end{vmatrix} - a_{12}\begin{vmatrix} 0 & a_{23} \\ 0 & a_{33} \end{vmatrix} + a_{13}\begin{vmatrix} 0 & a_{22} \\ 0 & 0 \end{vmatrix}$$

$$D = a_{11}a_{22}a_{33} - a_{12}(0) + a_{13}(0) = a_{11}a_{22}a_{33}$$

• بسیاری از سیستمهای مکانیکی دارای دستگاه معادلات با ماتریس ضرایب محدود (banded) هستند. برای این سیستم ها می توان روشهای حل بهینه تری استفاده نمود. از آنجا که تعدادی از درایه ها صفر می باشند لزومی به انجام مراحل حذفی برای آنها نیست و با توجه به نوع و ابعاد ماتریس می توان حدود حلقه های محاسباتی را به نحوی تعیین نمود که اصلا برای درایه های صفر عملیات حذف صورت نگیرد.

$$\begin{bmatrix} f_1 & g_1 & & & & & \\ e_2 & f_2 & g_2 & & & & \\ & e_3 & f_3 & g_3 & & & \\ & & \cdot & \cdot & \cdot & & \\ & & & \cdot & \cdot & \cdot & \\ & & & & \cdot & \cdot & \cdot \\ & & & & e_{n-1} & f_{n-1} & g_{n-1} \\ & & & & & e_n & f_n \end{bmatrix} \begin{Bmatrix} x_1 \\ x_2 \\ x_3 \\ \cdot \\ \cdot \\ \cdot \\ x_{n-1} \\ x_n \end{Bmatrix} = \begin{Bmatrix} r_1 \\ r_2 \\ r_3 \\ \cdot \\ \cdot \\ \cdot \\ r_{n-1} \\ r_n \end{Bmatrix}$$

مثال:

- انتقال گرما در یک میله دارای معادله دیفرانسیل زیر می باشد که می توان آن را به فرم گسسته نوشت:

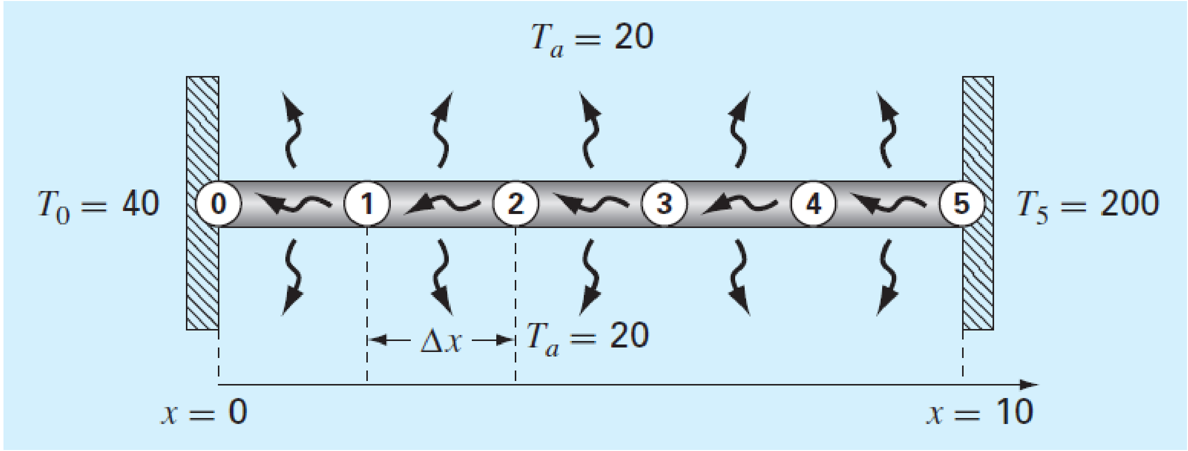
$$\frac{d^2T}{dx^2} + h'(T_a - T) = 0$$

گسسته سازی

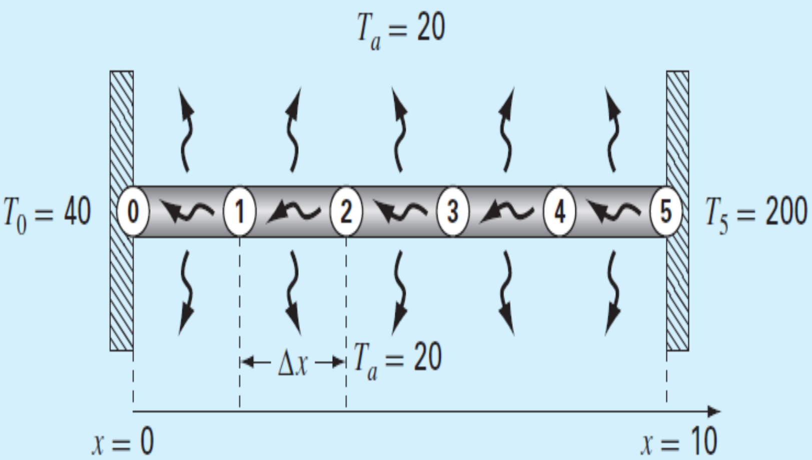
$$\frac{T_{i+1} - 2T_i + T_{i-1}}{\Delta x^2} + h'(T_a - T_i) = 0$$

$$\begin{aligned} -T_0 + 2.04T_1 - T_2 &= 0.8 \\ -T_1 + 2.04T_2 - T_3 &= 0.8 \\ -T_2 + 2.04T_3 - T_4 &= 0.8 \\ -T_3 + 2.04T_4 - T_5 &= 0.8 \end{aligned}$$

$$\begin{bmatrix} 2.04 & -1 & 0 & 0 \\ -1 & 2.04 & -1 & 0 \\ 0 & -1 & 2.04 & -1 \\ 0 & 0 & -1 & 2.04 \end{bmatrix} \begin{Bmatrix} T_1 \\ T_2 \\ T_3 \\ T_4 \end{Bmatrix} = \begin{Bmatrix} 40.8 \\ 0.8 \\ 0.8 \\ 200.8 \end{Bmatrix}$$

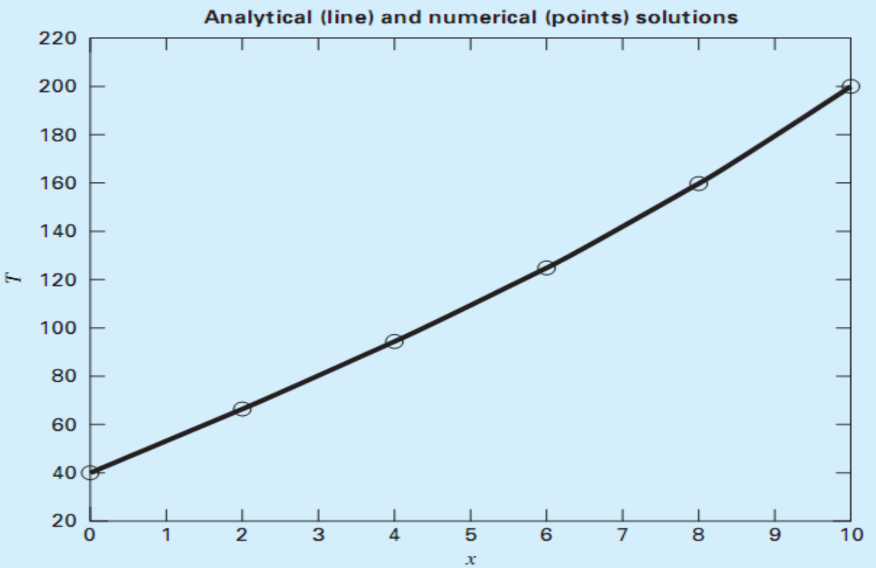


مثال:



$$\begin{bmatrix} 2.04 & -1 & 0 & 0 \\ -1 & 2.04 & -1 & 0 \\ 0 & -1 & 2.04 & -1 \\ 0 & 0 & -1 & 2.04 \end{bmatrix} \begin{Bmatrix} T_1 \\ T_2 \\ T_3 \\ T_4 \end{Bmatrix} = \begin{Bmatrix} 40.8 \\ 0.8 \\ 0.8 \\ 200.8 \end{Bmatrix}$$

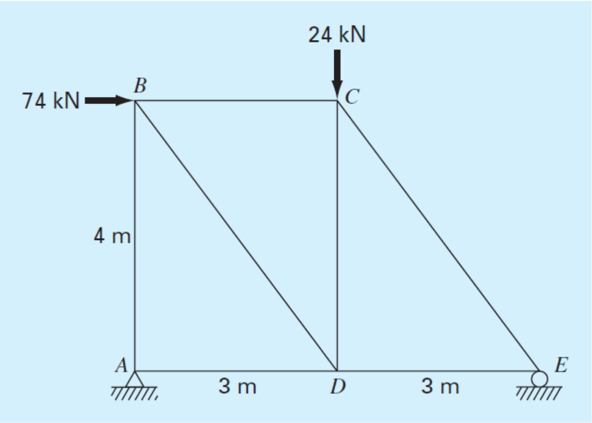
```
>> A=[2.04 -1 0 0
-1 2.04 -1 0
0 -1 2.04 -1
0 0 -1 2.04];
>> b=[40.8 0.8 0.8 200.8]';
>> T=(A\b)'

T =
   65.9698   93.7785  124.5382  159.4795
```



تمرین ۱:

- قرار است دستگاه معادلات مربوط به خرپای نشان داده شده را حل نمائیم. برنامه ای بنویسید که از روش حذفی گاوس با محورگیری جزیی این معادلات را حل نماید.

$$A_x + AD = 0 \qquad -24 - CD - (4/5)CE = 0$$
$$A_y + AB = 0 \qquad -AD + DE - (3/5)BD = 0$$
$$74 + BC + (3/5)BD = 0 \qquad CD + (4/5)BD = 0$$
$$-AB - (4/5)BD = 0 \qquad -DE - (3/5)CE = 0$$
$$-BC + (3/5)CE = 0 \qquad E_y + (4/5)CE = 0$$



تمرین ۲:

- برای حل دستگاه زیر برنامه ای بنویسید که از روش حذفی گاوس دستگاه محدود (banded) زیر را بصورت بهینه حل نماید.

