ALN – Sistemas Lineales

In. Co.
Facultad de Ingeniería
Universidad de la República

Introducción

$$Ax = b$$

Donde A es una matriz dada de nxn, b es un vector dado de largo n y x es un vector de incógnita (también de largo n).

La matriz A es invertible.

Introducción

- Aplicaciones
  - Resolución de ecuaciones diferenciales
  - Problemas de optimización.
  - Resolución de problemas no lineales (Newton-Raphson)
  - Muchos otros problemas ……

Introducción

- Distintas estrategias:
  - Métodos directos.
  - Métodos iterativos.

Introducción

Métodos directos:

- Llegan a la solución en un número específico de pasos, en función del tamaño del sistema a resolver.
- Consiguen, a menos de errores numéricos, la solución exacta del sistema lineal.
- Disponen de estrategias para reducir los efectos de los problemas ocasionados por el mal condicionamiento de las matrices, como son las técnicas de pivoteo.
- Para la resolución de matrices dispersas utilizando métodos directos es necesario controlar la aparición de nuevos coeficientes no nulos (*fill in*).

Introducción

Métodos iterativos:

- Intentan aproximar la solución mediante una sucesión generada iterativamente.
- Si el método converge se obtiene una solución aproximada del problema, cuyo error satisface algún criterio prefijado.

Métodos directos

Resolución de sistemas particulares

- Sistemas de una diagonal
  - Sistema Ax=b, donde la matriz A es diagonal.

$$\begin{pmatrix} 4 & 0 & 0 & 0 & 0 \\ 0 & 5 & 0 & 0 & 0 \\ 0 & 0 & 5 & 0 & 0 \\ 0 & 0 & 0 & 4 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{pmatrix}$$

Resolución de sistemas particulares

- Sistemas triangulares
  - Sistema Ax=b, donde la matriz A es triangular (superior o inferior)

$$\begin{pmatrix} 4 & 0 & 0 & 0 & 0 \\ 2 & 5 & 0 & 0 & 0 \\ 3 & 3 & 5 & 0 & 0 \\ 2 & 1 & 9 & 4 & 0 \\ 3 & 4 & 7 & 8 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{pmatrix} \qquad \begin{pmatrix} 4 & 4 & 6 & 7 & 8 \\ 0 & 5 & 1 & 0 & 3 \\ 0 & 0 & 5 & 6 & 4 \\ 0 & 0 & 0 & 4 & 2 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{pmatrix}$$

Resolución de sistemas particulares

- Sistemas triangular superior (sustitución hacia atrás)

$$\begin{pmatrix} u_{11} & . & . & . & u_n \\ 0 & . & . & . & . \\ 0 & 0 & . & . & . \\ 0 & 0 & 0 & . & . \\ 0 & 0 & 0 & 0 & u_{nn} \end{pmatrix} \begin{pmatrix} x_1 \\ \\ . \\ \\ x_n \end{pmatrix} = \begin{pmatrix} b_1 \\ . \\ . \\ . \\ b_n \end{pmatrix}$$

$$x_n = \frac{b_n}{u_{nn}}$$

$$x_i = \frac{b_i - \sum_{k=i+1}^{n} x_k u_{ik}}{u_{ii}}$$

Resolución de sistemas particulares

- Contando operaciones ….

$$op \approx \sum_{k=1}^{n}(k-1) = \frac{1}{2}(n)(n-1) \approx \frac{1}{2}n^2$$

Lo que implica que el algoritmo sea de $O(n^2)$

Resolución de sistemas particulares

- Sistemas triangular inferior
  - sustitución hacia delante
  - análogo a la sustitución hacia atrás

Eliminación Gaussiana

- Uno de los métodos más difundidos.
- También denominada escalerización y eliminación de Gauss (por Karl Gauss [1777-1855]).
- Asume que la matriz A es invertible, por lo que el sistema posee una única solución.

Eliminación Gaussiana

- Consiste en transformar la matriz A y el vector b, mediante una serie de transformaciones elementales (intercambiar dos filas, multiplicar una fila por un escalar distinto de 0 o sumar dos filas) para convertir el sistema en uno triangular.

Eliminación Gaussiana

- Para cada columna j desde la 1 a la n (en orden), "eliminar" los coeficientes distintos de 0 a partir de la fila j.
  - Para eliminar el coeficiente a_{ij} de la columna j resto la fila i menos la fila j multiplicada por $m_{ij} = \frac{a_{ij}}{a_{jj}}$

- Hay que realizar los mismos cambios en el vector b para que el sistema sea equivalente

Eliminación Gaussiana

- La segunda etapa del método consiste en resolver el sistema lineal triangular.
- Para esta etapa se utiliza la estrategia vista anteriormente.

