Máster Universitario en Computación Paralela y Distribuida
Algoritmos Paralelos Matriciales en Ingeniería

Tema 1.

Condicionamiento y estabilidad numérica

Bibliografía:

“Matrix Computations”. G.Golub & C. Van Loan. Baltimore ; London : Johns Hopkins University Press, 1996 (u otra edición del libro)

Lecturas recomendadas:

“Matrix Computations”. G.Golub & C. Van Loan.

Capítulo 1. Puntos 1.1, 1.2 y 1.3

Capítulo 2. Puntos 2.1, 2.2, 2.3, 2.4 y 2.7

Estabilidad Numérica

Problemas científicos y técnicos se resuelven en base a una función f que asocia a un elemento del conjunto de datos un elemento del conjunto de resultados.

Algoritmo: Conjunto finito de pasos que implementa la resolución del problema representado por f.

Problema
f:*D*------>*R*

Algoritmo
f*:*D*------>*R*

Para un mismo problema pueden existir diferentes algoritmos.

Ejemplo: Ecuación de 2º grado
Algoritmo 1: $x=(-b\pm(b^2-4ac)^{(1/2)})/(2a)$

Algoritmo 2: $x_1=(-b-sg(b)(b^2-4ac)^{(1/2)})/(2a)$ y $x_2=c/(ax_1)$

Algoritmo 1 proporciona resultados incorrectos: Algoritmo inestable
Algoritmo 2 proporciona resultados correctos: Algoritmo estable

Formas de analizar el comportamiento de los algoritmos en cuanto a su estabilidad

*Analizar como está de lejos la solución computada de la solución exacta:

Análisis de error progresivo o FORWARD ERROR ANALYSIS

Mejor si se conocen los datos exactos

*Analizar si la solución computada es la solución exacta de un problema próximo al problema original:

Análisis de error progresivo o BACKWARD ERROR ANALYSIS

Mejor si se conocen datos experimentales o se utiliza una representación que provoca errores

DEFINICIONES DE ESTABILIDAD

-Un algoritmo es progresivamente (forward) estable si la solución computada que produce está próxima a la solución exacta.

Ejemplo: $fl(x+y)= (x+y)(1+\delta)$, con $|\delta| \leq u$

es decir $fl(x+y)= (x+y) + e$,

con $e = (\delta)(x+y)$ pequeño

- Un algoritmo es regresivamente (backward) estable si la solución que produce es la solución exacta de un problema próximo al original.

Ejemplo: $fl(x+y)= (x+y)(1+\delta)$, con $|\delta| \leq u$

es decir $fl(x+y)= x(1+\delta)+y(1+\delta)=x'+y'$

con (x' próximo a x) e (y' próximo a y)

PROXIMIDAD

•Tradicionalmente se han venido utilizando normas vectoriales y matriciales para medir la proximidad entre vectores y entre matrices

•Este sistema tiene el inconveniente de que se mide una matriz entera (n^2 números) utilizando un solo número

•También se puede utilizar un análisis de proximidad orientado a elementos

Ejemplo:

$A, B \in \Re^{nxn}$

$fl(A+B) = [fl(a_{ij} + b_{ij})] = [(a_{ij} + b_{ij})(1+\delta_{ij})]$ con $|\delta_{ij}| \leq u$, i,j=1,2,...n

$fl(A+B) = [(a_{ij} + b_{ij})] + [(a_{ij} + b_{ij})\delta_{ij}] = (A+B) + E$

con $|\varepsilon_{ij}| \leq |(a_{ij} + b_{ij})| u$, i,j=1,2,...n, orientado a elementos

con $\|E\|_F$ pequeña, orientado a normas

EJEMPLO DE ALGORITMO NUMERICAMENTE INESTABLE:

Eliminación Gaussiana sin pivotamiento

Dado (A+E)x'=b, con $A,E \in \Re^{nxn}$ se puede demostrar que

$$\|E\|_\infty \leq c.n^3.r \|A\|_\infty \ u$$

Con u=precisión de la máquina y $r=(\max_k \max_{i,j} |a_{ij}^{(k)}|) / (\max_{i,j} |a_{ij}|)$

Si A= $\begin{bmatrix} 10^{-10} & 1 \\ 1 & 2 \end{bmatrix}$ ----> $A^{(2)} = \begin{bmatrix} 10^{-10} & 1 \\ 0 & 2-10^{10} \end{bmatrix}$

y $r = (10^{10})/2$

Esto no ocurre para todo $A \in \Re^{nxn}$

DEFINICION

Un algoritmo es regresivamente estable para una clase de matrices si para cada matriz de esa clase la solución computada es la solución exacta de un problema próximo