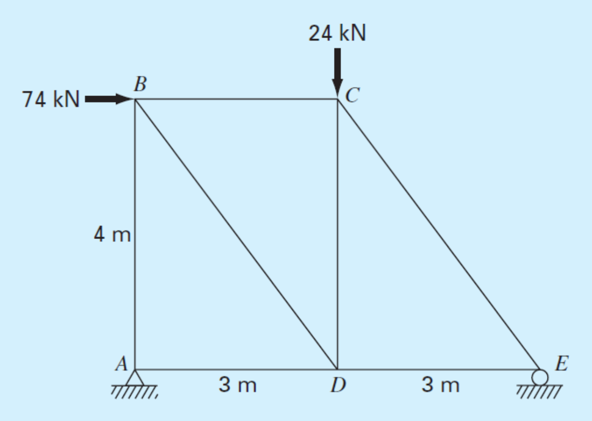
$$\begin{bmatrix} 8 & -2 & -1 & 0 & 0 \\ -2 & 9 & -4 & -1 & 0 \\ -1 & -3 & 7 & -1 & -2 \\ 0 & -4 & -2 & 12 & -5 \\ 0 & 0 & -7 & -3 & 15 \end{bmatrix} \begin{Bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{Bmatrix} = \begin{Bmatrix} 5 \\ 2 \\ 1 \\ 1 \\ 5 \end{Bmatrix}$$

10 فصل دهم

LU Factorization

حل دستگاه معادلات به روش جداسازی ماتریس ضرایب به ماتریسهای بالا و پائین مثلثی

- روش حذفی گاوس برای حل مسائل مناسب است اما برای حل مسائلی که ماتریس ضرایب یکسان و مقادیر ثابت متفاوتی دارند بهینه نیست. زیرا در حین عملیات حذف مجهولات (که بسیار زمانبر است)، عملیات بر روی مقادیر ثابت نیز اعمال شده و این مقادیر تغییر می کنند.
- بطور مثال خرپای نشان داده شده در شکل زیر را در نظر بگیرید. اگر بخواهیم بارگذاری وارد شده را تغییر داده و دوباره مسئله را از روش گاوس حل نمائیم لازم است کلیه مراحل حذف از ابتدا صورت گیرد.
- روش جداسازی ماتریس ضرائب به ماتریس های بالا و پائین مثلثی فرآیند زمان بر حذف مجهولات را از مقادیر ثابت جدا می کند.



10.1 OVERVIEW OF *LU* FACTORIZATION تفکیک به ماتریسهای بالا و پائین مثلثی

دستگاه معادلات مفروض:

$$[A]\{x\} - \{b\} = 0$$

با استفاده از روش حذفی گاوس:

$$\begin{bmatrix} u_{11} & u_{12} & u_{13} \\ 0 & u_{22} & u_{23} \\ 0 & 0 & u_{33} \end{bmatrix} \begin{Bmatrix} x_1 \\ x_2 \\ x_3 \end{Bmatrix} = \begin{Bmatrix} d_1 \\ d_2 \\ d_3 \end{Bmatrix}$$

$$[U]\{x\} - \{d\} = 0$$

حال فرض کنید ماتریسی مانند ماتریس روبرو وجود داشته باشد که با پیش ضرب آن در معادله فوق داشته باشیم:

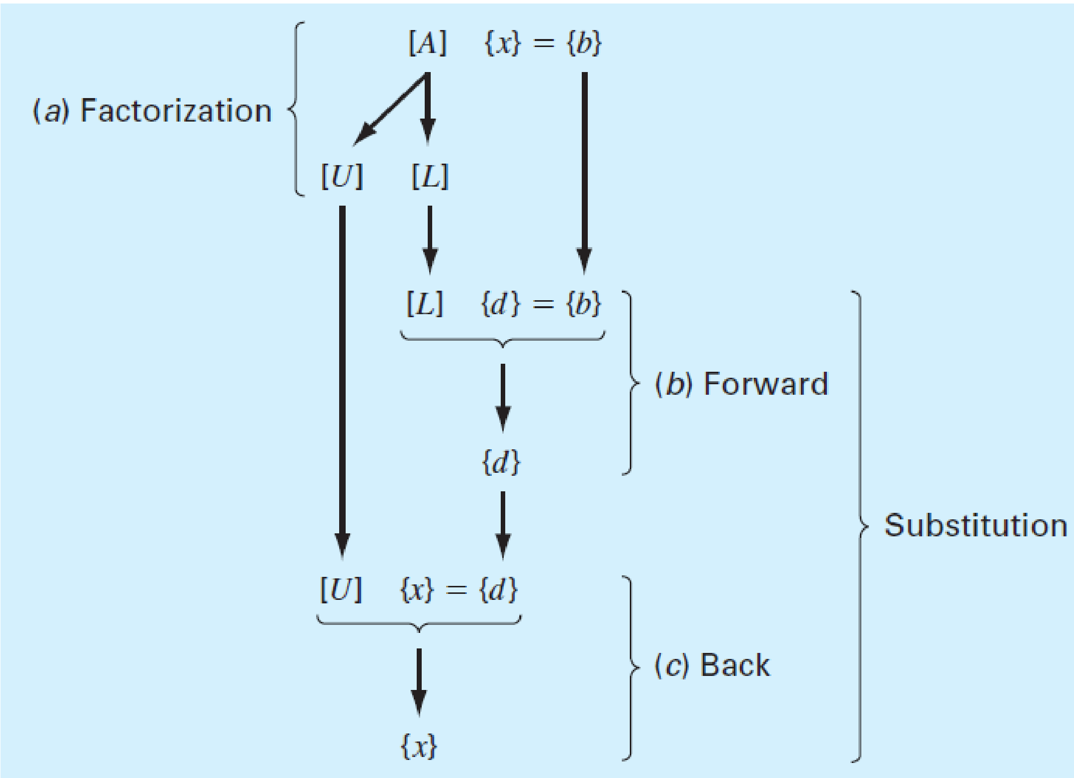
$$[L] = \begin{bmatrix} 1 & 0 & 0 \\ l_{21} & 1 & 0 \\ l_{31} & l_{32} & 1 \end{bmatrix}$$

$$[L]\{[U]\{x\} - \{d\}\} = [A]\{x\} - \{b\}$$

بنابراین باید داشته باشیم:

$$\left.\begin{aligned} &[L][U] = [A] \\ &[L]\{d\} = \{b\} \end{aligned}\right\}$$

مراحل حل دستگاه معادلات به روش تفکیک ماتریس ضرائب به ماتریسهای بالا و پائین مثلثی



a) تفکیک ماتریس ضرایب به دو ماتریس بالا و پائین مثلثی

b) استفاده از ماتریس پائین مثلثی برای محاسبه بردار d با استفاده از بردار b و روش جایگذاری مستقیم

c) استفاده از ماتریس بالا مثلثی برای محاسبه پاسخ با استفاده از بردار d و جایگذاری معکوس.

- مانند روش گاوس در این روش نیز می بایست از محورگیری (Pivoting) استفاده نمود تا از صفر شدن مخرج جلوگیری شود.

مثال:

برای ساخت ماتریس های بالا و پائین مثلثی از همان شیوه که در روش حذفی گاوس استفاده گردید می توان استفاده نمود.

$$[A] = \begin{bmatrix} 3 & -0.1 & -0.2 \\ 0.1 & 7 & -0.3 \\ 0.3 & -0.2 & 10 \end{bmatrix}$$

$$f_{21} = \frac{0.1}{3} = 0.0333333 \qquad f_{31} = \frac{0.3}{3} = 0.1000000$$

$$f_{32} = \frac{-0.19}{7.00333} = -0.0271300$$

$$[U] = \begin{bmatrix} 3 & -0.1 & -0.2 \\ 0 & 7.00333 & -0.293333 \\ 0 & 0 & 10.0120 \end{bmatrix} \qquad [L] = \begin{bmatrix} 1 & 0 & 0 \\ 0.0333333 & 1 & 0 \\ 0.100000 & -0.0271300 & 1 \end{bmatrix}$$

$$[A] = [L][U] = \begin{bmatrix} 1 & 0 & 0 \\ 0.0333333 & 1 & 0 \\ 0.100000 & -0.0271300 & 1 \end{bmatrix} \begin{bmatrix} 3 & -0.1 & -0.2 \\ 0 & 7.00333 & -0.293333 \\ 0 & 0 & 10.0120 \end{bmatrix}$$

$$[L][U] = \begin{bmatrix} 3 & -0.1 & -0.2 \\ 0.0999999 & 7 & -0.3 \\ 0.3 & -0.2 & 9.99996 \end{bmatrix}$$

روابط جایگذاری مستقیم و معکوس

- رابطه جایگذاری مستقیم برای محاسبه بردار d

$$d_i = b_i - \sum_{j=1}^{i-1} l_{ij} d_j \qquad \text{for } i = 1, 2, \ldots, n$$

- رابطه جایگذاری معکوس برای محاسبه بردارx (پاسخ)

$$x_n = d_n / u_{nn}$$

$$x_i = \frac{d_i - \sum\limits_{j=i+1}^{n} u_{ij} x_j}{u_{ii}} \qquad \text{for } i = n-1, n-2, \ldots, 1$$

- جایگذاری معکوس فوق دقیقا مشابه جایگذاری معکوس روش حذفی گاوس می باشد.

مثال:

- ماتریس نشان داده شده را به ماتریس های بالا و پائین مثلثی تفکیک نمائید.

$$[A] = \begin{bmatrix} 3 & -0.1 & -0.2 \\ 0.1 & 7 & -0.3 \\ 0.3 & -0.2 & 10 \end{bmatrix}$$

$$f_{21} = \frac{0.1}{3} = 0.0333333 \qquad f_{31} = \frac{0.3}{3} = 0.1000000 \qquad f_{32} = \frac{-0.19}{7.00333} = -0.0271300$$

$$[U] = \begin{bmatrix} 3 & -0.1 & -0.2 \\ 0 & 7.00333 & -0.293333 \\ 0 & 0 & 10.0120 \end{bmatrix} \qquad [L] = \begin{bmatrix} 1 & 0 & 0 \\ 0.0333333 & 1 & 0 \\ 0.100000 & -0.0271300 & 1 \end{bmatrix}$$

$$[A] = [L][U] = \begin{bmatrix} 1 & 0 & 0 \\ 0.0333333 & 1 & 0 \\ 0.100000 & -0.0271300 & 1 \end{bmatrix} \begin{bmatrix} 3 & -0.1 & -0.2 \\ 0 & 7.00333 & -0.293333 \\ 0 & 0 & 10.0120 \end{bmatrix}$$

$$[L][U] = \begin{bmatrix} 3 & -0.1 & -0.2 \\ 0.0999999 & 7 & -0.3 \\ 0.3 & -0.2 & 9.99996 \end{bmatrix}$$

حال دستگاه معادلات قبل با روش LU حل می شود:

$$\begin{bmatrix} 3 & -0.1 & -0.2 \\ 0.1 & 7 & -0.3 \\ 0.3 & -0.2 & 10 \end{bmatrix} \begin{Bmatrix} x_1 \\ x_2 \\ x_3 \end{Bmatrix} = \begin{Bmatrix} 7.85 \\ -19.3 \\ 71.4 \end{Bmatrix}$$

$$\begin{bmatrix} 1 & 0 & 0 \\ 0.0333333 & 1 & 0 \\ 0.100000 & -0.0271300 & 1 \end{bmatrix} \begin{Bmatrix} d_1 \\ d_2 \\ d_3 \end{Bmatrix} = \begin{Bmatrix} 7.85 \\ -19.3 \\ 71.4 \end{Bmatrix} \xrightarrow{\text{جایگذاری مستقیم}} \{d\} = \begin{Bmatrix} 7.85 \\ -19.5617 \\ 70.0843 \end{Bmatrix}$$

$$\begin{bmatrix} 3 & -0.1 & -0.2 \\ 0 & 7.00333 & -0.293333 \\ 0 & 0 & 10.0120 \end{bmatrix} \begin{Bmatrix} x_1 \\ x_2 \\ x_3 \end{Bmatrix} = \begin{Bmatrix} 7.85 \\ -19.5617 \\ 70.0843 \end{Bmatrix} \xrightarrow{\text{جایگذاری معکوس}} \{x\} = \begin{Bmatrix} 3 \\ -2.5 \\ 7.00003 \end{Bmatrix}$$

