Informática II

Clase 1: Resolución de sistemas de ecuaciones lineales

Mario Merino Martínez
mario.merino@upm.es

Escuela de Ingeniería Aeronáutica y del Espacio
Universidad Politécnica de Madrid

5 de febrero de 2013

Informática II

- **Profesor**: Mario Merino (mario.merino@upm.es)
- **Tutorías**: Miércoles de 11:00 a 13:00 y de 15:00 a 17:00
- El segundo cuatrimestre consta de 6 clases teóricas, 6 clases prácticas no evaluables, y 3 prácticas evaluables.
- Las clases presenciales son solo una **pequeña parte de la asignatura**; el punto central es el **trabajo individual**
  - Antes de cada clase: leer el tema, juntar dudas para preguntar
  - Después: hacer los problemas para casa, ir a tutorías
- Prestad atención a los avisos del moodle (incluye normas de evaluación, y el planning diario).
- Estas **transparencias** están disponibles en web.fmetsia.upm.es/mario

¿Qué es el Cálculo Numérico?

- **Resolver problemas complejos** (normalmente de forma *aproximada*, y haciendo uso de todo tipo de trucos a nuestra disposición), inabordables analíticamente
- **Diseñar e implementar algoritmos** para resolver dichos problemas en un ordenador, de forma **eficiente**, haciendo uso **mínimo de recursos** (RAM), y de la forma **más rápida posible**
- **Comprender los errores numéricos** (truncación, redondeo...) que se cometen al resolver, y encontrar maneras de **reducirlos** lo más posible.

Es una herramienta importantísima en la **ingeniería actual**.

Ejemplos de Cálculo Numérico

Ejemplos prácticos donde el CN se hace necesario:

- Calcular integrales

$$\int \arccos\left(1-\sqrt{x}\right) / \left[1 - x^2 \exp\left(-x^2\right)\right] dx$$

- Resolver ecuaciones

$$x \arctan x = x - 2$$

- Integrar ecuaciones diferenciales (ordinarias/parciales)

$$du/dt = t\sqrt{u^2 + \sin^2 t}$$

- Hallar autovalores, autovectores de matrices **gigantes**

$$A\boldsymbol{x} = \lambda \boldsymbol{x}$$

- Resolver sistemas de ecuaciones lineales **gigantes**

$$AX = B$$

Nota aclarativa sobre Fortran

- El **cálculo numérico** no asume un lenguaje de programación concreto: podríamos hacer todo en C++, Fortran, Matlab, Python...
- No obstante, **Fortran** está muy indicado para el cálculo numérico: es rápido y está preparado para trabajar con matrices.

Ya sabéis programar en **Fortran básico**. Lo usaremos **intensivamente** en este cuatrimestre.

- Es imprescindible que empecéis sin dudas sobre el funcionamiento de Fortran. **¡Venid a tutorías!**
- En esta parte de la asignatura (casi) no aprenderemos más Fortran. **Utilizaréis lo que ya conocéis para resolver problemas reales**.

Sistemas de Ecuaciones Lineales

Los **sistemas de ecuaciones lineales** $AX = B$ aparecen constantemente en la ingeniería

Ejemplo: la deformación $\boldsymbol{\delta}$ *de una estructura con matriz de rigidez* K *sometida a unas fuerzas* $\boldsymbol{F}$ *está dada por* $K\delta = F$

Sea A una matriz cuadrada regular de dimensión $n \gg 1$.

¿Cómo resolvemos $AX = B$? ¿Cuánto tardamos?

- ¿**Invirtiendo** la matriz? $t = O\left(n^3\right)$
- ¿Reduciéndolo a sistema triangular (**Gauss**)? $t = O\left(n^3\right)$
- ¿Usando la regla de **Cramer**? $t = O\left[(n+1)!\right]$

¿Y esto realmente importa? Si $n = 1000$, Gauss tarda 1 ms, Cramer tarda 10^{2551} años (en un ordenador corriente de 1 TFLOP)

El método LU

Realmente, $O\left(n^3\right)$ es lo más rápido que sabemos resolver $AX = B$.

El método LU es básicamente Gauss. Se basa en **descomponer** la matriz A en dos matrices triangulares (una triangular inferior, otra superior), de forma:

$$A = LU$$

Esto **también tarda** $O\left(n^3\right)$, pero posee **dos ventajas** adicionales:

- Podemos aplicar una única factorización para resolver varios problemas $AX = B_1$, $AX = B_2$, etc. Resolver tarda $O\left(n^2\right)$.
- Es posible almacenar las matrices L y U de una forma **muy económica** (sobreescribiendo A), de forma que casi no se necesita memoria adicional (ni siquiera temporal) para calcularlas (esto no es posible con A^{-1}).