Condicionamiento de un problema

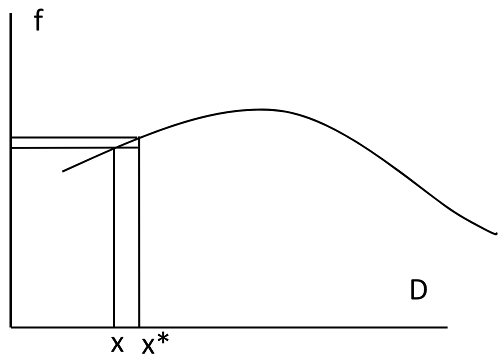
- Problemas científicos y técnicos se resuelven en base a una función f que asocia a un elemento del conjunto de datos un elemento del conjunto de resultados.

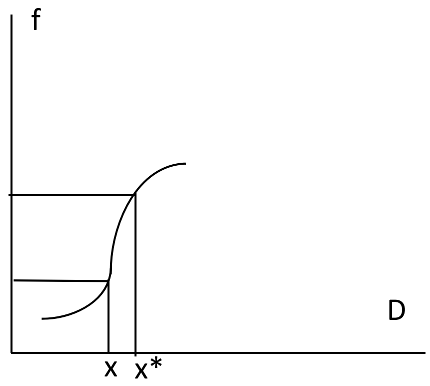
  Problema $\qquad$ f:*D*------>*R*

- Si se utiliza una representación no exacta de los datos y se quiere obtener un resultado preciso debemos asegurarnos de que la función f no produce grandes variaciones.

- Si $x \in D$ se aproxima por $x^* \in D$ ¿qué ocurre con $f(x^*)$? ¿Es próximo a $f(x)$?

Condicionamiento de un problema





Obsérvese que esta cuestión atañe al problema que resuelve f y no al algoritmo que implementemos.

DEFINICION

- Un **<u>problema está mal condicionado</u>** (con respecto a un conjunto de datos) si un error relativo pequeño en los datos provoca un error relativo grande en la solución computada, independientemente del algoritmo utilizado para encontrarla

<u>Ejemplo de problema mal condicionado:</u>

Polinomio de Wilkinson: $P(x)=(x-1)(x-2)...(x-20)=x^{20} - 210x^{19} +....$

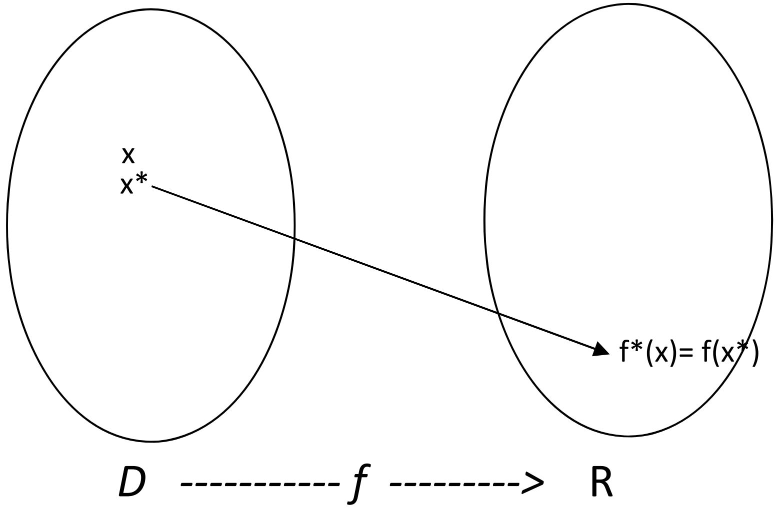
Raíces: 1,2,...,20

Si se cambia el coeficiente de x^{19} por $-210+2^{-23}$ aparecen raíces tales como:

$19.502439400 \pm 1.940330347\, i$, o $16.730737466 \pm 2.812624894\, i$, etc

Precisión de los algoritmos numéricos

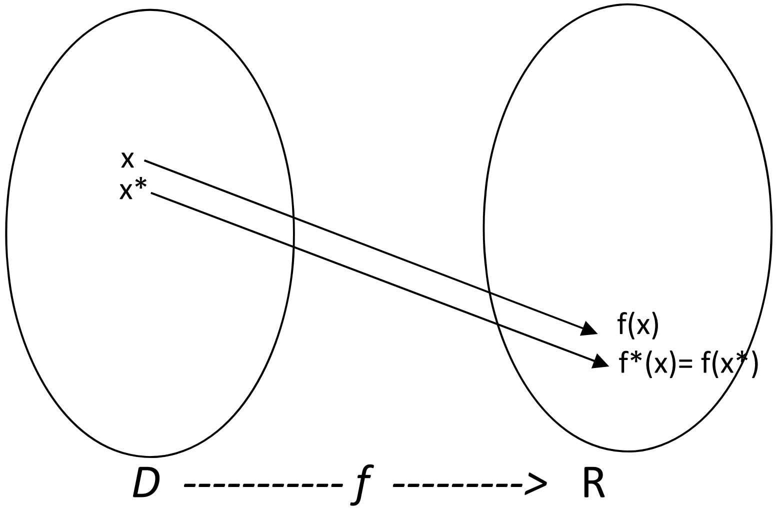
Un algoritmo backward estable no garantiza una solución precisa



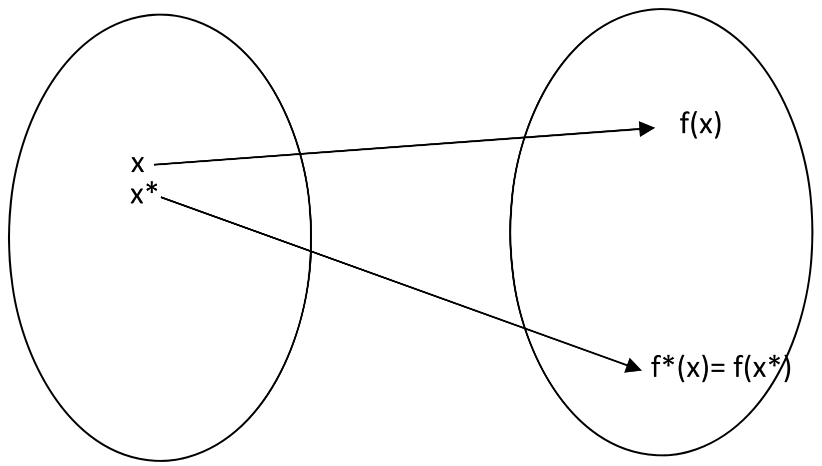
Para obtener una solución precisa deben darse dos condiciones:

-Usar un algoritmo backward estable

-Tener un problema bien condicionado

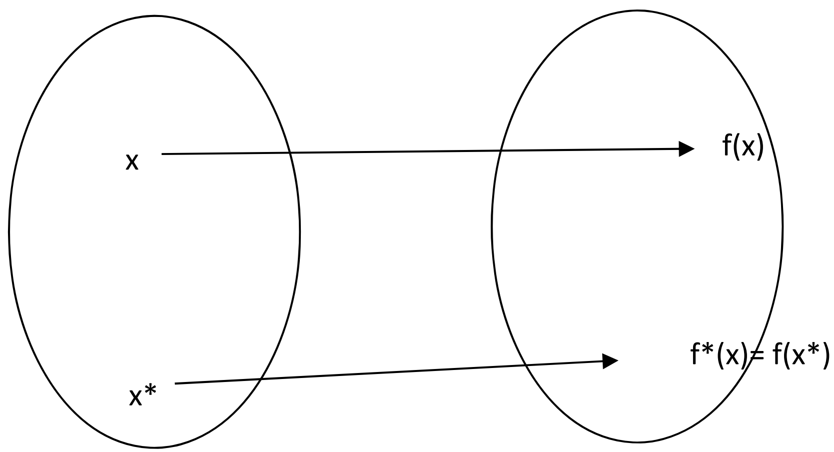


Si el problema está mal condicionado:



D -----------f ---------> R

Si el algoritmo no es estable



D -----------f ---------> R

DEFINICION

- **<u>Numero de condición de un problema:</u>** Indica si el problema está mal o bien condicionado. El número de condición de un problema proporciona una cota superior para el error relativo en la solución computada cuando se produce una pequeña perturbación en los datos de entrada.