- مقایسه با روش حذفی گاوس:

$$\begin{bmatrix} 3 & -0.1 & -0.2 \\ 0 & 7.00333 & -0.293333 \\ 0 & 0 & 10.0120 \end{bmatrix} \begin{Bmatrix} x_1 \\ x_2 \\ x_3 \end{Bmatrix} = \begin{Bmatrix} 7.85 \\ -19.5617 \\ 70.0843 \end{Bmatrix}$$

10.2.2 MATLAB Function: lu

تفکیک LU در متلب

```
[L,U] = lu(X)
```

- فرم دستور LU در متلب:
- بطور مثال دستگاه قبل با استفاده از متلب حل می شود.

$$\begin{bmatrix} 3 & -0.1 & -0.2 \\ 0.1 & 7 & -0.3 \\ 0.3 & -0.2 & 10 \end{bmatrix} \begin{Bmatrix} x_1 \\ x_2 \\ x_3 \end{Bmatrix} = \begin{Bmatrix} 7.85 \\ -19.3 \\ 71.4 \end{Bmatrix}$$

```
>> A = [3 -.1 -.2;.1 7 -.3;.3 -.2 10];
>> b = [7.85; -19.3; 71.4];

>> [L,U] = lu(A)

L =
    1.0000         0         0
    0.0333    1.0000         0
    0.1000   -0.0271    1.0000

U =
    3.0000   -0.1000   -0.2000
         0    7.0033   -0.2933
         0         0   10.0120
```

```
>> d = L\b

d =
    7.8500
  -19.5617
   70.0843

>> x = U\d

x =

    3.0000
   -2.5000
    7.0000
```

10.2.1 *LU* Factorization with Pivoting — تفکیک LU با محورگیری

- در روش تفکیک LU برای رسیدن به نتایج صحیح لازم است مشابه روش حذفی گاوس، از محورگیری جزیی استفاده شود.
- بدین منظور از یک ماتریس تبدیل یا جابجایی (permutation) استفاده می شود. این ماتریس پیش از تفکیک در ماتریس ضرایب پیش ضرب می شود و داریم:

$$[P][A] = [L][U]$$

- روند تفکیک با محورگیری را می توان به صورت زیر نشان داد:

۱– مرحله حذف: ماتریس بالا مثلثی U با حذف و محورگیری از ماتریس ضرایب حاصل می شود. ضرایب حذف کننده نیز در ماتریس L ذخیره می شوند. همچنین ماتریس تبدیل P ذخیره می شود تا نحوه جابجایی سطرها حفظ گردد.

۲- جایگذاری مستقیم: در این مرحله از ماتریس پائین مثلثی L و ماتریس انتقال P برای محاسبه بردار واسط d با روش جایگذاری مستقیم و محورگیری استفاده می شود.

$$[L]\{d\} = [P]\{b\}$$

۳- جایگذاری معکوس: در این مرحله از ماتریس بالا مثلثی U و بردار واسط d برای محاسبه بردار پاسخ x استفاده می شود. این مرحله دقیقا مشابه مرحله شوم در روش تفکیک LU بدون محورگیری است.

$$[U]\{x\} = \{d\}$$

مثال:

دستگاه معادلات زیر را با روش تفکیک **LU** با محورگیری حل می کنیم:

$$\begin{bmatrix} 0.0003 & 3.0000 \\ 1.0000 & 1.0000 \end{bmatrix} \begin{Bmatrix} x_1 \\ x_2 \end{Bmatrix} = \begin{Bmatrix} 2.0001 \\ 1.0000 \end{Bmatrix}$$

با توجه به مقادیر روی قطر اصلی، لازم است سطرهای اول و دوم جابجا شوند:

$$[A] = \begin{bmatrix} 1.0000 & 1.0000 \\ 0.0003 & 3.0000 \end{bmatrix}$$

به منظور حفظ نحوه جابجایی سطرها ماتریس جابجایی زیر ذخیره می شود:

$$[P] = \begin{bmatrix} 0.0000 & 1.0000 \\ 1.0000 & 0.0000 \end{bmatrix}$$

با استفاده از ماتریس ضرایب جدید ماتریس های بالا و پائین مثلثی بصورت زیر حاصل می شوند:

$$[U] = \begin{bmatrix} 1 & 1 \\ 0 & 2.9997 \end{bmatrix} \qquad [L] = \begin{bmatrix} 1 & 0 \\ 0.0003 & 1 \end{bmatrix}$$

از آنجا که سطرهای ماتریس ضرایب جابجا شده، از ماتریس تبدیل P برای جابجایی سطرهای بردار مقادیر ثابت استفاده می شود:

$$[P]\{b\} = \begin{bmatrix} 0.0000 & 1.0000 \\ 1.0000 & 0.0000 \end{bmatrix} \begin{Bmatrix} 2.0001 \\ 1 \end{Bmatrix} = \begin{Bmatrix} 1 \\ 2.0001 \end{Bmatrix}$$

بردار واسط d محاسبه می شود:

$$\begin{bmatrix} 1 & 0 \\ 0.0003 & 1 \end{bmatrix} \begin{Bmatrix} d_1 \\ d_2 \end{Bmatrix} = \begin{Bmatrix} 1 \\ 2.0001 \end{Bmatrix}$$

با استفاده از بردار واسط d، بردار مجهولات بدست می آید:

$$\begin{bmatrix} 1 & 1 \\ 0 & 2.9997 \end{bmatrix} \begin{Bmatrix} x_1 \\ x_2 \end{Bmatrix} = \begin{Bmatrix} 1 \\ 1.9998 \end{Bmatrix}$$

و با جایگذاری معکوس داریم:

$$x_2 = \frac{1.9998}{2.9997} = 0.66667$$

$$x_1 = \frac{1 - 1(0.66667)}{1} = 0.33333$$

- بسیاری از سیستمها دارای ماتریس متقارن می باشند. برای این ماتریسها روشهای حل ویژه ای وجود دارد. تقارن ماتریس باعث کاهش فضای ذخیره سازی و زمان حل به نصف می شود.

$$[A] = [A]^T$$

- یکی از مهمترین روشها برای ماتریسهای متقارن تفکیک چولسکی است. یک ماتریس مقارن را می توان به صورت زیر تفکیک کرد:

$$[A] = [U]^T[U]$$

- یعنی دو ماتریس بالا و پائین مثلثی ترانسپوز یکدیگرند.

$$u_{ii} = \sqrt{a_{ii} - \sum_{k=1}^{i-1} u_{ki}^2}$$

$$u_{ij} = \frac{a_{ij} - \sum_{k=1}^{i-1} u_{ki} u_{kj}}{u_{ii}} \qquad \text{for } j = i+1, \ldots, n$$

مثال: ماتریس نشان داده شده را به روش چولسکی تفکیک نمائید.

$$[A] = \begin{bmatrix} 6 & 15 & 55 \\ 15 & 55 & 225 \\ 55 & 225 & 979 \end{bmatrix}$$

$$u_{11} = \sqrt{a_{11}} = \sqrt{6} = 2.44949$$

$$u_{12} = \frac{a_{12}}{u_{11}} = \frac{15}{2.44949} = 6.123724$$

$$u_{13} = \frac{a_{13}}{u_{11}} = \frac{55}{2.44949} = 22.45366$$

$$u_{22} = \sqrt{a_{22} - u_{12}^2} = \sqrt{55 - (6.123724)^2} = 4.1833$$

$$u_{23} = \frac{a_{23} - u_{12}u_{13}}{u_{22}} = \frac{225 - 6.123724(22.45366)}{4.1833} = 20.9165$$

$$u_{33} = \sqrt{a_{33} - u_{13}^2 - u_{23}^2} = \sqrt{979 - (22.45366)^2 - (20.9165)^2} = 6.110101$$

$$[U] = \begin{bmatrix} 2.44949 & 6.123724 & 22.45366 \\ & 4.1833 & 20.9165 \\ & & 6.110101 \end{bmatrix}$$

10.3.1 MATLAB Function: chol — تفکیک چولسکی در متلب:

- به کمک این دستور دستگاه زیر را حل می نمائیم: `U = chol(X)`

$$[A] = \begin{bmatrix} 6 & 15 & 55 \\ 15 & 55 & 225 \\ 55 & 225 & 979 \end{bmatrix}$$

```
b =
    76
   295
  1259
```

```
>> U = chol(A)

U =
    2.4495    6.1237   22.4537
         0    4.1833   20.9165
         0         0    6.1101

>> U'*U

ans =
    6.0000   15.0000   55.0000
   15.0000   55.0000  225.0000
   55.0000  225.0000  979.0000
```

```
>> d = U'\b

d =
   31.0269
   25.0998
    6.1101

>> x = U\d

x =
    1.0000
    1.0000
    1.0000
```

تمرین ۱:

- دستگاه نشان داده شده را با استفاده از روش تجزیه به ماتریس های بالا و پائین مثلثی حل نمائید.

$$
\begin{aligned}
2x_1 - 6x_2 - x_3 &= -38 \\
-3x_1 - x_2 + 6x_3 &= -34 \\
-8x_1 + x_2 - 2x_3 &= -40
\end{aligned}
$$

تمرین ۲:

- دستگاه معادلات زیر را به روش چولسکی حل نمائید.

$$\begin{bmatrix} 8 & 20 & 16 \\ 20 & 80 & 50 \\ 16 & 50 & 60 \end{bmatrix} \begin{Bmatrix} x_1 \\ x_2 \\ x_3 \end{Bmatrix} = \begin{Bmatrix} 100 \\ 250 \\ 100 \end{Bmatrix}$$

- نتایج خود را با نتایج حاصل از تابع متلب مقایسه نمائید.

حل:

$$\begin{bmatrix} 8 & 20 & 16 \\ 20 & 80 & 50 \\ 16 & 50 & 60 \end{bmatrix} \begin{Bmatrix} x_1 \\ x_2 \\ x_3 \end{Bmatrix} = \begin{Bmatrix} 100 \\ 250 \\ 100 \end{Bmatrix}$$

```
clc
clear
a=[۸ ۲۰ ۱۶;۲۰ ۸۰ ۵۰;۱۶ ۵۰ ۶۰]
u=zeros(length(a),length(a));
n=length(a);
for i=۱:n
   sum۱=۰;
   sum۲=۰;
   for k=۱:i-۱
      sum۱=sum۱+u(k,i)^۲;
   end
   u(i,i)=(a(i,i)-sum۱)^۰.۵;
   for j=i+۱:n
      for k=۱:i-۱
         sum۲=sum۲+u(k,i)*u(k,j);
      end
      u(i,j)=(a(i,j)-sum۲)/u(i,i);
   end
end
u
```

$$u_{ii} = \sqrt{a_{ii} - \sum_{k=1}^{i-1} u_{ki}^2}$$

$$u_{ij} = \frac{a_{ij} - \sum_{k=1}^{i-1} u_{ki} u_{kj}}{u_{ii}} \qquad \text{for } j = i+1, \ldots, n$$

```
u =

   ۲.۸۲۸۴   ۷.۰۷۱۱   ۵.۶۵۶۹
        ۰   ۵.۴۷۷۲   ۱.۸۲۵۷
        ۰        ۰   ۴.۹۶۶۶

>> chol(a)

ans =

   ۲.۸۲۸۴   ۷.۰۷۱۱   ۵.۶۵۶۹
        ۰   ۵.۴۷۷۲   ۱.۸۲۵۷
        ۰        ۰   ۴.۹۶۶۶
```

