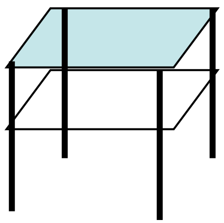
Tema 0

Modelado de problemas en Ingeniería. Problemas lineales y no lineales. Problemas matriciales en Ingeniería

Metodología de diseño en Ingeniería

Problema físico → Modelo físico → Modelo matemático →

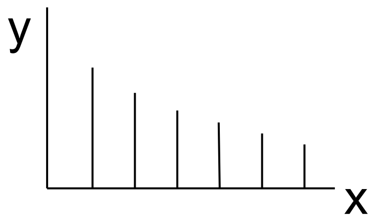


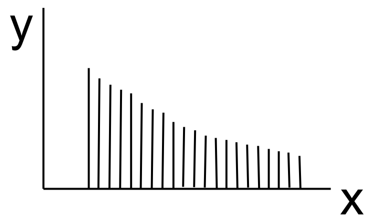
$$\sum_i f_i = 0$$

$$\frac{dy^2}{dx^2} + ky = 0$$

→ Solución formal → Solución numérica → Solución computacional

$$y = K_1 e^{-\lambda x}$$





Objetivos del Ingeniero Informático especialista en Ciencias Computacionales

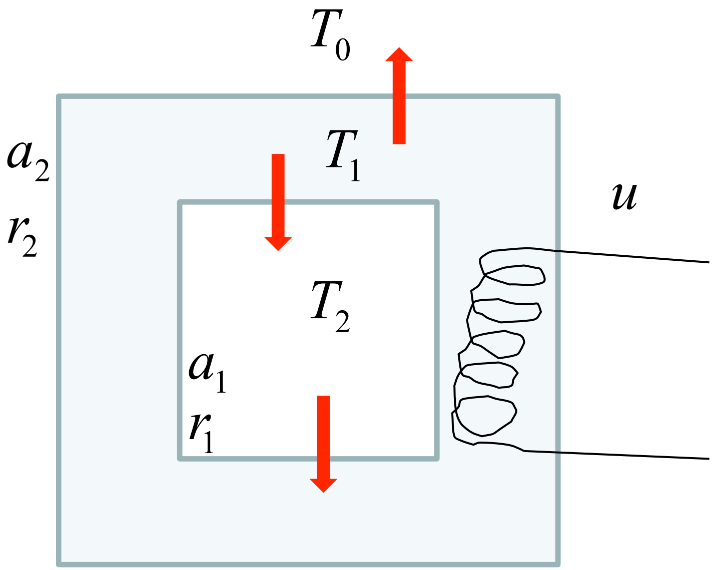
- Expresar los modelos matemáticos en forma adecuada para ser tratados eficientemente con un computador:
  - Computación matricial
- Desarrollar herramientas físicas y lógicas que permitan resolver los problemas computacionales planteados:
  - Computación paralela
  - Librerías numéricas y matriciales de altas prestaciones
- Ser capaz de resolver los problemas planteados utilizando las herramientas de altas prestaciones:

  Diseño e implementación de algoritmos paralelos eficientes basados en las herramientas de altas prestaciones

Casos de estudio

CASO 1

Control de la temperatura interior de un horno mediante una entrada de calor



Se puede modelar el sistema de la siguiente forma

- Sean a_1 y a_2 las superficies interior y exterior
- Sean c_1 y c_2 la capacidad calorífica del interior del horno y de la camisa
- Sean r_1 y r_2 los coeficientes de radiación de las superficies interior y exterior

Entonces

$$c_1 \dot{T}_1 = -a_2 r_2 (T_1 - T_0) - a_1 r_1 (T_1 - T_2) + u$$

$$c_2 \dot{T}_2 = -a_1 r_1 (T_1 - T_2)$$

O bien

$$c_1 \dot{T}_1 = -a_2 r_2 (T_1 - T_0) - a_1 r_1 (T_1 - T_0 + T_0 - T_2) + u$$

$$c_2 \dot{T}_2 = -a_1 r_1 (T_1 - T_0 + T_0 - T_2)$$

Reorganizando términos

$$\dot{T}_1 = \frac{-(a_2 r_2 + a_1 r_1)}{c_1}(T_1 - T_0) + \frac{(a_1 r_1)}{c_1}(T_2 - T_0) + \left(\frac{1}{c_1}\right) u$$