Eliminación Gaussiana

- Ejemplo:

$$Ax = b \to \begin{pmatrix} 1 & 1 & 1 & 1 \\ 2 & 3 & 5 & 4 \\ 1 & 3 & 9 & 6 \\ 2 & 3 & 7 & 9 \end{pmatrix} x = \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \end{pmatrix} \to \begin{pmatrix} 1 & 1 & 1 & 1 \\ 0 & 1 & 3 & 2 \\ 0 & 2 & 8 & 5 \\ 0 & 1 & 5 & 7 \end{pmatrix} x = \begin{pmatrix} 1 \\ -1 \\ 0 \\ -1 \end{pmatrix} \quad \cdots \quad \begin{pmatrix} 1 & 1 & 1 & 1 \\ 0 & 1 & 3 & 2 \\ 0 & 0 & 2 & 1 \\ 0 & 0 & 0 & 4 \end{pmatrix} x = \begin{pmatrix} 1 \\ -1 \\ 2 \\ -2 \end{pmatrix}$$

$$Ax = b \to Lx = \tilde{b}, \tilde{b} = \begin{pmatrix} 1 \\ -1 \\ 2 \\ -2 \end{pmatrix} \to x = \begin{pmatrix} 4.0 \\ -3.75 \\ 1.25 \\ -0.5 \end{pmatrix}$$

Eliminación Gaussiana

- Escalerizando ...

$$Ax = b \rightarrow M_1 Ax = M_1 b$$

$$M_1 = \begin{pmatrix} 1 & 0 & \cdots & & 0 \\ -m_{21} & 1 & \ddots & & \vdots \\ -m_{31} & 0 & \ddots & & \\ \vdots & \vdots & \ddots & & 0 \\ -m_{n1} & 0 & \cdots & 0 & 1 \end{pmatrix}$$

$$M_1 Ax = M_1 b \rightarrow M_{n-1} \ldots M_1 Ax = M_{n-1} \ldots M_1 b$$

Eliminación Gaussiana

- Contando operaciones ….
  - Para cada columna k (todas las columnas excepto la última):
    - Calculo (n-k) multiplicadores: (n-k) divisiones.
    - Actualizo $(n\text{-}k)^2$ elementos: una mult. y una suma por cada uno.

$$\sum_{k=1}^{n-1}(n-k)+2\sum_{k=1}^{n-1}(n-k)^2=\frac{2n(n-1)(2n-1)}{6}+\frac{n(n-1)}{2}=\frac{4n^3-3n^2-n}{6}=\frac{2(n^3)}{3}$$

- Por lo tanto el orden del algoritmo es $O\left(n^3\right)$

Eliminación Gaussiana

- Los elementos a_{ii} (en el paso i de la escalerización) se denominan pivotes.

- ¿Que pasa si uno de los pivotes es (cercano a) cero?
  - Pivoteo!!!!!: Intercambiar filas/columnas para que los pivotes sean distintos de cero.
  - Dos estrategias:
    - Pivoteo parcial
    - Pivoteo completo

Eliminación Gaussiana

- Estrategias de pivoteo:
  - Pivoteo parcial: intercambiar las filas *k* y *r* siendo *k* la columna actual y *r* el mínimo entero para el cual

$$\left|a_{rk}^{(k)}\right| = \max_{k \le i \le n} \left|a_{ik}^{(k)}\right|$$

  - Pivoteo completo: intercambiar las filas *k* con *r* y columnas *k* con *s* tal que *r* y *s* son los mínimos enteros para los cuales

$$\left|a_{rs}^{(k)}\right| = \max_{k \le i,\, j \le n} \left|a_{ij}^{(k)}\right|$$

Eliminación Gaussiana

- Se pueden resolver varios vectores independientes (varios b).
- Se escalerizan todos juntos, luego se resuelve cada uno de los sistemas triangulares.

Factorización LU

- Ax = b --> LU=A --> Ly = b, Ux=y
  - L es triangular inferior
  - U es triangular superior
  - Si utilizamos factorización LU, con la matriz L con unos en la diagonal la descomposición es única.
- Es útil cuando quiero resolver varios vectores derechos sin conocerlos

Factorización LU

- Relación con la escalerización

$$M_1 Ax = M_1 b \to M_{n-1} \ldots . M_1 Ax = M_{n-1} \ldots . M_1 b$$

$$MA = U \to A = M^{-1} U \to M^{-1} = L$$

Recordar que la matriz M es la multiplicación de matrices triangulares inferiores

Factorización LU

- Un algoritmo

```
U=A;
L=I;
For k=1:n-1
    For j=k+1:n
        L(j,k)=U(j,k)/U(k,k)
        U(j,k:n)=U(j,k:n)-L(j,k)U(k,k:n)
```

Factorización LU

- Un algoritmo “in place”

```
For i=1:n-1
   For j=i+1:n
   A(j,i)=A(j,i)/A(i,i)
   For k=i+1:n
      A(j,k)=A(j,k)-A(j,i)A(i,k)
```