11

فصل یازدهم

Matrix Inverse and Condition

معکوس ماتریس و حالت ماتریس

11.1 THE MATRIX INVERSE ماتریس معکوس با استفاده از تفکیک LU

- ماتریس معکوس را می توان به صورت ستون به ستون محاسبه نمود. در این روش دستگاه معادلاتی در نظر گرفته می شود که سمت راست آن بردار مقادیر ثابت شامل یک عدد ۱ و بقیه مقادیر برابر صفر است. بردار نتایج ستونی از ماتریس معکوس است. بطور مثال اگر بردار مقادیر ثابت دارای ۱ در سطر اول باشد بردار بدست آمده اولین ستون ماتریس معکوس است.

$$\{b\} = \begin{Bmatrix} 1 \\ 0 \\ 0 \end{Bmatrix}$$

برداری مورد استفاده برای محاسبه ستون اول ماتریس معکوس

$$\{b\} = \begin{Bmatrix} 0 \\ 1 \\ 0 \end{Bmatrix}$$

برداری مورد استفاده برای محاسبه ستون دوم ماتریس معکوس

- بهترین روش برای محاسبه ماتریس معکوس با این شیوه روش تفکیک LU است. زیرا همانگونه که اشاره گردید در این روش با تغییر بردار مقادیر ثابت نیازی به محاسبه مجدد L و U نیست.

$$[A][A]^{-1} = [A]^{-1}[A] = [I]$$

مثال:

- معکوس ماتریس زیر با روش تفکیک LU محاسبه می گردد:

$$[A] = \begin{bmatrix} 3 & -0.1 & -0.2 \\ 0.1 & 7 & -0.3 \\ 0.3 & -0.2 & 10 \end{bmatrix}$$

حل: ابتدا ماتریس های بالا و پائین مثلثی مربوط به ماتریس فوق محاسبه می شود:

$$[U] = \begin{bmatrix} 3 & -0.1 & -0.2 \\ 0 & 7.00333 & -0.293333 \\ 0 & 0 & 10.0120 \end{bmatrix} \qquad [L] = \begin{bmatrix} 1 & 0 & 0 \\ 0.0333333 & 1 & 0 \\ 0.100000 & -0.0271300 & 1 \end{bmatrix}$$

حال بردار مقادیر ثابت را بصورت زیر در نظر می گیریم تا ستون اول ماتریس معکوس بدست آید:

$$\begin{bmatrix} 1 & 0 & 0 \\ 0.0333333 & 1 & 0 \\ 0.100000 & -0.0271300 & 1 \end{bmatrix} \begin{Bmatrix} d_1 \\ d_2 \\ d_3 \end{Bmatrix} = \begin{Bmatrix} 1 \\ 0 \\ 0 \end{Bmatrix}$$

$$\{d\}^T = \lfloor 1 \quad -0.03333 \quad -0.1009 \rfloor$$

$$\begin{bmatrix} 3 & -0.1 & -0.2 \\ 0 & 7.00333 & -0.293333 \\ 0 & 0 & 10.0120 \end{bmatrix} \begin{Bmatrix} x_1 \\ x_2 \\ x_3 \end{Bmatrix} = \begin{Bmatrix} 1 \\ -0.03333 \\ -0.1009 \end{Bmatrix}$$

$$\{x\}^T = \lfloor 0.33249 \quad -0.00518 \quad -0.01008 \rfloor$$

$$[A]^{-1} = \begin{bmatrix} 0.33249 & 0 & 0 \\ -0.00518 & 0 & 0 \\ -0.01008 & 0 & 0 \end{bmatrix}$$

- درصورتیکه از بردار زیر به عنوان مقادیر ثابت استفاده شود داریم:

$$\{b\} = \begin{Bmatrix} 0 \\ 1 \\ 0 \end{Bmatrix} \Rightarrow \begin{bmatrix} 1 & 0 & 0 \\ 0.0333333 & 1 & 0 \\ 0.100000 & -0.0271300 & 1 \end{bmatrix} \begin{Bmatrix} d_1 \\ d_2 \\ d_3 \end{Bmatrix} = \begin{Bmatrix} 0 \\ 1 \\ 0 \end{Bmatrix}$$

$$\{x\}^T = \lfloor 0.004944 \quad 0.142903 \quad 0.00271 \rfloor$$

- بنابراین ستون دوم ماتریس معکوس به صورت زیر خواهد بود:

$$[A]^{-1} = \begin{bmatrix} 0.33249 & 0.004944 & 0 \\ -0.00518 & 0.142903 & 0 \\ -0.01008 & 0.002710 & 0 \end{bmatrix}$$

- و برای ستون سوم:

$$\{b\}^T = \lfloor 0 \quad 0 \quad 1 \rfloor \Rightarrow \{x\}^T = \lfloor 0.006798 \quad 0.004183 \quad 0.09988 \rfloor$$

$$[A]^{-1} = \begin{bmatrix} 0.33249 & 0.004944 & 0.006798 \\ -0.00518 & 0.142903 & 0.004183 \\ -0.01008 & 0.002710 & 0.099880 \end{bmatrix}$$

- و برای چک کردن نتیجه حاصل:

$$[A][A]^{-1} = [I].$$

برای بررسی حالت سیستم معادلات می توان از معکوس ماتریس استفاده نمود و سه روش برای این کار وجود دارد:

۱- ماتریس را به نحوی مقیاس بندی می کنیم که بزرگترین عدد در هر ردیف ۱ باشد. سپس ماتریس معکوس محاسبه می شود و اگر مرتبه درایه های ماتریس معکوس بسیار بزرگتر از ۱ باشد این سیستم مریض می باشد.

۲- پس از آنکه معکوس ماتریس در خودش ضرب شود در صورتیکه حاصل از ماتریس همانی خیلی اختلاف داشته باشد این سیستم مریض است.

۳- مجددا معکوس ماتریس معکوس محاسبه می شود و درصورتیکه با ماتریس اولیه اختلاف زیادی داشته باشد این سیستم مریض است.

در نرم افزار متلب می توان از دستور Cond برای محاسبه حالت ماتریس استفاده نمود. هرچه عدد حالت ماتریس از ۱ بیشتر اختلاف داشته باشد ماتریس مریض تر است. فرم دستور: `cond(A)`

مثال:

ماتریس زیر را در نظر بگیرید:

$$[A] = \begin{bmatrix} 1 & \frac{1}{2} & \frac{1}{3} \\ 1 & \frac{2}{3} & \frac{1}{2} \\ 1 & \frac{3}{4} & \frac{3}{5} \end{bmatrix}$$

برای محاسبه عدد حالت از دستور cond استفاده می شود:

```
>> cond(A)

ans =
  366.3503
```

با توجه به نتیجه حاصل ماتریس مریض است.

12 فصل دوازدهم

Iterative Methods

روشهای تکراری

- این روش معروف ترین روش تکراری برای حل سیستم های خطی است. فرض کنید یک دستگاه سه معادله – سه مجهول می بایست حل گردد. در اینصورت از هر معادله مجهول روی قطر آن استخراج می شود:

$$x_1^j = \frac{b_1 - a_{12}x_2^{j-1} - a_{13}x_3^{j-1}}{a_{11}}$$

$$x_2^j = \frac{b_2 - a_{21}x_1^j - a_{23}x_3^{j-1}}{a_{22}}$$

$$x_3^j = \frac{b_3 - a_{31}x_1^j - a_{32}x_2^j}{a_{33}}$$

- رابطه محاسبه درصد خطای نسبی تقریبی:

$$\varepsilon_{a,i} = \left| \frac{x_i^j - x_i^{j-1}}{x_i^j} \right| \times 100\% \le \varepsilon_s$$

- رابطه فوق می تواند به عنوان شرط توقف در برنامه مورد استفاده قرار گیرد.

مثال: حل دستگاه نشان داده شده به روش گاوس-سایدل

$$
\begin{aligned}
3x_1 - 0.1x_2 - 0.2x_3 &= 7.85 \\
0.1x_1 + 7x_2 - 0.3x_3 &= -19.3 \\
0.3x_1 - 0.2x_2 + 10x_3 &= 71.4
\end{aligned}
$$

the solution is $x_1 = 3$, $x_2 = -2.5$, and $x_3 = 7$.

$$x_1 = \frac{7.85 + 0.1x_2 + 0.2x_3}{3}$$

$$x_2 = \frac{-19.3 - 0.1x_1 + 0.3x_3}{7}$$

$$x_3 = \frac{71.4 - 0.3x_1 + 0.2x_2}{10}$$

By assuming that x_2 and x_3 are zero,

$$x_1 = \frac{7.85 + 0.1(-2.794524) + 0.2(7.005610)}{3} = 2.990557$$

$$x_2 = \frac{-19.3 - 0.1(2.990557) + 0.3(7.005610)}{7} = -2.499625$$

$$x_3 = \frac{71.4 - 0.3(2.990557) + 0.2(-2.499625)}{10} = 7.000291$$

شرط همگرایی روش گاوس-سایدل

می توان ثابت کرد که درصورتیکه ماتریس ضرایب غالب قطری(Diagonally Dominant) باشد روش گاوس-سایدل به پاسخ صحیح همگرا خواهد شد. یعنی شرط همگرایی بصورت زیر می باشد:

$$|a_{ii}| > \sum_{\substack{j=1 \\ j \neq i}}^{n} |a_{ij}|$$

روش تکراری ژاکوبی در مقایسه با روش گاوس- سایدل

First iteration

$$x_1 = (b_1 - a_{12}x_2 - a_{13}x_3)/a_{11}$$

$$x_2 = (b_2 - a_{21}x_1 - a_{23}x_3)/a_{22}$$

$$x_3 = (b_3 - a_{31}x_1 - a_{32}x_2)/a_{33}$$

Second iteration

$$x_1 = (b_1 - a_{12}x_2 - a_{13}x_3)/a_{11}$$

$$x_2 = (b_2 - a_{21}x_1 - a_{23}x_3)/a_{22}$$

$$x_3 = (b_3 - a_{31}x_1 - a_{32}x_2)/a_{33}$$

(*a*)

First iteration

$$x_1 = (b_1 - a_{12}x_2 - a_{13}x_3)/a_{11}$$

$$x_2 = (b_2 - a_{21}x_1 - a_{23}x_3)/a_{22}$$

$$x_3 = (b_3 - a_{31}x_1 - a_{32}x_2)/a_{33}$$

Second iteration

$$x_1 = (b_1 - a_{12}x_2 - a_{13}x_3)/a_{11}$$

$$x_2 = (b_2 - a_{21}x_1 - a_{23}x_3)/a_{22}$$

$$x_3 = (b_3 - a_{31}x_1 - a_{32}x_2)/a_{33}$$

(*b*)

در روش گاوس- سایدل مقادیر بدست آمده در هر مرحله بلافاصله در معادله بعدی جایگذاری می شود.

در روش ژاکوبی مقادیر بدست آمده در هر مرحله بلافاصله در معادلات بعدی جایگذاری نمی شوند بلکه در مرحله بعد بصورت همزمان استفاده می گردند.