$$\dot{T}_2 = \frac{a_1 r_1}{c_2}(T_1 - T_0) - \frac{a_1 r_1}{c_2}(T_2 - T_0)$$

O en forma matricial
$$\begin{bmatrix} \dot{T}_1 \\ \dot{T}_2 \end{bmatrix} = \begin{bmatrix} \dfrac{-(a_2 r_2 + a_1 r_1)}{c_1} & \dfrac{a_1 r_1}{c_1} \\ \dfrac{a_1 r_1}{c_2} & \dfrac{-a_1 r_1}{c_2} \end{bmatrix} \begin{bmatrix} T_1 - T_0 \\ T_2 - T_0 \end{bmatrix} + \begin{bmatrix} 1/c_1 \\ 0 \end{bmatrix} u$$

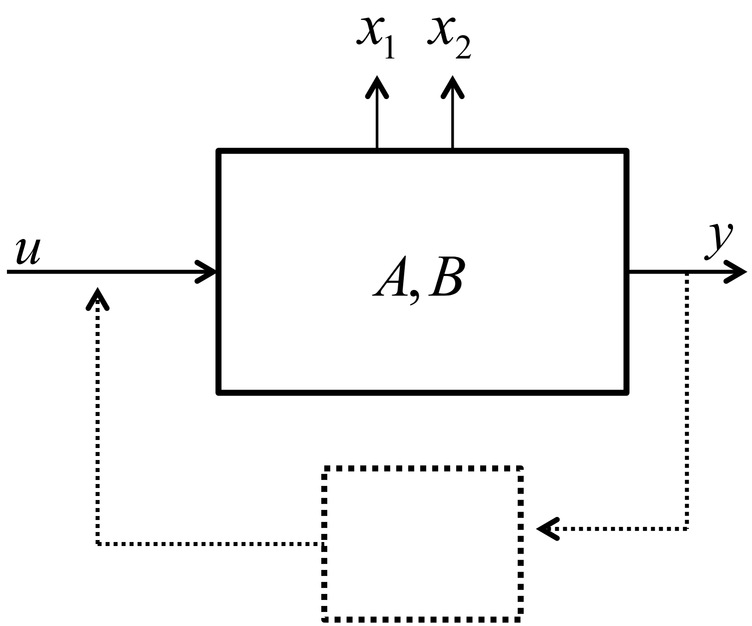
Si

$$x_1 = T_1 - T_0 \quad , \quad x_2 = T_2 - T_0 \quad , \qquad A = \begin{bmatrix} \dfrac{-(a_2 r_2 + a_1 r_1)}{c_1} & \dfrac{a_1 r_1}{c_1} \\ \dfrac{a_1 r_1}{c_2} & \dfrac{-a_1 r_1}{c_2} \end{bmatrix} \quad \text{y} \quad B = \begin{bmatrix} 1/c_1 \\ 0 \end{bmatrix}$$

Se tiene

$$\dot{x}(t) = Ax(t) + Bu$$

$$\text{con } A \in \Re^{n\times n}, B \in \Re^{n\times k}, n = 2, k = 1$$



con $y = x_1 + T_0$

Sistema lineal de control
en el modelo de espacio
de estados

Caso homogéneo

$$\dot{x}(t) = Ax(t)$$

$$\begin{cases} \dfrac{dx_1}{dt} = a_{11}x_1 + a_{12}x_2 + \ldots + a_{1n}x_n \\ \dfrac{dx_2}{dt} = a_{21}x_1 + a_{22}x_2 + \ldots + a_{2n}x_n \\ \ldots\ldots\ldots\ldots\ldots\ldots \\ \dfrac{dx_n}{dt} = a_{n1}x_1 + a_{n2}x_2 + \ldots + a_{nn}x_n \end{cases}$$

Solución del Caso homogéneo

Si se ensaya como solución $x(t) = e^{\lambda t} v$ siendo $v \in \Re^n$ o $v \in C^n$ se cumple $\dot{x}(t) = \lambda e^{\lambda t} v$