Factorización LU

- Acceso a la matriz
  - Por bloques
  - Con distinto patrón de acceso

  - Se realizan operaciones matriz-matriz que son más eficientes que vector-vector

- LU a bloques
  - Descomponer la matriz A en sub-bloques

$$A = \begin{pmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{pmatrix} = \begin{pmatrix} L_{11} & 0 \\ L_{21} & L_{22} \end{pmatrix} \begin{pmatrix} U_{11} & U_{12} \\ 0 & U_{22} \end{pmatrix}$$

Factorización LU

- Un algoritmo por bloques

$$A_{11} = L_{11}U_{11}$$
$$A_{21} = L_{21}U_{11}$$
$$A_{12} = L_{11}U_{12}$$
$$A_{22} = L_{21}U_{12} + L_{22}U_{22}$$

$$A_{11} = L_{11}U_{11}$$
$$A_{21} = XU_{11}$$
$$A_{12} = L_{11}Y$$

Se resuelve L_{11} y U_{11}

Se resuelve L_{21}

Se resuelve U_{12}

Luego se actualiza $\tilde{A}_{22} = A_{22} - L_{21}U_{12}$

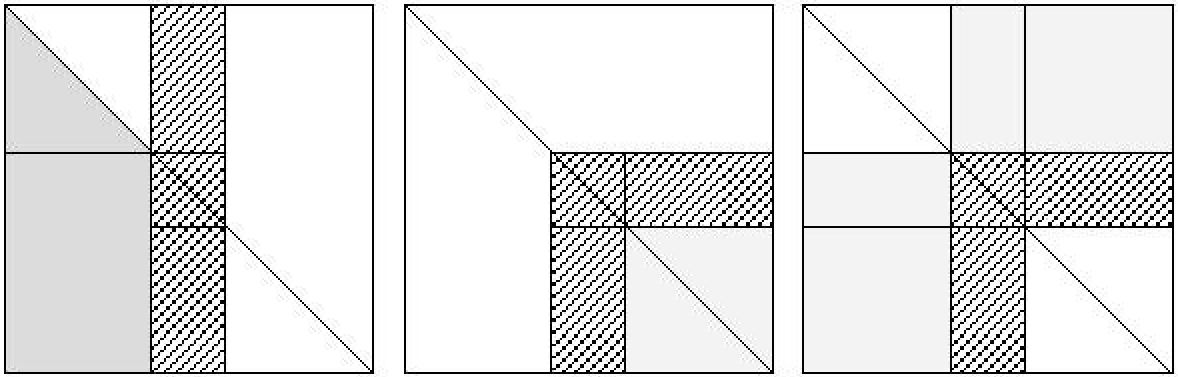
Por último se factoriza $\tilde{A}_{22}$

Factorización LU

Diferentes tipo de patrón de acceso a la matriz (por bloques) según que datos utiliza/calcula.

Factorización LU

- Diferentes estrategias de acceso



Left-looking Right-looking Crout-LU

- Acceso a la matriz “Right-looking”, el que vimos anteriormente

$$A = \begin{pmatrix} L_{11} & & \\ L_{21} & I & \\ L_{31} & 0 & I \end{pmatrix} \begin{pmatrix} U_{11} & U_{12} & U_{13} \\ & \tilde{A}_{22} & \tilde{A}_{23} \\ & \tilde{A}_{32} & \tilde{A}_{33} \end{pmatrix}$$

Factorización LU

- Acceso "Left"

$$A=\begin{pmatrix} L_{11} & & \\ L_{21} & I & \\ L_{31} & 0 & I \end{pmatrix}\begin{pmatrix} U_{11} & A_{12} & A_{13} \\ & A_{22} & A_{23} \\ & A_{32} & A_{33} \end{pmatrix}$$

$$U_{12}=L_{11}^{-1}A_{12}$$
$$\tilde{U}_{22}=A_{22}-L_{21}U_{12}$$
$$\tilde{U}_{32}=A_{32}-L_{31}U_{12}$$

$$\begin{pmatrix} \tilde{U}_{22} \\ \tilde{U}_{32} \end{pmatrix}=\begin{pmatrix} A_{22} \\ A_{32} \end{pmatrix}-\begin{pmatrix} L_{21} \\ L_{31} \end{pmatrix}U_{12}$$

$$A=\begin{pmatrix} L_{11} & & \\ L_{21} & I & \\ L_{31} & 0 & I \end{pmatrix}\begin{pmatrix} U_{11} & U_{12} & A_{13} \\ & \tilde{U}_{22} & A_{23} \\ & \tilde{U}_{32} & A_{33} \end{pmatrix}$$