Primera etapa: Factorización LU

El método LU consta de dos etapas (factorización y substitución)

Para resolver $AX = B$, **primeramente factorizamos** $A = LU$ de la forma (para $n = 3$):

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} = \begin{bmatrix} l_{11} & 0 & 0 \\ l_{21} & l_{22} & 0 \\ l_{31} & l_{32} & l_{33} \end{bmatrix} \cdot \begin{bmatrix} 1 & u_{12} & u_{13} \\ 0 & 1 & u_{23} \\ 0 & 0 & 1 \end{bmatrix}$$

(nótese que hemos tomado $u_{ii} = 1$. Si no, habría $n^2 + n$ incógnitas para n^2 ecuaciones. Equivalentemente, podríamos haber elegido e.g. $l_{ii} = 1$)

Los l_{ij} y los u_{ij} **pueden calcularse sucesivamente** de forma directa: e.g. $l_{11} = a_{11}$, $u_{12} = a_{12}/l_{11}$, etc.

Segunda etapa: Substitución LU

Una vez hemos calculado L y U, tenemos que resolver

$$LUX = B$$

1. **Llamamos** $Y = UX$, y así nos queda:

$$LY = B$$

1. **Resolvemos** este sistema triangular (trivial) para obtener Y. Entonces, hallamos X **resolviendo** un segundo sistema triangular:

$$UX = Y$$

Nótese que la factorización LU depende sólo de A, y que el B concreto que tengamos **solo importa en la etapa de resolución**.

Factorización LU (I)

¿Cómo podemos calcular L y U de forma eficiente?

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} = \begin{bmatrix} l_{11} & 0 & 0 \\ l_{21} & l_{22} & 0 \\ l_{31} & l_{32} & l_{33} \end{bmatrix} \cdot \begin{bmatrix} 1 & u_{12} & u_{13} \\ 0 & 1 & u_{23} \\ 0 & 0 & 1 \end{bmatrix}$$

Multiplicando la primera columna de U por L obtenemos cuánto vale la primera columna de L:

$$l_{11} = a_{11}; \quad l_{21} = a_{21}; \quad l_{31} = a_{31}$$

Factorización LU (II)

Ya tenemos l_{11}, l_{22}, l_{33} (en **negrita**)

$$\begin{bmatrix} \boldsymbol{a_{11}} & a_{12} & a_{13} \\ \boldsymbol{a_{21}} & a_{22} & a_{23} \\ \boldsymbol{a_{31}} & a_{32} & a_{33} \end{bmatrix} = \begin{bmatrix} \boldsymbol{l_{11}} & 0 & 0 \\ \boldsymbol{l_{21}} & l_{22} & 0 \\ \boldsymbol{l_{31}} & l_{32} & l_{33} \end{bmatrix} \cdot \begin{bmatrix} 1 & u_{12} & u_{13} \\ 0 & 1 & u_{23} \\ 0 & 0 & 1 \end{bmatrix}$$

Multiplicando la primera fila de L por U obtenemos cuánto vale la primera fila de U:

$$u_{12} = a_{12}/l_{11}; \quad u_{13} = a_{13}/l_{11}$$

Factorización LU (III)

$$\begin{bmatrix} \boldsymbol{a_{11}} & \boldsymbol{a_{12}} & \boldsymbol{a_{13}} \\ \boldsymbol{a_{21}} & a_{22} & a_{23} \\ \boldsymbol{a_{31}} & a_{32} & a_{33} \end{bmatrix} = \begin{bmatrix} \boldsymbol{l_{11}} & 0 & 0 \\ \boldsymbol{l_{21}} & l_{22} & 0 \\ \boldsymbol{l_{31}} & l_{32} & l_{33} \end{bmatrix} \cdot \begin{bmatrix} 1 & \boldsymbol{u_{12}} & \boldsymbol{u_{13}} \\ 0 & 1 & u_{23} \\ 0 & 0 & 1 \end{bmatrix}$$

Ahora, multiplicando la segunda columna de U por L obtenemos cuánto vale la segunda columna de L:

$$l_{22} = a_{22} - l_{21}u_{12}; \quad l_{32} = a_{32} - l_{31}u_{12}$$

Obsérvese que cada nuevo elemento l_{ij} ó u_{ij} que calculamos depende **sólo** de A y de los elementos ya **conocidos** de L y U.

Factorización LU (IV)

Continuamos del mismo modo (columna, fila, columna, fila...):

$$\begin{bmatrix} \boldsymbol{a_{11}} & \boldsymbol{a_{12}} & \boldsymbol{a_{13}} \\ \boldsymbol{a_{21}} & \boldsymbol{a_{22}} & a_{23} \\ \boldsymbol{a_{31}} & \boldsymbol{a_{32}} & a_{33} \end{bmatrix} = \begin{bmatrix} \boldsymbol{l_{11}} & 0 & 0 \\ \boldsymbol{l_{21}} & \boldsymbol{l_{22}} & 0 \\ \boldsymbol{l_{31}} & \boldsymbol{l_{32}} & l_{33} \end{bmatrix} \cdot \begin{bmatrix} 1 & \boldsymbol{u_{12}} & \boldsymbol{u_{13}} \\ 0 & 1 & u_{23} \\ 0 & 0 & 1 \end{bmatrix}$$