y por tanto $\lambda e^{\lambda t} v = A e^{\lambda t} v$

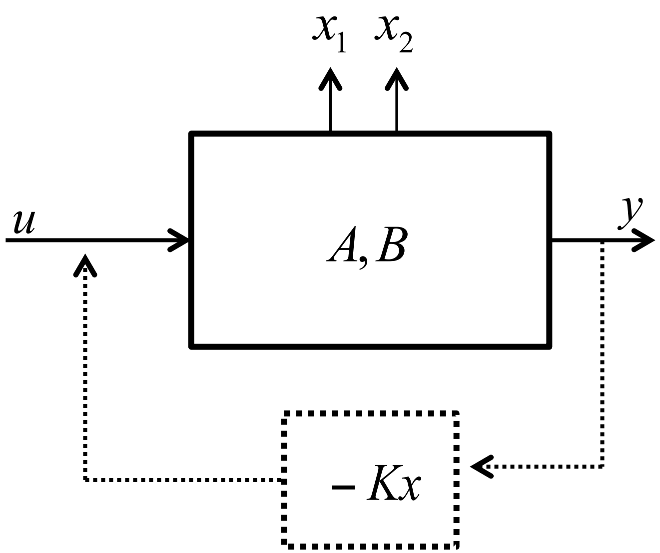
o bien simplificando $Av = \lambda v$

Es decir la solución depende de los valores propios de A

Además, ciertas propiedades del sistema dependen de los valores propios de A, por ejemplo la estabilidad:

$$\mathrm{Re}(\lambda) > 0 \Rightarrow Inestabilidad$$

Realimentación: Sistema en lazo cerrado



$$\dot{x} = Ax + B(u - Kx)$$

$$\boxed{\dot{x} = (A - BK)x + Bu}$$

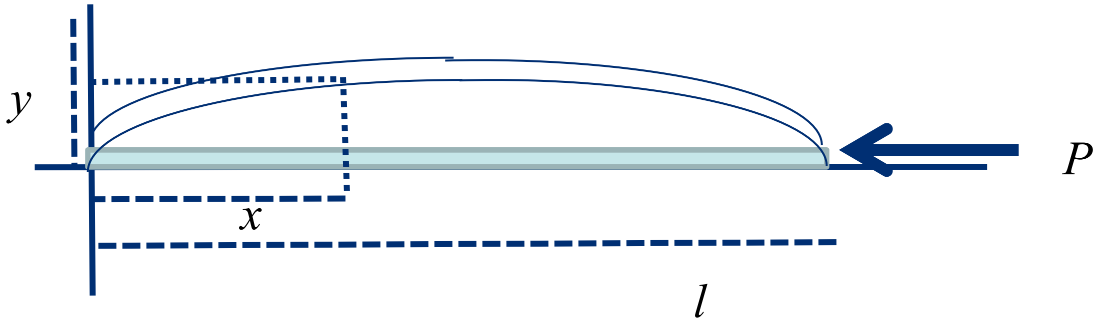
Problema de asignación de polos

Determinar *K* con la condición de que los valores propios de *A-BK* tengan parte real negativa

CASO 2

Flexión de una barra rígida

- Consideremos una barra rígida de longitud *l* y sección constante *a*, que soporta una fuerza *P* en un extremos y no se le permite el movimiento vertical en los extremos



- La barra rígida permanece inalterada para valores pequeños de *P*, pero a partir de una fuerza umbral se pandea tal como se ve en la figura, con $y=y(x)$.

- La ecuación que permite modelar este fenómeno viene dada por:

$$EI\frac{d^2y}{dx^2}=-Py$$

siendo *E* el módulo de elasticidad de la barra e *I* el momento de inercia de la sección de la barra.

- Al no poder moverse en los extremos se dan las condiciones de contorno $y(0)=y(l)=0$

Se puede resolver esta ecuación diferencial utilizando una aproximación numérica.

Si se divide la barra en n intervalos de longitud $h=l/n$ se tiene

$$0 = x_0 < x_1 < x_2 < \ldots < x_{n-1} < x_n = l$$

Se puede aproximar la función $y=y(x)$ en cada punto de la barra por

$$y(x+h) = y(x) + hy'(x) + (h^2/2)y''(x) + O(h^3)$$

Si se desprecian los términos de orden superior al 2º y se particulariza la función para los puntos $x_{i+1} = x_i + h$ y $x_{i-1} = x_i - h$ se tiene

$$\begin{cases} y(x_{i+1}) = y(x_i + h) = y(x_i) + hy'(x_i) + (h^2/2)y''(x_i) \\ y(x_{i-1}) = y(x_i - h) = y(x_i) - hy'(x_i) + (h^2/2)y''(x_i) \end{cases}$$

Sumando miembro a miembro $\quad y(x_{i+1}) + y(x_{i-1}) = 2y(x_i) + h^2 y''(x_i)$

y despejando la segunda derivada $\quad y''(x_i) = \dfrac{y(x_{i+1}) + y(x_{i-1}) - 2y(x_i)}{h^2}$, para $i = 1,2,\ldots,n\text{-}1$