Factorizando

$$\begin{pmatrix} \tilde{U}_{22} \\ \tilde{U}_{32} \end{pmatrix}\rightarrow\begin{pmatrix} L_{22} \\ L_{32} \end{pmatrix}U_{22}$$

$$A=\begin{pmatrix} L_{11} & & \\ L_{21} & L_{22} & \\ L_{31} & L_{32} & I \end{pmatrix}\begin{pmatrix} U_{11} & U_{12} & A_{13} \\ & U_{22} & A_{23} \\ & 0 & A_{33} \end{pmatrix}$$

Factorización LU

- Acceso Crout

$$A = \begin{pmatrix} L_{11} & & \\ 0 & I & \\ 0 & 0 & I \end{pmatrix} \begin{pmatrix} I & U_{12} & U_{13} \\ L_{21} & A_{22} & A_{23} \\ L_{31} & A_{32} & A_{33} \end{pmatrix} \begin{pmatrix} U_{11} & 0 & 0 \\ & I & 0 \\ & & I \end{pmatrix}$$

$$\begin{pmatrix} A_{22} & A_{23} \end{pmatrix} \leftarrow \begin{pmatrix} A_{22} & A_{23} \end{pmatrix} - L_{21} \begin{pmatrix} U_{12} & U_{13} \end{pmatrix}$$

$$A_{32} \leftarrow A_{32} - L_{21} U_{12}$$

$Factorizando$

$$\begin{pmatrix} A_{22} \\ A_{32} \end{pmatrix} \leftarrow \begin{pmatrix} L_{22} \\ L_{32} \end{pmatrix} U_{22}$$

$$\text{con } U_{23} = L_{21}^{-1} A_{23}$$

$$A = \begin{pmatrix} L_{11} & & \\ L_{21} & L_{22} & \\ 0 & 0 & I \end{pmatrix} \begin{pmatrix} I & 0 & U_{13} \\ 0 & I & A_{23} \\ L_{31} & A_{32} & A_{33} \end{pmatrix} \begin{pmatrix} U_{11} & U_{12} & 0 \\ & U_{22} & 0 \\ & & I \end{pmatrix}$$

Factorización LU

- Estrategias de acceso, conclusiones:
  - mejoran la localidad de datos en el acceso, tanto de lectura como de escritura.
  - el acceso por columna se puede hacer por fila.
  - mejoran el trabajo en computadores vectoriales.
  - mejoran el desempeño al utilizar bibliotecas blas.

Factorización LU

- Pivoteo en LU
  - $LU=PA$, $PAx=Pb$
- En la factorización LU, el pivoteo parcial se aplica de la siguiente forma:
  - $Ly=Pb$, $Ux = y$

Factorización LU

- Pivoteo en bloques …
  - completo muy difícil ..
  - solo “pivotear” en el bloque !!
  - otras soluciones
    - Pivoteo 2*2
    - Matrices aleatorias

Factorización LU

Matrices especiales

- Simétricas (indefinidas)
  - Descomposición LDL^t
- Simétricas y definidas positivas (Cholesky)
  - Descomposición LL^t

Errores

Ejemplo: $A = \begin{bmatrix} 1 & 1 \\ 10.05 & 10 \end{bmatrix}$ $\quad b = \begin{bmatrix} 2 \\ 21 \end{bmatrix}$ $\quad$ solución: $x = \begin{bmatrix} 20 \\ -18 \end{bmatrix}$