(a)روش گاوس سایدل

(b)روش ژاکوبی

12.1.3 Relaxation

- روش گاوس- سایدل مشابه روش تکراری یک نقطه ثابت برای حل معادلات است که گاهی اوقات واگرا می شود. به همین دلیل روش Relaxation استفاده می شود.
- این روش تغییر یافته روش گاوس- سایدل است تا همگرایی آن را بهبود بخشد.
- در این روش از مجموع وزنی مقادیر جدید و قدیم برای جایگذاری در معادلات بعدی استفاده می شود.

$$x_i^{\text{new}} = \lambda x_i^{\text{new}} + (1-\lambda)x_i^{\text{old}}$$

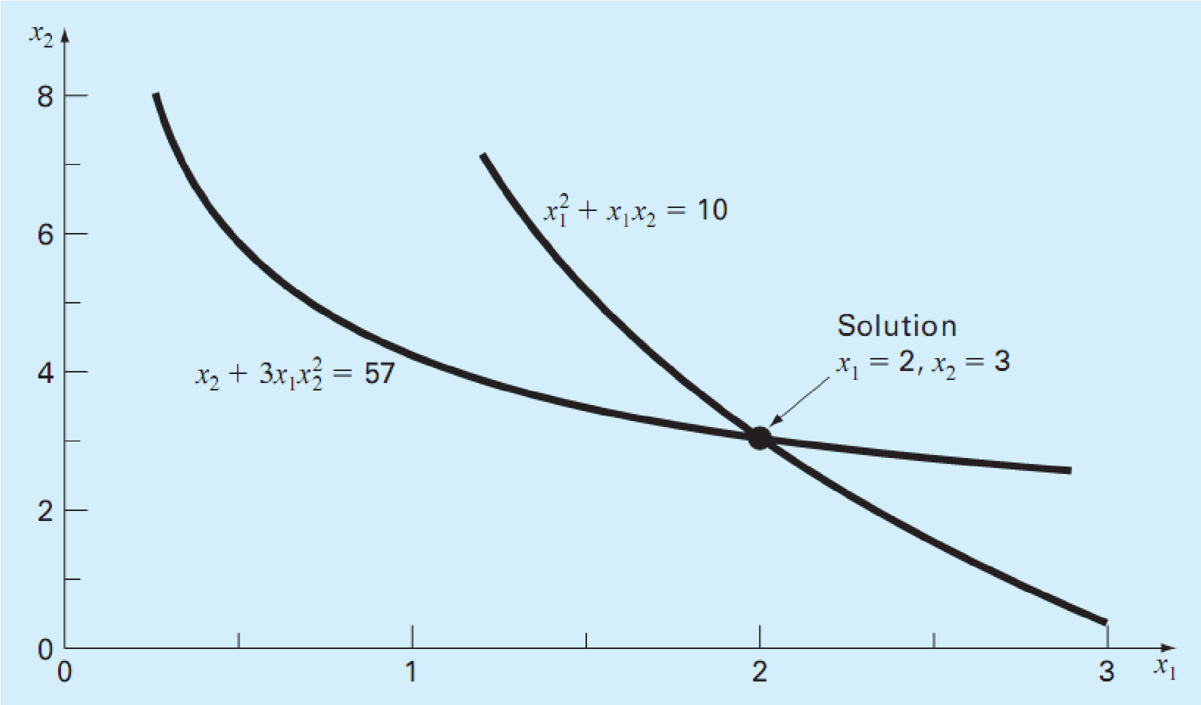
- در صورتیکه λ عددی بین ۰ و ۱ باشد روش Underrelaxation نام دارد و برای همگرا نمودن سیستم های واگرا استفاده می شود.
- در صورتیکه λ عددی بین ۱ و ۲ باشد روش Overrelaxation نام دارد و برای سریعتر نمودن همگرایی سیستم های همگرا استفاده می شود.

- یک سیستم معادلات غیر خطی را می توان به شکل زیر نشان داد.

$$
\begin{aligned}
f_1(x_1, x_2, \ldots, x_n) &= 0 \\
f_2(x_1, x_2, \ldots, x_n) &= 0 \\
&\vdots \\
f_n(x_1, x_2, \ldots, x_n) &= 0
\end{aligned}
$$

- بطور مثال دستگاه نشان داده شده دارای حل زیر است:

$$
\begin{aligned}
x_1^2 + x_1 x_2 &= 10 \\
x_2 + 3x_1 x_2^2 &= 57
\end{aligned}
$$



12.2.1 Successive Substitution جایگذاری مکرر

- در این روش هر معادله را بر حسب یکی از متغیرها ضمنی کرده و با شروع از یک مقدار اولیه فرضی برای متغیرها، مقادیر جدید بدست می آید. مقادیر جدید مجددا در معادلات ضمنی جایگذاری می شوند.
- با ادامه این روند ممکن است مقادیر حاصل همگرا یا واگرا گردند که همگرایی یا واگرایی به نوع ضمنی کردن معادلات بستگی دارد.
- حتی در مواردی که سیستم می تواند همگرا شود، با انتخاب مقادیر اولیه دور از هدف ممکن است واگرایی رخ دهد.

مثال:

$$x_1^2 + x_1x_2 = 10$$
$$x_2 + 3x_1x_2^2 = 57$$

correct pair of roots is $x_1 = 2$ and $x_2 = 3$.

$$x_1^2 + x_1x_2 = 10$$
$$x_2 + 3x_1x_2^2 = 57$$

$$\Rightarrow \begin{cases} x_1 = \dfrac{10 - x_1^2}{x_2} \\ x_2 = 57 - 3x_1x_2^2 \end{cases}$$

$$x_1 = \frac{10 - (1.5)^2}{3.5} = 2.21429$$
$$x_2 = 57 - 3(2.21429)(3.5)^2 = -24.37516$$

$$x_1 = \frac{10 - (2.21429)^2}{-24.37516} = -0.20910$$
$$x_2 = 57 - 3(-0.20910)(-24.37516)^2 = 429.709$$

واگرا

مثال:

$$x_1^2 + x_1x_2 = 10$$
$$x_2 + 3x_1x_2^2 = 57$$

correct pair of roots is $x_1 = 2$ and $x_2 = 3$.

$$x_1^2 + x_1x_2 = 10$$
$$x_2 + 3x_1x_2^2 = 57$$

$$\Rightarrow \begin{cases} x_1 = \sqrt{10 - x_1x_2} \\ x_2 = \sqrt{\dfrac{57 - x_2}{3x_1}} \end{cases}$$

$$x_1 = \sqrt{10 - 1.5(3.5)} = 2.17945$$

$$x_2 = \sqrt{\frac{57 - 3.5}{3(2.17945)}} = 2.86051$$

$$x_1 = \sqrt{10 - 2.17945(2.86051)} = 1.94053$$

$$x_2 = \sqrt{\frac{57 - 2.86051}{3(1.94053)}} = 3.04955$$

همگرا

12.2.2 Newton-Raphson روش نیوتن-رافسون

$$f(x_{i+1}) = f(x_i) + (x_{i+1} - x_i) f'(x_i) \xrightarrow{f(x_{i+1}) = 0} x_{i+1} = x_i - \frac{f(x_i)}{f'(x_i)}$$

بسط تیلور

توابع دومتغیره

$$\left\{\begin{aligned} f_{1,i+1} &= f_{1,i} + (x_{1,i+1} - x_{1,i}) \frac{\partial f_{1,i}}{\partial x_1} + (x_{2,i+1} - x_{2,i}) \frac{\partial f_{1,i}}{\partial x_2} \\ f_{2,i+1} &= f_{2,i} + (x_{1,i+1} - x_{1,i}) \frac{\partial f_{2,i}}{\partial x_1} + (x_{2,i+1} - x_{2,i}) \frac{\partial f_{2,i}}{\partial x_2} \end{aligned}\right\} \xrightarrow{\substack{f_{1,i+1} = 0 \\ f_{2,i+1} = 0}}$$

$$\left\{\begin{aligned} \frac{\partial f_{1,i}}{\partial x_1} x_{1,i+1} + \frac{\partial f_{1,i}}{\partial x_2} x_{2,i+1} &= -f_{1,i} + x_{1,i} \frac{\partial f_{1,i}}{\partial x_1} + x_{2,i} \frac{\partial f_{1,i}}{\partial x_2} \\ \frac{\partial f_{2,i}}{\partial x_1} x_{1,i+1} + \frac{\partial f_{2,i}}{\partial x_2} x_{2,i+1} &= -f_{2,i} + x_{1,i} \frac{\partial f_{2,i}}{\partial x_1} + x_{2,i} \frac{\partial f_{2,i}}{\partial x_2} \end{aligned}\right.$$

$$x_{1,i+1} = x_{1,i} - \frac{f_{1,i} \dfrac{\partial f_{2,i}}{\partial x_2} - f_{2,i} \dfrac{\partial f_{1,i}}{\partial x_2}}{\boxed{\dfrac{\partial f_{1,i}}{\partial x_1} \dfrac{\partial f_{2,i}}{\partial x_2} - \dfrac{\partial f_{1,i}}{\partial x_2} \dfrac{\partial f_{2,i}}{\partial x_1}}}$$

$$x_{2,i+1} = x_{2,i} - \frac{f_{2,i} \dfrac{\partial f_{1,i}}{\partial x_1} - f_{1,i} \dfrac{\partial f_{2,i}}{\partial x_1}}{\boxed{\dfrac{\partial f_{1,i}}{\partial x_1} \dfrac{\partial f_{2,i}}{\partial x_2} - \dfrac{\partial f_{1,i}}{\partial x_2} \dfrac{\partial f_{2,i}}{\partial x_1}}}$$

دترمینان ماتریس ژاکوبین

فرم ماتریسی روش نیوتن-رافسون برای سیستم های غیرخطی

$$[J]\{x_{i+1}\} = -\{f\} + [J]\{x_i\}$$

$$[J] = \begin{bmatrix} \frac{\partial f_{1,i}}{\partial x_1} & \frac{\partial f_{1,i}}{\partial x_2} & \cdots & \frac{\partial f_{1,i}}{\partial x_n} \\ \frac{\partial f_{2,i}}{\partial x_1} & \frac{\partial f_{2,i}}{\partial x_2} & \cdots & \frac{\partial f_{2,i}}{\partial x_n} \\ \vdots & \vdots & & \vdots \\ \frac{\partial f_{n,i}}{\partial x_1} & \frac{\partial f_{n,i}}{\partial x_2} & \cdots & \frac{\partial f_{n,i}}{\partial x_n} \end{bmatrix}$$

$$\{x_{i+1}\}^T = \lfloor x_{1,i+1} \quad x_{2,i+1} \quad \cdots \quad x_{n,i+1} \rfloor$$

$$\{x_i\}^T = \lfloor x_{1,i} \quad x_{2,i} \quad \cdots \quad x_{n,i} \rfloor$$

$$\{f\}^T = \lfloor f_{1,i} \quad f_{2,i} \quad \cdots \quad f_{n,i} \rfloor$$

$$x_{i+1} = x_i - \frac{f(x_i)}{f'(x_i)}$$

$$\{x_{i+1}\} = \{x_i\} - [J]^{-1}\{f\}$$

تشابه روش نیوتن-رافسون برای
یک معادله و سیستم معادلات

عیوب روش نیوتن-رافسون

۱- در برخی موارد محاسبه ماتریس ژاکوبین برای سیستم های بزرگ مشکل است. لذا بجای استفاده از مشتق گیری، روش تفاضل محدود برای محاسبه ماتریس ژاکوبین استفاده می شود.

۲- مشکل دیگر به انتخاب نقاط شروع اولیه برمیگردد. در واقع اگر نقاط شروع به اندازه کافی به هدف نزدیک نباشند این روش واگرا می شود.

