Informática II

Clase 2: Aplicaciones LU: determinante e inversa
Ecuaciones no lineales: Método de Newton

Mario Merino Martínez
mario.merino@upm.es

Escuela de Ingeniería Aeronáutica y del Espacio
Universidad Politécnica de Madrid

19 de febrero de 2013

Recordatorio método LU

Para resolver $AX = B$ **primero** factorizamos $A = LU$:

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

y **después** resolvemos consecutivamente dos problemas triangulares ($LUX = B$):

$$LY = B$$

$$UX = Y$$

Cálculo de Determinantes

Si ya **tenemos factorizada la matriz** A en la forma $A = LU$, obtener su determinante es trivial:

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

- Por ser L, U matrices triangulares, su determinante es el **producto de los elementos de la diagonal**.
- $\det L = \prod_{k=1}^{n} l_{kk}$ y $\det U = 1$, por lo que, sencillamente:

$$\det A = l_{11} l_{22} \cdots l_{nn} = \prod_{k=1}^{n} l_{kk}$$

Esto puede realizarse dentro de una **función** que acepte A, n, factorice, y devuelva $\det A$

Cálculo de la matriz inversa I

Cuando calculamos la **matriz inversa de** A, buscamos una matriz $Z = A^{-1}$ que cumpla

$$\begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & & a_{2n} \\ \vdots & & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{bmatrix} \cdot \begin{bmatrix} & & & \\ & Z & & \\ & & & \end{bmatrix} = \begin{bmatrix} 1 & 0 & \cdots & 0 \\ 0 & 1 & & 0 \\ \vdots & & \ddots & \vdots \\ 0 & 0 & \cdots & 1 \end{bmatrix}$$

La siguiente observación es la clave para usar el **método LU en el cálculo de la inversa**:

- La columna k de la matriz identidad I resulta sólo de **multiplicar la columna k de Z por la matriz A al completo**

Cálculo de la matriz inversa II

Por ejemplo, para $n = 3$, podemos calcular **la primera columna de** $Z = A^{-1}$ así:

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

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

$\Rightarrow$ Podemos calcular A^{-1} columna a columna, **resolviendo** n **problemas del tipo** $AX = B$

Implementación de la inversa

Tenemos que **resolver** n **veces** $AX = B$, variando $B = [0, \ldots, 1, \ldots, 0]^T$. Hemos de guardar la solución de cada problema en la correspondiente columna de $Z = A^{-1}$.

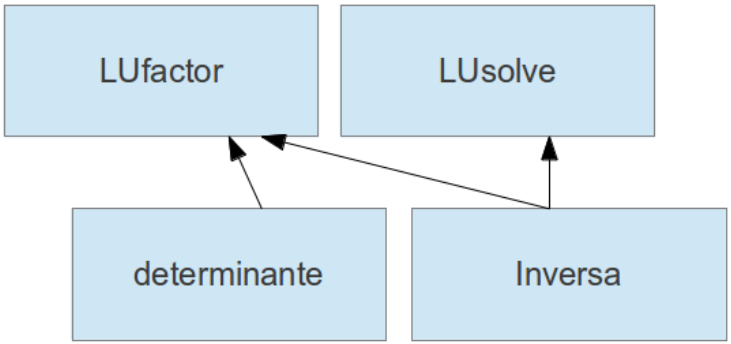
1. Primero de todo, factorizamos A: `call factor(A,n)`
1. Creamos una matriz Z del mismo tamaño, y la **inicializamos** a la matriz identidad I

```
Z = 0d0 ! Hacemos todo Z_ij=0
do k=1,n
  Z(k,k) = 1d0 ! Así cada columna de Z sirve de vector B
end do
```

1. En un **bucle**, llamamos a solve con A factorizada **y la columna** k **de** Z

```
do k=1,n
  call solve(A,Z(1:n,k:k),n) ! Sobre Z se almacena A^-1
end do
```

Visión general herramientas desarrolladas



Cada **programa que desarrolláis** añade funcionalidad a vuestra "caja de herramientas numéricas"

- Cada subrutina lleva a cabo **una única tarea bien definida**, con unos **inputs y outputs bien definidos**.
- Nos **apoyamos** en las subrutinas que ya tenemos para **desarrollar otras** (i.e., pueden depender unas de otras)

Introducción al Método de Newton

A menudo **no podemos resolver una ecuación** $f(x) = 0$ **de forma analítica**. Pero sabemos más o menos (e.g. gracias a la gráfica de f, o modelos simplificados) dónde se encuentran las raíces.

- El método de **Newton-Raphson** (y otros similares) toma una aproximación inicial x_0 a una raíz $x = \alpha$ y **la va mejorando progresivamente**, con tanta precisión como queramos.

- Si todo va bien, el método **converge** y las distintas aproximaciones forman una sucesión $\{x_n\}$ tal que $x_n \to \alpha$ cuando $n \to \infty$.

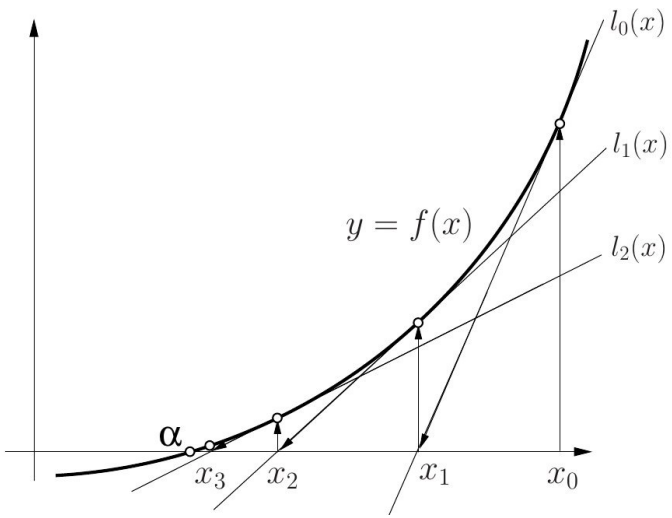
Método de Newton I

Conocida la función f y su derivada f', intentamos hallar una raíz $x = \alpha$ de $f(x) = 0$ a partir de una **aproximación inicial** x_0:

- Asumimos que f es "casi lineal" en torno a x_0, y **aproximamos f por su recta tangente en ese punto**: $\tilde{f} = f(x_0) + f'(x_0)(x - x_0)$.
- La **recta tangente corta al eje** $(\tilde{f}(x) = 0)$ en $x = x_0 - f(x_0)/f'(x_0)$. Como $\tilde{f}$ no es realmente f, esta x no es la raíz buscada. Pero esperablemente, *estaremos más cerca de ella*.

Método de Newton II

- **Repetimos la operación**: linealizamos f con la recta tangente **en el nuevo punto hallado**, hacemos $\tilde{f} = 0$, obtenemos una nueva x, y así sucesivamente.



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

- La **convergencia del método** depende de la forma de la función en sí y de la aproximación inicial x_0

Método de Newton III

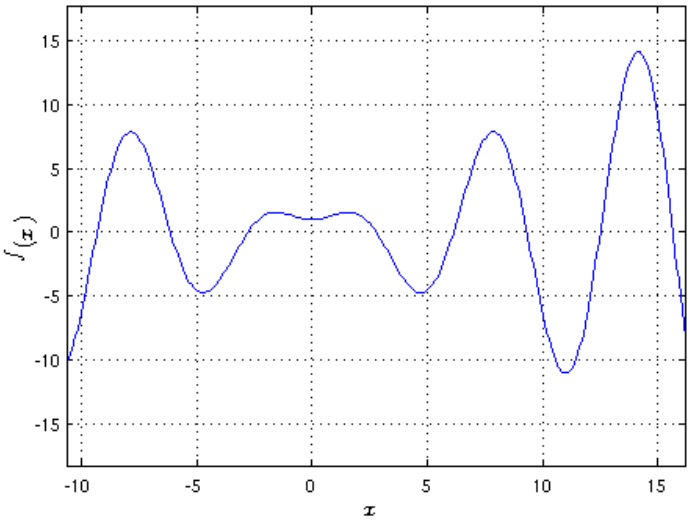
- **¿Hasta cuando seguimos calculando?** Fijamos un **criterio de parada** que comprobamos en cada paso. *Por ejemplo:*

$$\frac{|\alpha - x_i|}{|\alpha|} \simeq \frac{|x_{i+1} - x_i|}{|x_{i+1}|} < \varepsilon \quad y \quad |f(x_{i+1})| < \varepsilon$$

  para un cierto $\varepsilon > 0$ elegido de antemano. Además, hay que **fijar un número máximo de iteraciones**, para que si el método **no converge** se termine el bucle y se muestre un mensaje de error.

- Cuando **se verifica el criterio de parada**, damos por terminado el cálculo de la raíz (4 ó 5 iteraciones suelen bastar).

Ejemplo de Implementación




```
program newton ! Ex. 203
implicit none
integer :: i, nmax=10
real*8 :: x, dx, f, df, eps = 1d-8
x = 1d0 ! Aprox. inicial
do i=1,nmax
  f = x*sin(x)+cos(x) ! funcion
  df = x*cos(x) ! Su derivada
  dx = -f/df ! La corrección de x
  x = x + dx ! Nuevo valor de x
  if (abs(dx/x)<eps .and. abs(f)<eps) then ! Criterio parada
    print*, 'solucion: ', x, ' en iteracion: ', i
    exit ! Sale del bucle
  end if
end do
if (i>nmax) print*, 'error: no convergencia'
end program newton
```

Convergencia cuadrática del método

En torno a la raíz $x = \alpha$, para $e = x - \alpha$ **pequeño**, $f(x)$ puede aproximarse como:

$$f(\alpha + e) = 0 + f'(\alpha)\, e + \frac{f''(\alpha)}{2} e^2 + \ldots$$

Puesto que $x_n = x_{n-1} - f_{n-1}/f'_{n-1}$, el error en la n-iteración, $e_n = \alpha - x_n$, puede escribirse **en función de** e_{n-1} como

$$e_n \simeq e_{n-1} - \frac{f'(\alpha)\, e_{n-1} + f''(\alpha)\, e_{n-1}^2/2}{f'(\alpha) + f''(\alpha)\, e_{n-1}} = \frac{f''(\alpha)}{2f'(\alpha)} e_{n-1}^2 + \mathcal{O}\left(e_{n-1}^3\right)$$

Es decir, si $f'(\alpha) \neq 0$, entonces $e_n \simeq C e_{n-1}^2$: ¡el tamaño del **error se reduce de forma cuadrática** en cada iteración!

Observaciones

- Conviene definir f y f' como dos `function`, y convertir programa `newton` **en subrutina** (así tenéis una herramienta más para vuestra colección)
- Si $f'(\alpha) = 0$ (e.g. raíces múltiples), la **convergencia es peor que cuadrática** (o **falla** directamente) . Este caso puede requerir un tratamiento especial
- Podéis **mostrar por pantalla el error en cada iteración** ($\simeq x_n - x_{n-1}$). Si no es cuadrático, habéis programado algo mal. Casi siempre es la derivada
- La **condición inicial,** x_0, ha de elegirse con cuidado: la física del problema, modelos simplificados, etc. ayudan
- Cuando el problema a resolver es $f(x) = g(x)$, simplemente haced $h(x) \equiv f(x) - g(x) = 0$