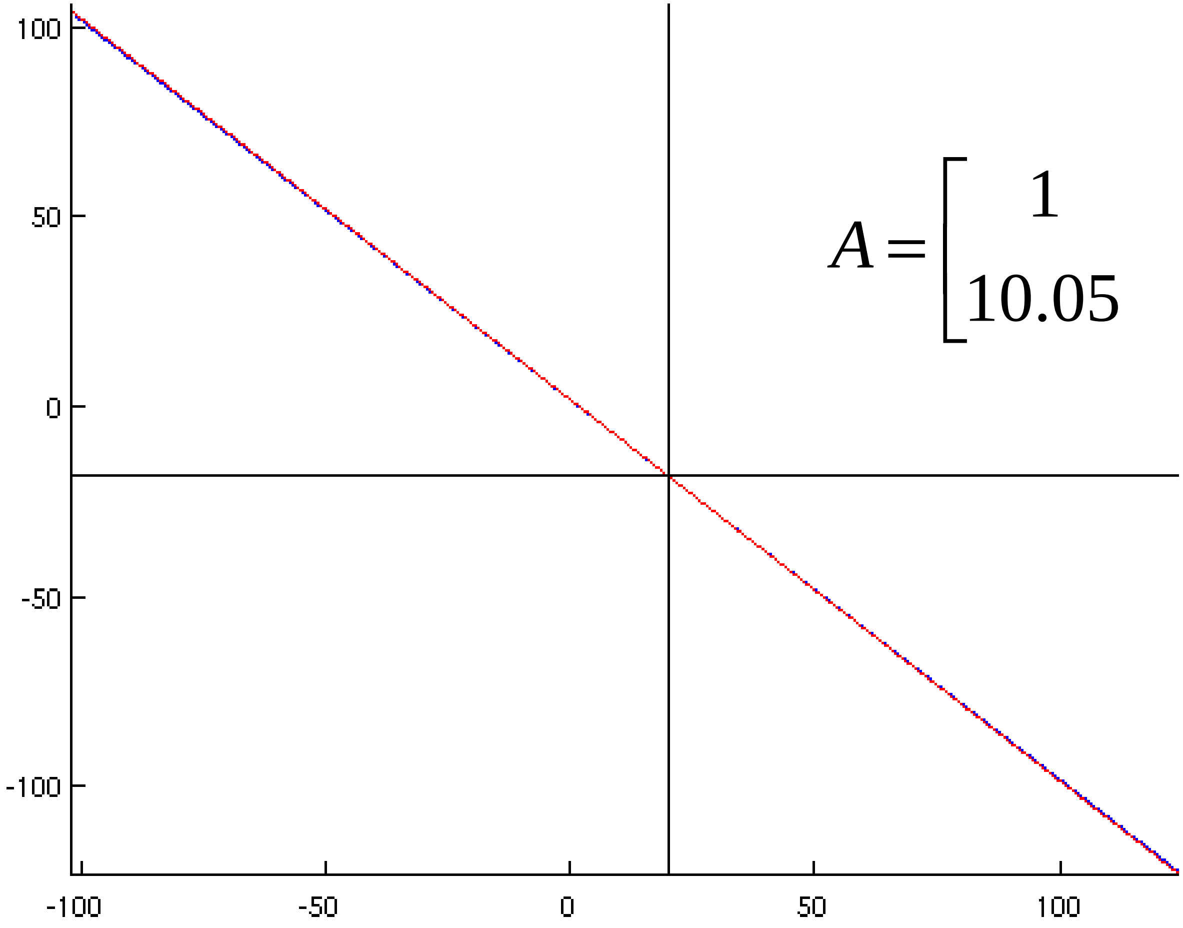
Si modificamos “un poco” el sistema esperaríamos una pequeña variación en la solución, pero esto no siempre sucede:

$A = \begin{bmatrix} 1 & 1 \\ 10.1 & 10 \end{bmatrix}$ $\quad b = \begin{bmatrix} 2 \\ 21 \end{bmatrix}$ $\quad$ solución: $x = \begin{bmatrix} 10 \\ -8 \end{bmatrix}$