تمرین ۱:

دستگاه معادلات نشان داده شده را با روش گوس- سایدل

الف) بدون Relaxation

ب) با Relaxation $(\lambda = 1.2)$

حل نمائید.

$$\begin{aligned} 2x_1 - 6x_2 - x_3 &= -38 \\ -3x_1 - x_2 + 7x_3 &= -34 \\ -8x_1 + x_2 - 2x_3 &= -20 \end{aligned}$$

تمرین۲:

- دستگاه معادلات زیر را با روش
- الف) ترسیمی
- ب) جایگذاری مکرر
- حل نمائید. برای حدس اولیه می توانید از مقادیر داده شده استفاده نمائید.

$$x^2 = 5 - y^2$$
$$y + 1 = x^2$$

$$x = y = 1.5.$$

تمرین ۳:

- حل دستگاه معادلات غیرخطی زیر را با روش نیوتن-رافسون و با استفاده از نقاط شروع داده شده بیابید.

$$y = -x^2 + x + 0.5$$
$$y + 5xy = x^2$$

$$x = y = 1.2.$$

13 فصل سیزدهم

Eigenvalues

مقادیر ویژه

$$[[A] - \lambda[I]]\{x\} = 0$$

- برای داشتن حل غیر بدیهی:

$$|[A] - \lambda[I]| = 0$$

$$(a_{11} - \lambda)x_1 + \qquad a_{12}x_2 = 0$$

$$a_{21}x_1 + (a_{22} - \lambda)x_2 = 0$$

- ریشه های چندجمله ای مشخصه مقادیر ویژه سیستم می باشند:

$$\begin{vmatrix} a_{11} - \lambda & a_{12} \\ a_{21} & a_{22} - \lambda \end{vmatrix} = \underbrace{\lambda^2 - (a_{11} + a_{22})\lambda - a_{12}a_{21}}_{\textit{characteristic polynomial.}}$$

$$\begin{matrix} \lambda_1 \\ \lambda_2 \end{matrix} = \frac{(a_{11} - a_{22})^2 \pm \sqrt{(a_{11} - a_{22})^2 - 4a_{12}a_{21}}}{2}$$

- به این روش محاسبه مقادیر ویژه روش چندجمله ای گویند زیرا از چندجمله ای مشخصه استفاده می شود.

مثال:

تعیین مقادیر ویژه از روش چندجمله ای شاخص:

$$\begin{aligned}(10-\lambda)x_1 \qquad\qquad - 5x_2 &= 0\\ -5x_1 + (10-\lambda)x_2 &= 0\end{aligned} \qquad \begin{vmatrix} 10-\lambda & -5 \\ -5 & 10-\lambda \end{vmatrix} = \lambda^2 - 20\lambda + 75$$

$$\begin{matrix}\lambda_1\\ \lambda_2\end{matrix} = \frac{20 \pm \sqrt{20^2 - 4(1)75}}{2} = 15, 5$$

در مقادیر ویژه بینهایت جواب خواهد داشت که نسبت آنها ثابت است.

for $\lambda_1 = 15,$

$$\begin{aligned}-5x_1 - 5x_2 &= 0\\ -5x_1 - 5x_2 &= 0\end{aligned} \qquad \{x\} = \begin{Bmatrix} -1 \\ 1 \end{Bmatrix}$$

یک جواب ممکن:

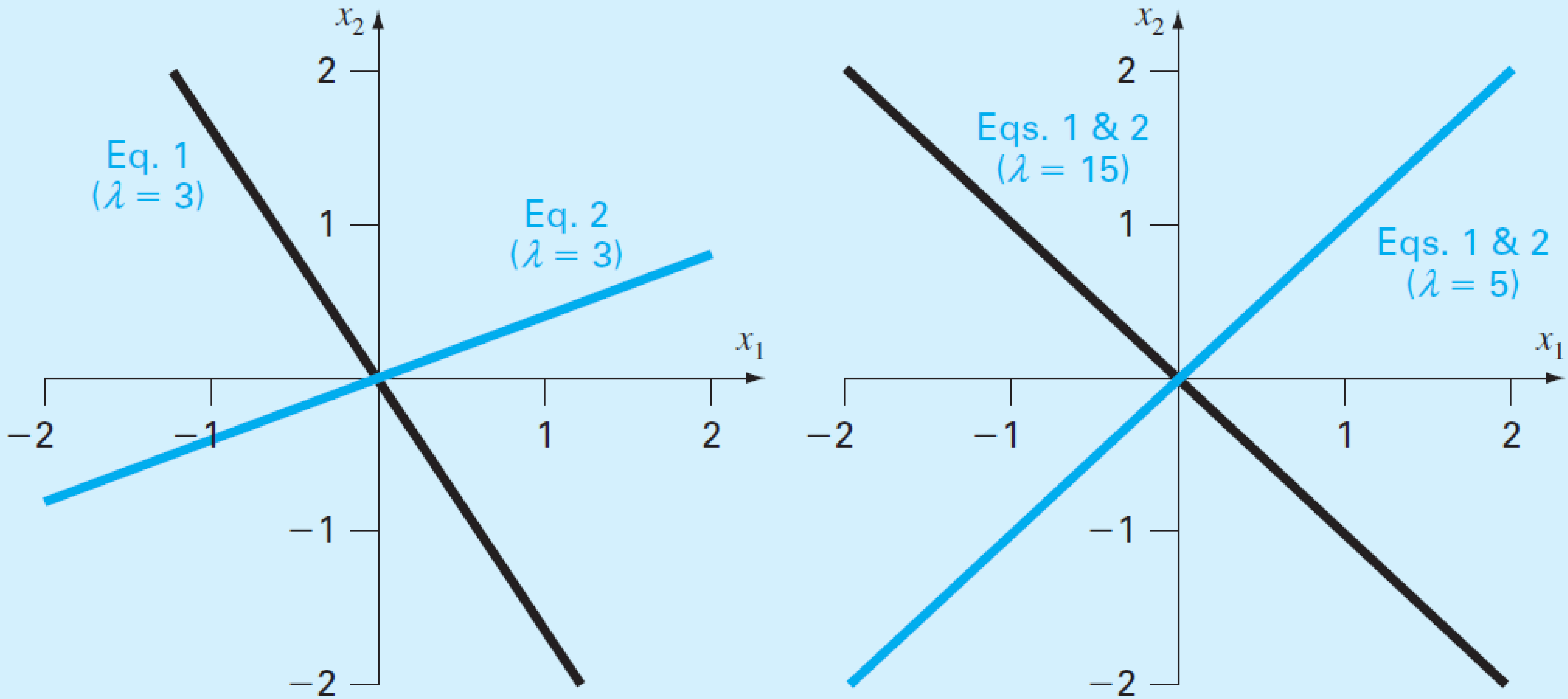
for $\lambda_2 = 5,$

$$\begin{aligned}5x_1 - 5x_2 &= 0\\ -5x_1 + 5x_2 &= 0\end{aligned} \qquad \{x\} = \begin{Bmatrix} 1 \\ 1 \end{Bmatrix}$$

for $\lambda = 3,$

$$\begin{aligned}7x_1 - 5x_2 &= 0\\ -5x_1 + 7x_2 &= 0\end{aligned} \qquad x_1 = x_2 = 0.$$

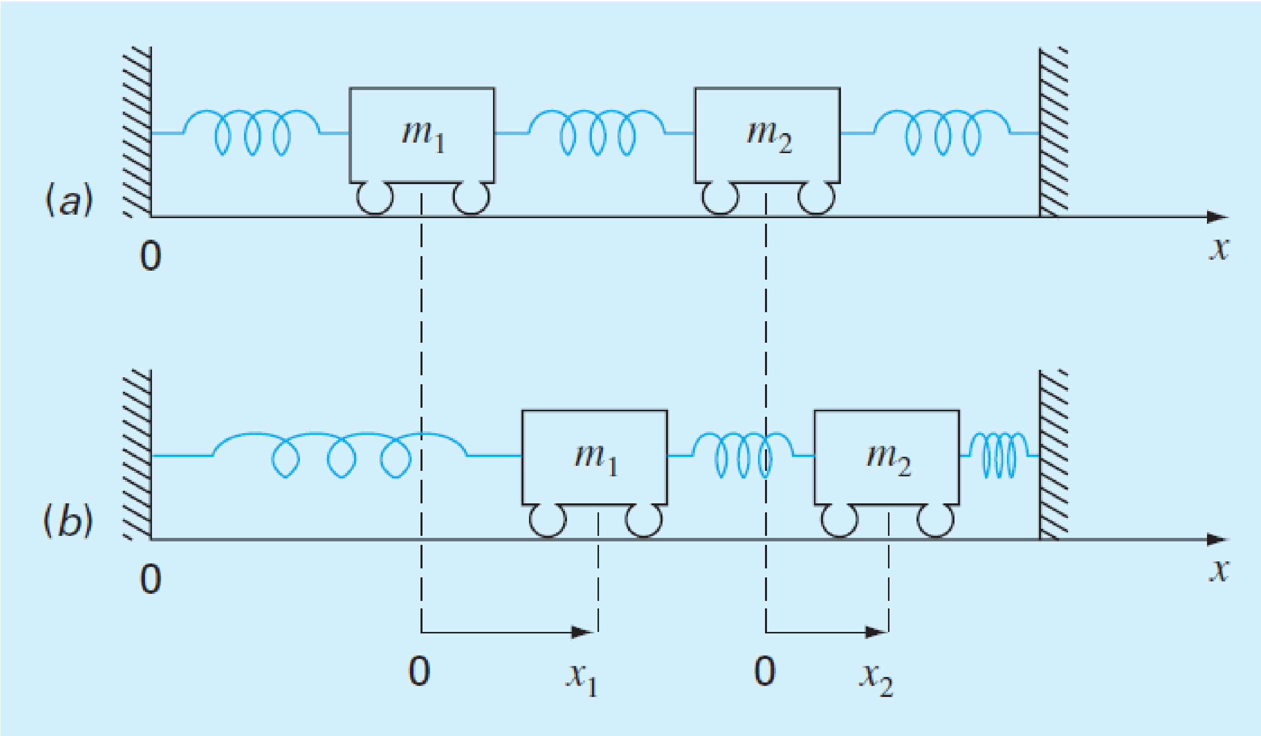
تنها جواب ممکن:



(*a*) Incorrect eigenvalue　　(*b*) Correct eigenvalues

در حالت (a) مقدار لامبدا برابر با عددی غیر از مقادیر ویژه سیستم است. لذا سیستم معادلات تنها دارای جواب بدیهی (۰,۰) می باشد. در حالیکه در حالت (b) سیستم به ازا لامبدا برابر مقادیر ویژه ترسیم شده است و مشاهده می شود که خطوط معادلات اول و دوم بر روی یکدیگر منطبق شده اند و این یعنی اینکه سیستم دارای بینهایت جواب می باشد. البته نسبت مجهولات همیشه عددی ثابت است. همچنین خطوط به ازا مقادیر ویژه متفاوت بر هم عمود هستند که این مورد برای کلیه ماتریس های متقارن با مقادیر ویژه مجزا صادق است.