La segunda fila de L por U proporciona:

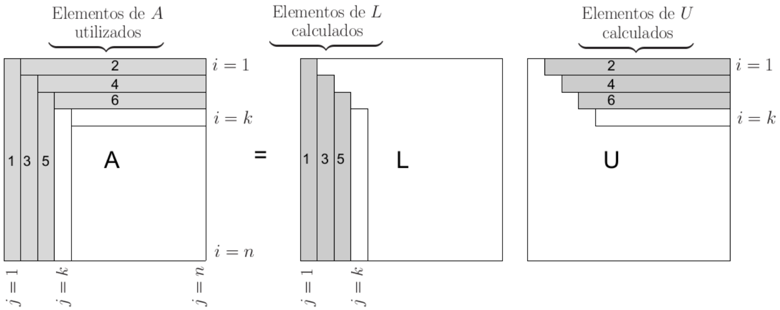
$$u_{23} = (a_{23} - l_{21} u_{13}) \, / \, l_{22}$$

La tercera columna de U por L proporciona:

$$l_{33} = a_{33} - l_{31} u_{13} - l_{32} u_{23}$$

Generalización (I)

Para mayor n, el procedimiento es idéntico: usando multiplicaciones por columnas de U y por filas de L conseguimos **resolver las dos matrices de forma directa**.



Generalización (II)

- La columna k de L es $(i = k, \ldots, n)$:

$$l_{ik} = a_{ik} - \sum_{h=1}^{k-1} l_{ih} u_{hk}$$

- La fila k de U es $(i = k+1, \ldots, n)$:

$$u_{ki} = \left(a_{ki} - \sum_{h=1}^{k-1} l_{kh} u_{hi} \right) / l_{kk}$$

El ***problema****: ¿qué pasa si algún $l_{kk} = 0$? entonces hay que acondicionar primero la matriz A, reordenando filas y/o columnas (**proceso de pivotado**) $\rightarrow$ no lo veremos aquí, pero preguntadme*

Substitución LU (I)

Una vez tenemos L y U, resolvemos $LUX = B$ en dos pasos (resolviendo $LY = B$ primero y $UX = Y$ después).

- **Primer paso**: $LY = B$:

$$\begin{bmatrix} l_{11} & 0 & 0 \\ l_{21} & l_{22} & 0 \\ l_{31} & l_{32} & l_{33} \end{bmatrix} \cdot \begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix}$$

Hallamos **de forma directa** *(i.e., las ecuaciones no están acopladas)*:

$$y_1 = b_1 / l_{11}; \quad y_2 = (b_2 - l_{21} y_1) / l_{22};$$

$$y_3 = (b_3 - l_{31} y_1 - l_{32} y_2) / l_{33}$$

Generalización:

$$y_k = \left(b_k - \sum_{h=1}^{k-1} l_{kh} y_h \right) / l_{kk}$$

Substitución LU (II)

Una vez tenemos L y U, resolvemos $LUX = B$ en dos pasos (resolviendo $LY = B$ y después $UX = Y$).

- **Segundo paso**: $UX = Y$:

$$\begin{bmatrix} 1 & u_{12} & u_{13} \\ 0 & 1 & u_{23} \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix}$$

De donde (resolviendo en orden $n, n-1, n-2, \ldots, 1$):

$$x_3 = y_3; \quad x_2 = y_2 - u_{23}x_3; \quad x_1 = y_1 - u_{12}x_2 - u_{13}x_3$$

Generalización:

$$x_k = y_k - \sum_{h=k+1}^{n} u_{kh}x_h$$

Almacenaje de L y U sobre A

- **Interesantemente**, *una vez utilizado un a_{ij} para calcular el correspondiente l_{ij} ó u_{ij}, nunca más vuelve a ser utilizado*.

- Puesto que los "0" y "1" de las matrices L, U son siempre los mismos, esta propiedad nos permite **guardar las matrices L y U a medida que las vamos calculando**, de forma muy compacta y económica, **sobreescribiendo** la matriz A:

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \rightarrow \begin{bmatrix} l_{11} & u_{12} & u_{13} \\ l_{21} & l_{22} & u_{23} \\ l_{31} & l_{32} & l_{33} \end{bmatrix}$$

- **Del mismo modo**, podemos sobreescribir B con Y, y luego con X.

Resumen

- El **método LU** es una manera rápida y eficaz de resolver $AX = B$ (con A cuadrada regular).
- Consta de **dos etapas**: factorización ($A = LU$) y substitución ($LY = B$, $UX = Y$)
- La **factorización**, $t = O\left(n^3\right)$, puede hacerse sucesivamente de forma directa (ecuaciones no acopladas)
- La **substitución**, $t = O\left(n^2\right)$, también es directa (dos sistemas triangulares). Podemos resolver para varios B
- Si no nos importa perder la matriz A, **las matrices L y U pueden irse guardando encima de A según se calculan**, ya que una vez utilizado un elemento de A, no vuelve a hacer falta (y también Y, X encima de B).