Errores

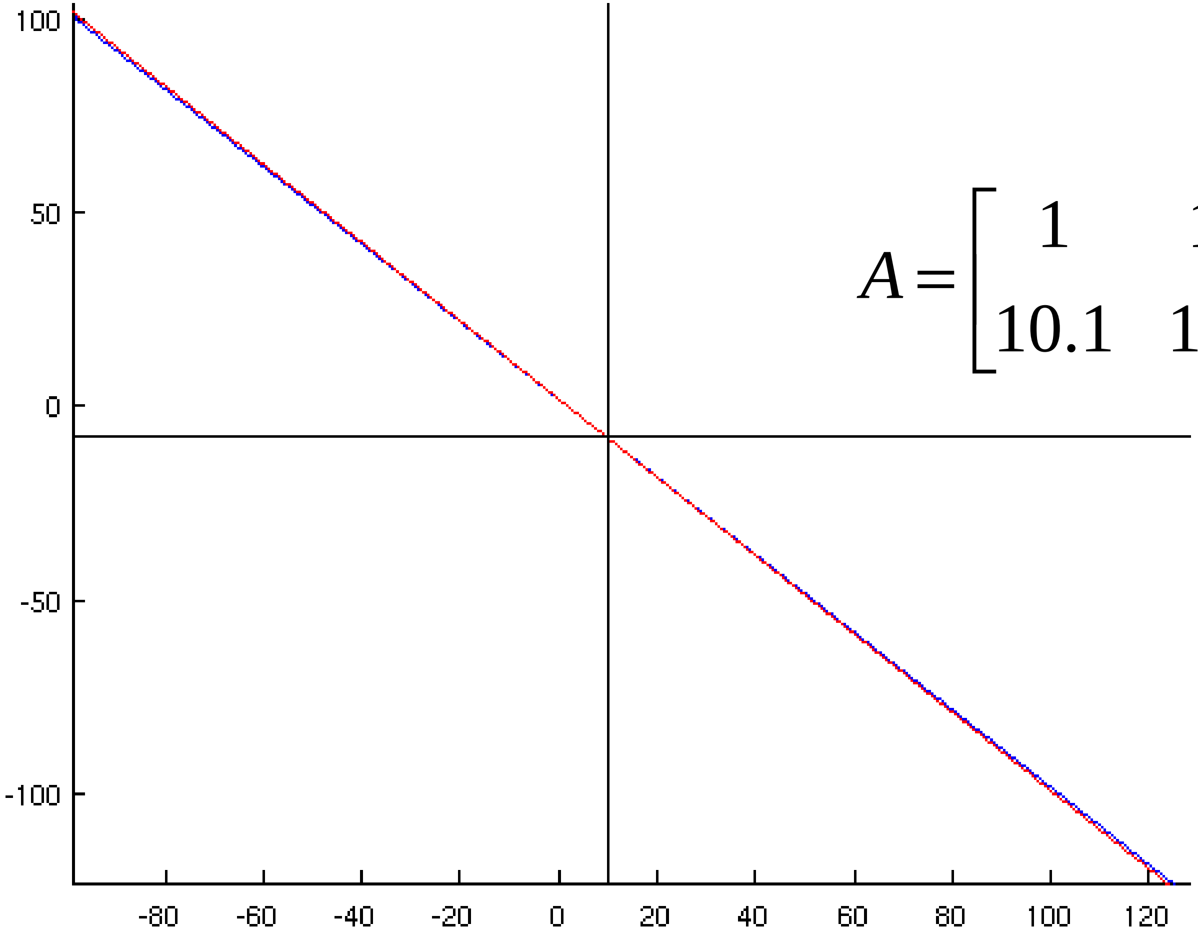
- Otra forma de verlo: como rectas…

Errores



$$A = \begin{bmatrix} 1 & 1 \\ 10.05 & 10 \end{bmatrix} \qquad b = \begin{bmatrix} 2 \\ 21 \end{bmatrix}$$

Errores



$$A = \begin{bmatrix} 1 & 1 \\ 10.1 & 10 \end{bmatrix} \qquad b = \begin{bmatrix} 2 \\ 21 \end{bmatrix}$$

Errores…

- Perturbando las matrices conocidas y acotando errores
- Analizando la estabilidad

Estudio del error

- Si perturbamos b,

$$A(x+\delta x)=b+\delta b \longrightarrow A\delta x=\delta b$$

$$\delta x = A^{-1}\delta b \longrightarrow \left\|\delta x\right\| \leq \left\|A^{-1}\right\|\left\|\delta b\right\|$$

$$\frac{\left\|\delta x\right\|}{\left\|x\right\|} \leq \left\|A\right\|\left\|A^{-1}\right\|\frac{\left\|\delta b\right\|}{\left\|b\right\|}$$

Estudio del error

- Si perturbamos A

$$(A+\delta A)(x+\delta x)=b$$

$$A\delta x=-\delta A(\delta x+x)\rightarrow \delta x=A^{-1}(-\delta A)(\delta x+x)$$

$$\|\delta x\| \leq \frac{\|A\|\,\|A^{-1}\|\,\|\delta A\|}{\|A\|}\|x+\delta x\|$$

$$\frac{\|\delta x\|}{\|x+\delta x\|}\leq\|A\|\|A^{-1}\|\frac{\|\delta A\|}{\|A\|}$$

Estudio del error

- Si además $\|A^{-1}\| \|\delta A\| < 1$

$$\|\delta x\| \leq \|A^{-1}\| \|\delta A\| \|x + \delta x\|$$

$$\|\delta x\| \leq \|A^{-1}\| \|\delta A\| \|x + \delta x\| \leq \|A^{-1}\| \|\delta A\| \|x\| + \|A^{-1}\| \|\delta A\| \|\delta x\|$$

$$1 - \frac{\|A\| \|A^{-1}\| \|\delta A\|}{\|A\|} \leq \frac{\|A\| \|A^{-1}\| \|\delta A\|}{\|A\|} \frac{\|x\|}{\|\delta x\|}$$

$$\frac{\|\delta x\|}{\|x\|} \leq \left(\frac{\|A\| \|A^{-1}\| \|\delta A\|}{\|A\|} \right) / \left(1 - \frac{\|A\| \|A^{-1}\| \|\delta A\|}{\|A\|} \right)$$

Estudio del error

Número de condición:

El factor que controla la amplificación de los errores relativos de los datos

$$cond(A) = \|A\| \|A^{-1}\|$$

Estudio del error

- Con esta definición podemos reescribir las fórmulas anteriores para la perturbación relativa de x:
  - Perturbando b

$$\frac{||\delta x||}{||x||} \leq cond(A)\,\frac{||\delta b||}{||b||}$$

  - Perturbando A

$$\frac{||\delta x||}{||x||} \leq \left(cond(A)\,\frac{||\delta A||}{||A||}\right) / \left(1 - cond(A)\,\frac{||\delta A||}{||A||}\right)$$