مفهوم فیزیکی مقادیر ویژه و بردارهای ویژه



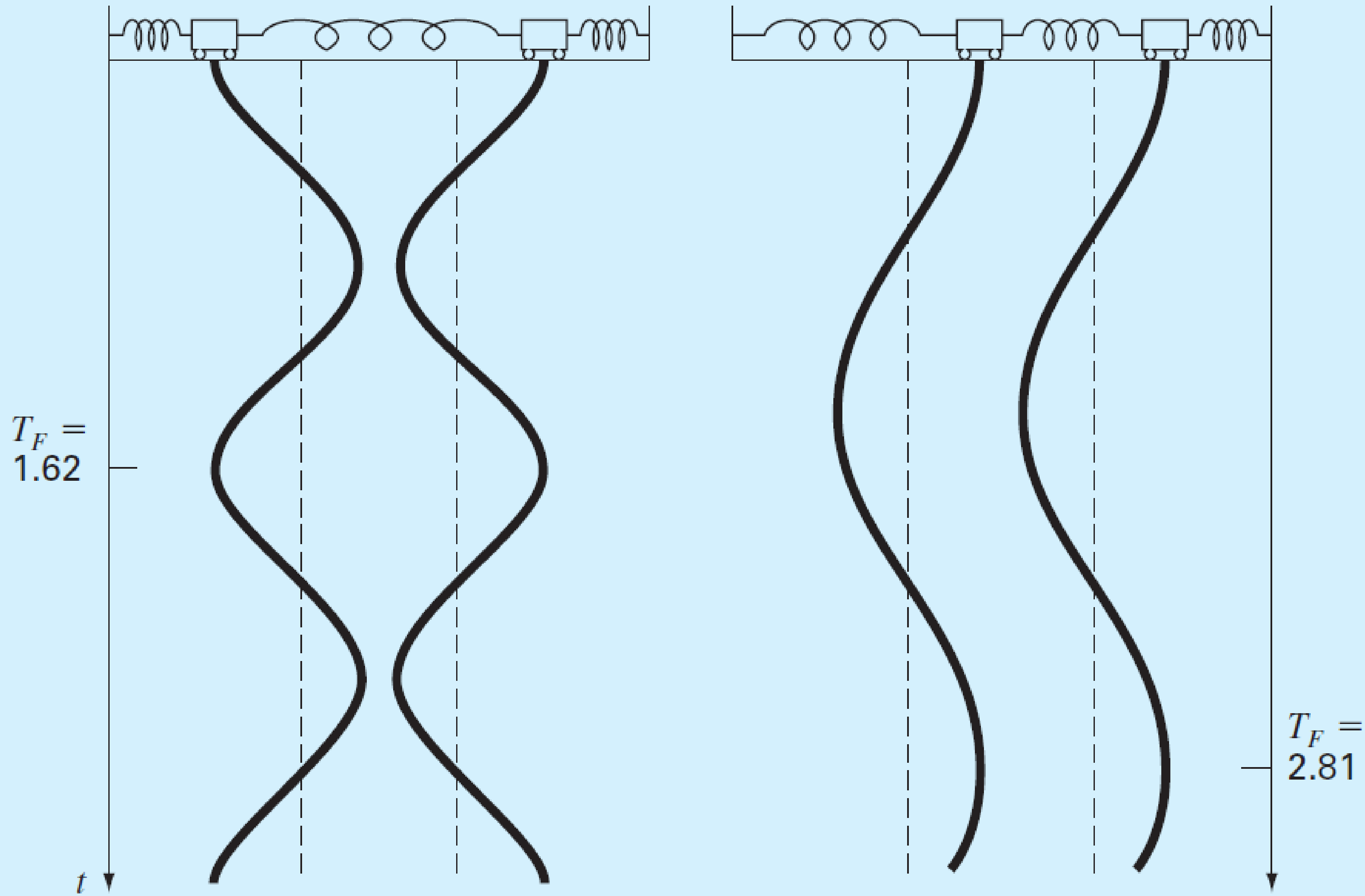
$$\left(\frac{2k}{m_1}-\omega^2\right)X_1-\frac{k}{m_1}X_2=0$$

$$-\frac{k}{m_2}X_1+\left(\frac{2k}{m_2}-\omega^2\right)X_2=0$$

$$\omega=\frac{2\pi}{T_p}$$

If $m_1 = m_2 = 40$ kg and $k = 200$ N/m,

$$(10 - \lambda)x_1 - 5x_2 = 0$$
$$-5x_1 + (10 - \lambda)x_2 = 0$$



(*a*) First mode

(*b*) Second mode

13.3 THE POWER METHOD

- این روش یک روش تکراری برای محاسبه بزرگترین مقدار ویژه (مقدار ویژه غالب) می باشد. با کمی تغییر روش می توان از آن برای محاسبه کوچکترین مقدار ویژه نیز استفاده نمود. برای این منظور از معکوس ماتریس ضرائب استفاده می شود.
- مزیت این روش در این است که همزمان با محاسبه مقدار ویژه، بردار ویژه نیز محاسبه می شود.
- در این روش از فرم زیر برای محاسبه مقدار ویژه استفاده می شود:

$$[A]\{x\} = \lambda\{x\}$$

مثال: محاسبه مقادیر و بردارهای ویژه با روش Power Method

$$
\begin{aligned}
40X_1 - 20X_2 \quad\quad\quad &= \lambda X_1 \\
-20X_1 + 40X_2 - 20X_3 &= \lambda X_2 \\
-20X_2 + 40X_3 &= \lambda X_3
\end{aligned}
$$

با حدس اولیه $\{X\} = \begin{Bmatrix} 1 \\ 1 \\ 1 \end{Bmatrix}$ سمت چپ تساوی محاسبه می شود.

$$
\begin{bmatrix} 40 & -20 & 0 \\ -20 & 40 & -20 \\ 0 & -20 & 40 \end{bmatrix} \begin{Bmatrix} 1 \\ 1 \\ 1 \end{Bmatrix} = \begin{Bmatrix} 20 \\ 0 \\ 20 \end{Bmatrix} = 20 \begin{Bmatrix} 1 \\ 0 \\ 1 \end{Bmatrix}
$$

پس از نرمال کردن بردار ویژه با مقدار $\{X\} = \begin{Bmatrix} 1 \\ 0 \\ 1 \end{Bmatrix}$ سمت چپ دوباره محاسبه می شود.

$$
\begin{bmatrix} 40 & -20 & 0 \\ -20 & 40 & -20 \\ 0 & -20 & 40 \end{bmatrix} \begin{Bmatrix} 1 \\ 0 \\ 1 \end{Bmatrix} = \begin{Bmatrix} 40 \\ -40 \\ 40 \end{Bmatrix} = 40 \begin{Bmatrix} 1 \\ -1 \\ 1 \end{Bmatrix}
$$

- با ادامه روند فوق مقادیر و بردارهای ویژه به اعداد ثابتی همگرا می شوند:

$$\begin{bmatrix} 40 & -20 & 0 \\ -20 & 40 & -20 \\ 0 & -20 & 40 \end{bmatrix} \begin{Bmatrix} 1 \\ -1 \\ 1 \end{Bmatrix} = \begin{Bmatrix} 60 \\ -80 \\ 60 \end{Bmatrix} = -80 \begin{Bmatrix} -0.75 \\ 1 \\ -0.75 \end{Bmatrix}$$

$$\begin{bmatrix} 40 & -20 & 0 \\ -20 & 40 & -20 \\ 0 & -20 & 40 \end{bmatrix} \begin{Bmatrix} -0.75 \\ 1 \\ -0.75 \end{Bmatrix} = \begin{Bmatrix} -50 \\ 70 \\ -50 \end{Bmatrix} = 70 \begin{Bmatrix} -0.71429 \\ 1 \\ -0.71429 \end{Bmatrix}$$

$$\begin{bmatrix} 40 & -20 & 0 \\ -20 & 40 & -20 \\ 0 & -20 & 40 \end{bmatrix} \begin{Bmatrix} -0.71429 \\ 1 \\ -0.71429 \end{Bmatrix} = \begin{Bmatrix} -48.51714 \\ 68.51714 \\ -48.51714 \end{Bmatrix} = 68.51714 \begin{Bmatrix} -0.70833 \\ 1 \\ -0.70833 \end{Bmatrix}$$

- مقدار ویژه نهایی: 68.28427

- بردار ویژه نهایی: $\{-0.707107 \; 1 \; -0.707107\}^T.$

- فرمهای مختلف دستور در متلب:

```
>> e = eig(A)
>> [V,D] = eig(A)
```

- مثال:

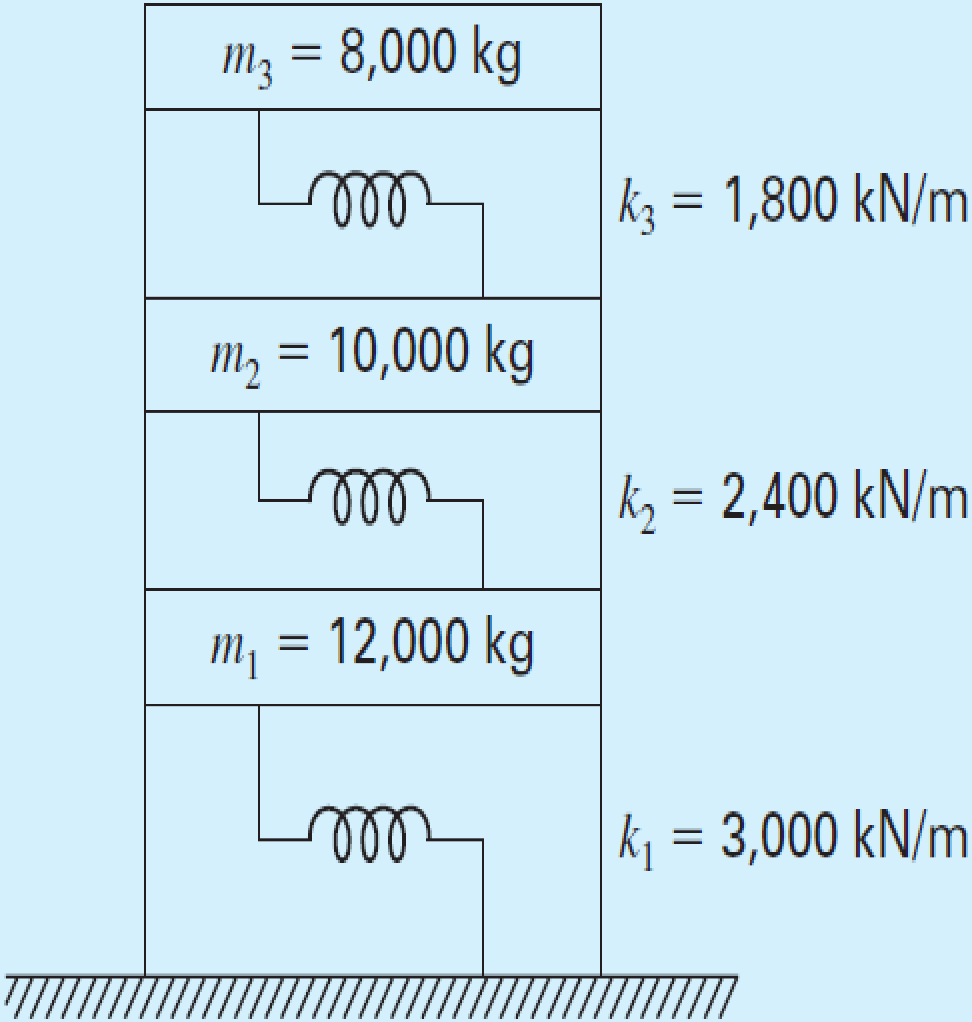
$$\begin{bmatrix} 40 & -20 & 0 \\ -20 & 40 & -20 \\ 0 & -20 & 40 \end{bmatrix}$$

```
>> A = [40 -20 0;-20 40 -20;0 -20 40];
```

```
>> e = eig(A)

e =
   11.7157
   40.0000
   68.2843
```

```
>> [v,d] = eig(A)

v =
    0.5000   -0.7071   -0.5000
    0.7071   -0.0000    0.7071
    0.5000    0.7071   -0.5000

d =
   11.7157         0         0
         0   40.0000         0
         0         0    8.2843
```