- **<u>Definición genérica:</u>**

Numero de condición del problema f

$$\kappa(f) = \max_{x,x^*} \frac{\left\| f(x) - f(x^*) \right\|}{\left\| x - x^* \right\|}$$

Número de condición de la Resolución de un Sistema de Ecuaciones

$$A, F \in \Re^{nxn}, b, f \in \Re^{n}$$

Sistema exacto — Sistema perturbado

$$Ax = b \quad \rightarrow \quad (A + \varepsilon F)x(\varepsilon) = b + \varepsilon f$$

Si desarrollamos x(ε) en serie de Taylor alrededor del origen se tiene:

$$x(\varepsilon) = x(0) + \varepsilon . \dot{x}(0) + O(\varepsilon^2)$$

Para calcular $\dot{x}(0)$ derivamos la igualdad del sistema perturbado con respecto a ε

$$Fx(\varepsilon) + (A + \varepsilon F)\dot{x}(\varepsilon) = f$$

En ε=0 se tiene

$$Fx(0) + (A)\dot{x}(0) = f$$

O bien, teniendo en cuenta que x(0)=x

$$\dot{x}(0) = A^{-1}(f - Fx)$$

y

$$x(\varepsilon) \approx x(0) + \varepsilon . A^{-1}(f - Fx) \rightarrow x(\varepsilon) - x(0) \approx \varepsilon . A^{-1}(f - Fx)$$

Por tanto

$$\left\| x(\varepsilon) - x(0) \right\|_p \approx |\varepsilon| . \left\| A^{-1}(f - Fx) \right\|_p \leq |\varepsilon| . \left\| A^{-1} \right\|_p \left\| (f - Fx) \right\|_p$$

Entonces

$$\frac{\left\| x(\varepsilon) - x \right\|_p}{\left\| x \right\|_p} \leq |\varepsilon| . \frac{\left\| A^{-1} \right\|_p \left\| (f - Fx) \right\|_p}{\left\| x \right\|_p}$$

O bien

$$E_r \leq |\varepsilon| . \frac{\left\| A^{-1} \right\|_p (\| f \|_p + \| F \|_p \| x \|_p)}{\left\| x \right\|_p}$$

o bien

$$E_r \le |\varepsilon| \left\| A^{-1} \right\|_p \left(\frac{\|f\|_p}{\|x\|_p} + \|F\|_p \right) \left(\frac{\|A\|_p}{\|A\|_p} \right)$$

es decir

$$E_r \le \left\| A^{-1} \right\|_p \|A\|_p \left(|\varepsilon| \frac{\|f\|_p}{\|A\|_p \|x\|_p} + |\varepsilon| \frac{\|F\|_p}{\|A\|_p} \right)$$

Puesto que

$$Ax = b \Rightarrow \|Ax\|_p = \|b\|_p \Rightarrow \|b\|_p \le \|A\|_p \|x\|_p \Rightarrow \frac{1}{\|A\|_p \|x\|_p} \le \frac{1}{\|b\|_p}$$

se tiene

$$|\varepsilon| \frac{\|f\|_p}{\|A\|_p \|x\|_p} \le |\varepsilon| \frac{\|f\|_p}{\|b\|_p}$$

y por tanto

$$E_r \le \left\| A^{-1} \right\|_p \|A\|_p \left(|\varepsilon| \frac{\|f\|_p}{\|b\|_p} + |\varepsilon| \frac{\|F\|_p}{\|A\|_p} \right)$$

Si se denomina

$$\kappa_p = \left\| A^{-1} \right\|_p \left\| A \right\|_p \qquad E_{rA} = |\varepsilon| \frac{\|F\|_p}{\|A\|_p} \qquad E_{rb} = |\varepsilon| \frac{\|f\|_p}{\|b\|_p}$$

se tiene

$$E_r \leq \kappa_p (E_{rA} + E_{rb})$$

siendo

$$\kappa_p = \left\| A^{-1} \right\|_p \left\| A \right\|_p$$

el p-número de condición del problema

Número de condición en Matlab

```
>> A=rand(100);
>> cond(A)
ans =
   3.937296076830096e+003
>> cond(A,1)
ans =
   9.290236187527715e+003
>> cond(A,2)
ans =
   3.937296076830096e+003
>> cond(A,inf)
ans =
   1.038151127140350e+004
>> rcond(A)
ans =
   1.076398898601218e-004
```

Número de condición en Matlab

```
>> A=hilb(100);

>> cond(A)
ans =
   6.595325688333352e+019

>> rcond(A)
ans =
   1.067235828659013e-020

>> cond(A,2)
ans =
   6.595325688333352e+019
>>
```

```
>> cond(A,1)
Warning: Matrix is close to
singular or badly scaled.
         Results may be inaccurate.
RCOND = 2.144574e-021.
> In cond at 47
ans =
   4.662929869643412e+020

>> cond(A,inf)
Warning: Matrix is close to
singular or badly scaled.
         Results may be inaccurate.
RCOND = 2.144574e-021.
> In cond at 47
ans =
   4.662929869643412e+020
>>
```