Estudio del error

- Si perturbamos A y b,

$$(A+\delta A)(x+\delta x)=b+\delta b$$

$$\frac{\|\delta x\|}{\|x\|}\leq\frac{cond(A)}{1-cond(A)\dfrac{\|\delta A\|}{\|A\|}}\left(\frac{\|\delta A\|}{\|A\|}+\frac{\|\delta b\|}{\|b\|}\right)$$

Algunos conceptos matemáticos

- Teorema 1: Cualquier norma matricial es una función continua de los elementos de la matriz.
- Teorema 2: Para cada par de normas matriciales existen constantes positivas *m y M* tales que :

$$m\|A\|_i \leq \|A\|_j \leq M\|A\|_i$$

Algunos conceptos matemáticos

- Teorema 3: Para cualquier norma natural y cualquier matriz cuadrada se verifica:

$$\rho(A) \leq \|A\|$$

- Teorema 4: Para cualquier matriz cuadrada y cualquier valor existe alguna norma natural tal que:

$$\rho(A) \leq \|A\| \leq \rho(A) + \varepsilon$$

- Corolario: Para cualquier matriz cuadrada:

$$\rho(A) \leq \inf \|A\|$$

Estudio del error

- Para la norma euclidiana y para una matriz normal (AA* = A*A)

$$cond(A) = \frac{\lambda_1}{\lambda_n}$$

Estudio del error

- Otra forma de obtener en forma exacta el número de condición de una matriz es mediante la formula:

$$cond(A) = \frac{\sigma_1}{\sigma_n}$$

$$donde, \sigma_1 \text{ es el mayor valor singular y } \sigma_n \text{ el menor}$$

Estudio del error

- Estabilidad de la factorización L U, ante los errores de redondeo.
- Como afectan la estrategias de pivoteo

Estudio del error

- El condicionamiento numérico indica como se amplifican los errores.
- Entonces, si introducimos errores de redondeo en las cuentas los resultados se verán afectados
- Se puede hacer:
  - <u>Análisis directo</u>: se estudia cómo se propagan los errores (resultados muy pesimistas).
  - <u>Análisis inverso</u>: se estiman los errores como perturbaciones de los datos y se intenta acotar los errores.

Estudio del error

- Si realizamos una factorización (sin pivoteo) de una matriz A (no singular) obtenemos $LU = A + \delta A$ debido a los errores de redondeo
- Se cumple el siguiente resultado: $\|\delta A\| \leq n\varepsilon \|L\| \|U\|$
- Definimos el factor de crecimiento respecto a la norma $\|.\|$ como $\rho = \dfrac{\|L\| \, \|U\|}{\|A\|}$

Estudio del error

- Con la def. anterior tenemos que: $\frac{\|\delta A\|}{\|A\|} \leq n\varepsilon\rho$
- ρ refleja el crecimiento en magnitud de las entradas de LU respecto de las de A.
- Si se utiliza pivoteo, los valores de la matriz L se obtienen mediante divisiones en las cuales el divisor siempre es mayor al dividendo. Podemos asumir que la norma de la matriz L no crecerá sustancialmente y que ρ está determinado por el crecimiento de U.

Estudio del error

- El factor de crecimiento que utiliza normas matriciales se sustituye por:

$$\rho = \frac{\max_{i,j} \left| u_{ij} \right|}{\max_{i,j} \left| a_{ij} \right|}$$

- Podemos reescribir la formula:

$$\frac{\left\| \delta A \right\|}{\left\| A \right\|} = O\left(\rho n \varepsilon_{mach} \right)$$

Estudio del error

- Estudiando un caso particular ….
- Con pivoteo parcial, todo solucionado ??

$$A = \begin{pmatrix} 1 & 0 & 1 \\ -1 & 1 & 1 \\ -1 & -1 & 1 \end{pmatrix}$$

Estudio del error

$$A = LU = \begin{pmatrix} 1 & 0 & 0 \\ -1 & 1 & 0 \\ -1 & -1 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & 2 \\ 0 & 0 & 4 \end{pmatrix} \qquad \rho = \frac{\max_{i,j} |u_{ij}|}{\max_{i,j} |a_{ij}|} = 4$$

Para una matriz con esta forma pero de dimensión n el ρ es 2^{n-1}

Y que paso con nuestra teoría ?

Estudio del error

- Para la gran mayoría de las matrices se puede afirmar que la factorización LU con pivoteo parcial es estable.
- Que tan grande es la mayoría ?!?!
  - Wilkinson en 1965 escribió: “Ningún ejemplo que haya surgido naturalmente en mi experiencia dio un factor de crecimiento mayor a 16”
  - Existen cotas más duras para matrices especiales:
    - Hessenberg: $\rho \leq n$
    - Tridiagonales: $\rho \leq 2$

Estudio del error

- En general es necesario algún tipo de pivoteo para garantizar la estabilidad de la factorización LU.
- Sin embargo existen familias de matrices para las cuales no es necesario pivoteo:
  - Matrices diagonal-dominantes
  - Matrices definidas positivas
  - Matrices totalmente positivas

Refinamiento iterativo

$$Ax = b \to A\tilde{x} = b$$

$$r = b - A\tilde{x}$$

$$A\delta x = r$$

$$x = \tilde{x} + \delta x$$

- Solo es útil si se trabaja con distintas precisiones
- Existen otros refinamientos sin esta exigencia pero son menos intuitivos

Escalado I

- Dividir cada fila de la matriz por el mayor valor de la fila.
- No garantiza mejorar el condicionamiento.