مثال: بیان مفهوم فیزیکی مقادیر و بردارهای ویژه



$$\left(\frac{k_1+k_2}{m_1}-\omega_n^2\right)X_1 - \frac{k_2}{m_1}X_2 = 0$$

$$-\frac{k_2}{m_2}X_1 + \left(\frac{k_2+k_3}{m_2}-\omega_n^2\right)X_2 - \frac{k_3}{m_2}X_3 = 0$$

$$-\frac{k_3}{m_3}X_2 + \left(\frac{k_3}{m_3}-\omega_n^2\right)X_3 = 0$$

$$(450-\omega_n^2)X_1 - 200X_2 = 0$$

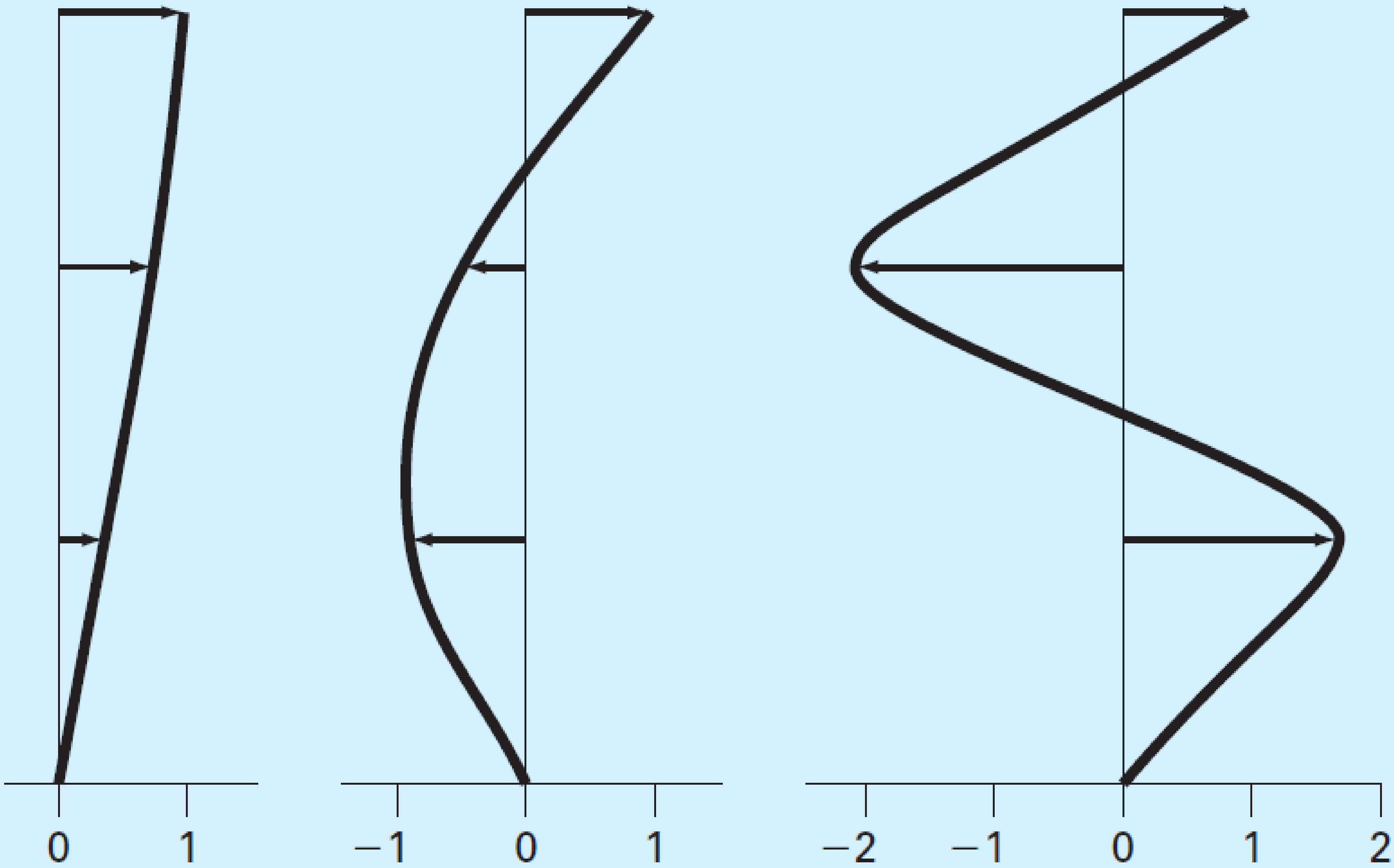
$$-240X_1 + (420-\omega_n^2)X_2 - 180X_3 = 0$$

$$-225X_2 + (225-\omega_n^2)X_3 = 0$$

```
>> A=[450 -200 0;-240 420 -180;0 -225 225];
>> [v,d]=eig(A)

v =
   -0.5879   -0.6344    0.2913
    0.7307   -0.3506    0.5725
   -0.3471    0.6890    0.7664

d =
  698.5982         0         0
         0  339.4779         0
         0         0   56.9239

>> wn=sqrt(diag(d))'/2/pi

wn =

    4.2066    2.9324    1.2008
```

$$\begin{Bmatrix}1.6934\\-2.1049\\1\end{Bmatrix}\quad\begin{Bmatrix}-0.9207\\-0.5088\\1\end{Bmatrix}\quad\begin{Bmatrix}0.3801\\0.7470\\1\end{Bmatrix}$$

مفهوم فیزیکی بردارهای ویژه



Mode 1 (ω_n = 1.2008 Hz)
Mode 2 (ω_n = 2.9324 Hz)
Mode 3 (ω_n = 4.2066 Hz)

تمرین ۱:

- از روش Power Method برای محاسبه بزرگترین مقدار ویژه و بردار ویژه ماتریس زیر استفاده نمائید:

$$\begin{bmatrix} 2-\lambda & 8 & 10 \\ 8 & 4-\lambda & 5 \\ 10 & 5 & 7-\lambda \end{bmatrix}$$

- برای ماتریس فوق کوچکترین مقدار ویژه و بردار ویژه مربوط به آن را نیز بدست آورید.

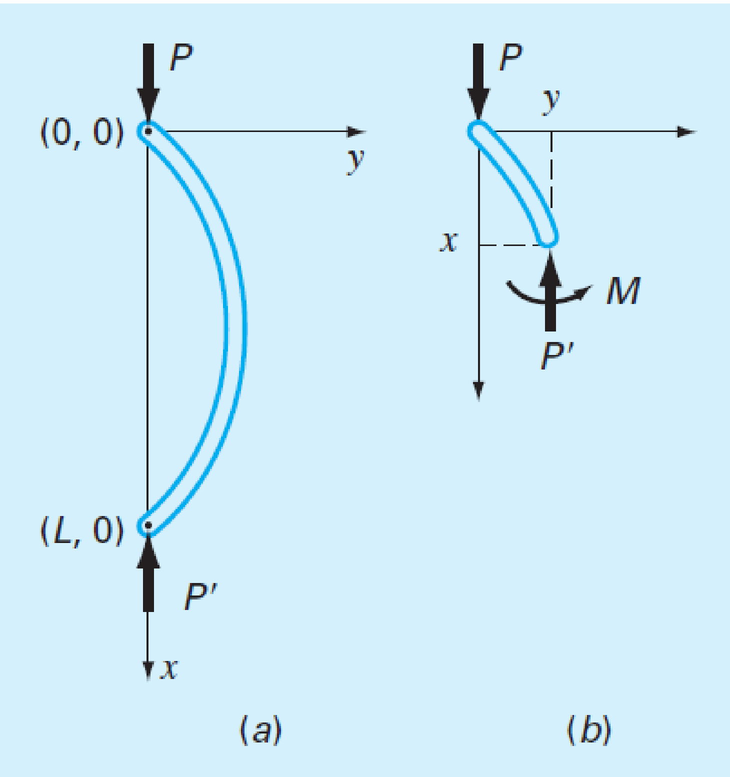
تمرین ۲:

- یک دستگاه معادلات دیفرانسیل مرتبه اول مطابق زیر مفروض است.

$$\frac{dy_1}{dt} = -5y_1 + 3y_2, \qquad y_1(0) = 50$$

$$\frac{dy_2}{dt} = 100y_1 - 301y_2, \qquad y_2(0) = 100$$

- پاسخ این سیستم به فرم $y_i = ce^{\lambda t}$ می باشد. با جایگذاری این پاسخ در معادلات فوق فرم مقدار ویژه دستگاه را بدست آورید.
- با استفاده از دستورات متلب، مقادیر ویژه و بردارهای ویژه را بدست آورید.
- با استفاده از نتایج مرحله قبل و شرایط اولیه پاسخ کلی سیستم را بدست آورید.
- پاسخ سیستم را در بازه زمانی ۰ تا ۱ ثانیه ترسیم نمائید.



$E = 10 \times 10^9$ Pa, $I = 1.25 \times 10^{-5}$ m^4, and $L = 3$ m.

تمرین ۳:

یک ستون تحت بار محوری P مطابق شکل مفروض است.
معادله حاکم بر تغییر شکل آن به فرم زیر خواهد بود:

$$\frac{d^2y}{dx^2} + p^2 y = 0 \qquad \text{where} \qquad p^2 = \frac{P}{EI}$$

با استفاده از تفاضل محدود می توان معادله فوق را به شکل
زیر گسسته سازی کرد که به فرم دستگاه مقادیر ویژه در می آید.

$$\frac{y_{i+1} - 2y_i + y_{i-1}}{\Delta x^2} + p^2 y_i = 0 \quad \Rightarrow$$

$$y_{i-1} - (2 - \Delta x^2 p^2) y_i + y_{i+1} = 0 \quad \Rightarrow$$

برای پنج قسمت (چهار گره داخلی) خواهیم داشت:

$$\begin{bmatrix} (2-\Delta x^2 p^2) & -1 & 0 & 0 \\ -1 & (2-\Delta x^2 p^2) & -1 & 0 \\ 0 & -1 & (2-\Delta x^2 p^2) & -1 \\ 0 & 0 & -1 & (2-\Delta x^2 p^2) \end{bmatrix} \begin{Bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \end{Bmatrix} = 0$$

الف) با استفاده از چند جمله ای مشخصه و با کمک متلب مقادیر ویژه این سیستم را بدست آورید.
ب) از دستور eig متلب برای محاسبه مقادیر ویژه و بردارهای ویژه استفاده نمائید.
ج) از روش Power Method استفاده کرده و بزرگترین مقدار ویژه و بردار ویژه مربوط به آن را محاسبه کنید.