Escalado

$$\frac{\|\tilde{x}-x\|}{\|x\|} \approx \mu K(A)$$

$$(D_1^{-1} A D_2) y = D_1^{-1} b \rightarrow \mu K(D_1^{-1} A D_2)$$

$$Si \;\; K(D_1^{-1} A D_2) < K(A)$$

- Elegir D_1 para que todas las filas de A tengan la misma norma
- Elegir D_1 y D_2 para que las columnas y filas tengan norma similar

Es difícil determinar estrategias generales de escalado es necesario estudiar cada problema

Estimación del número de condición I

$$cond(A) = \|A\| \|A^{-1}\| \qquad \|A\| \text{ se puede calcular}$$

$$\text{Si A es no singular y B es singular} \rightarrow \|A^{-1}\| \geq \frac{1}{\|A - B\|}$$

$$cond(A) \geq \frac{\|A\|}{\|A - B\|}$$

Estimación del número de condición

Como B es singular $\exists x \neq 0 \, / \, Bx = 0$

$$Ax = (A - B)x \rightarrow A^{-1}Ax = A^{-1}(A - B)x$$

$$x = A^{-1}(A - B)x$$

$$1 \leq \left\| A^{-1} \right\| \left\| A - B \right\|$$

Si se toma la matriz B “parecida” a la matriz A, se obtiene una buena cota para el número de condición.

Estimación del número de condición II

$$cond(A) = \|A\| \|A^{-1}\|$$

$\|A\|$ se puede calcular

Un estimador $Ay = d \rightarrow \|A^{-1}\| \geq \dfrac{\|y\|}{\|d\|}$

$$cond(A) \approx \frac{\|A\| \|y\|}{\|d\|}$$

Otra aplicación de LU

Cálculo del determinante

$$\det(A) = \det(LU) = \det(L)\det(U)$$

Pero el determinante de matrices triangulares es igual a la multiplicatoria de la diagonal

$$\det(L)\det(U) = \det(U) = u_{11} * u_{22} * \ldots * u_{nn}$$

Tri-diagonales

Queremos resolver un sistema tri-diagonal (caso particular de un sistema de banda) donde la matriz es de la siguiente forma:

$$A = \begin{bmatrix} d_1 & s_1 & 0 & .. & 0 \\ i_2 & d_2 & s_2 & & . \\ 0 & & . & & 0 \\ . & & & . & s_{n-1} \\ 0 & . & 0 & i_n & d_n \end{bmatrix}$$

Tri-diagonales

Al hacer una factorización LU

$$L = \begin{bmatrix} 1 & 0 & 0 & .. & 0 \\ \alpha_2 & 1 & 0 & & . \\ 0 & & . & & 0 \\ . & & & . & 0 \\ 0 & . & 0 & \alpha_n & 1 \end{bmatrix} \qquad U = \begin{bmatrix} \beta_1 & s_1 & 0 & .. & 0 \\ 0 & \beta_2 & s_2 & & . \\ 0 & & . & & 0 \\ . & & & . & s_{n-1} \\ 0 & . & 0 & 0 & \beta_n \end{bmatrix}$$

Si multiplicamos L*U e igualamos a la matriz A
podemos deducir el algoritmo

Tri-diagonales

$$LU = \begin{bmatrix} \beta_1 & s_1 & 0 & .. & 0 \\ \alpha_2\beta_1 & \alpha_2 s_1 + \beta_2 & s_2 & & . \\ 0 & & . & & 0 \\ . & & & . & s_{n-1} \\ 0 & . & 0 & \alpha_n \beta_{n-1} & \alpha_n s_{n-1} + \beta_n \end{bmatrix} \qquad A = \begin{bmatrix} d_1 & s_1 & 0 & .. & 0 \\ i_2 & d_2 & s_2 & & . \\ 0 & & . & & 0 \\ . & & & . & s_{n-1} \\ 0 & . & 0 & i_n & d_n \end{bmatrix}$$

Tri-diagonales

$$\left.\begin{aligned}\beta_1 &= d_1\\ \alpha_j &= \frac{i_j}{\beta_{j-1}}\\ \beta_j &= d_j - \alpha_j s_{j-1}\end{aligned}\right\} para \qquad j = 2,3..n$$