Puesto que en los puntos $x = x_i$, con $i = 1,2,\ldots,n\text{-}1$ se verifica la ecuación $EI\dfrac{d^2y}{dx^2} = -Py$ podemos escribir

$$EI\,\frac{y(x_{i+1}) + y(x_{i-1}) - 2y(x_i)}{h^2} = -Py(x_i), \text{ para } i = 1,2,\ldots,n\text{-}1$$

Si denominamos $\lambda = -(Ph^2)/(EI)$ y escribimos la ecuación anterior para cada punto $x = x_i$, con $i = 1,2,\ldots,n\text{-}1$ queda

$$\begin{cases} y(x_0) - 2y(x_1) + y(x_2) = \lambda y(x_1) \\ y(x_1) - 2y(x_2) + y(x_3) = \lambda y(x_2) \\ \ldots\ldots\ldots \quad \ldots\ldots \\ y(x_{n-2}) - 2y(x_{n-1}) + y(x_n) = \lambda y(x_n) \end{cases}$$

Teniendo en cuenta las condiciones de contorno *y(0)=y(l)=0* queda

$$\begin{cases} -2y(x_1) + y(x_2) = \lambda y(x_1) \\ y(x_1) - 2y(x_2) + y(x_3) = \lambda y(x_2) \\ \ldots\ldots\ldots \quad \ldots\ldots \\ y(x_{n-2}) - 2y(x_{n-1}) = \lambda y(x_n) \end{cases}$$

Y expresado en forma matricial

$$\begin{bmatrix} -2 & 1 & & & & & \\ 1 & -2 & 1 & & & & \\ & 1 & -2 & & & & \\ & & & \cdots & & & \\ & & & & -2 & 1 & \\ & & & & 1 & -2 & 1 \\ & & & & & 1 & -2 \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ .. \\ y_{n-2} \\ y_{n-1} \\ y_n \end{bmatrix} = \lambda \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ .. \\ y_{n-2} \\ y_{n-1} \\ y_n \end{bmatrix}$$

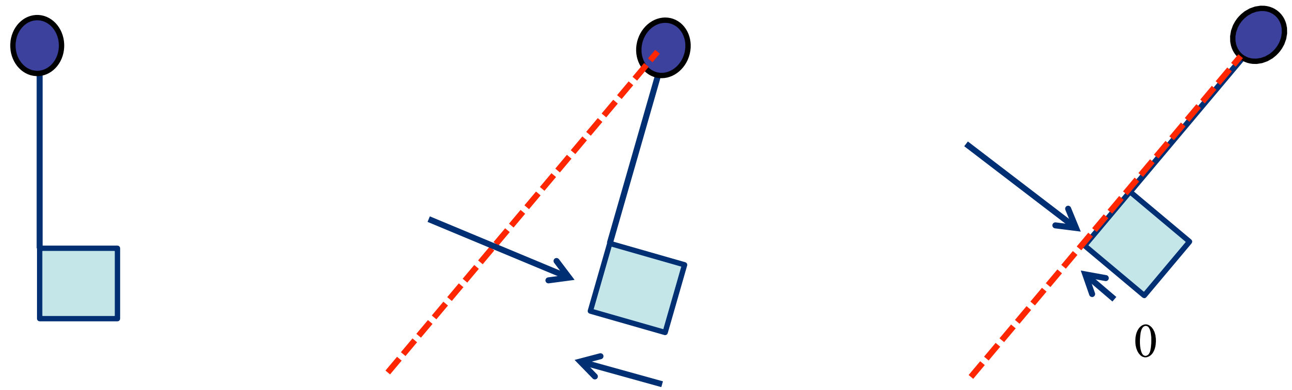
Es decir $\quad Av = \lambda v$

Cada valor propio determina una carga $p = \lambda(EI/h^2)$ denominada carga crítica. Estas cargas críticas son las únicas de interés práctico, ya que son las que determinan los diferentes pandeos de la barra en estado de equilibrio. El valor más pequeño de *P* suele ser de vital importancia.

Otros problemas de ingeniería ligados a los valores propios: resonancia

- **Todas las estructuras vibran. Su frecuencia natural está relacionada con los valores propios que se obtienen al hacer un análisis dinámico de la estructura en equilibrio.**

- **Ejemplo típico de resonancia: el columpio.**
- **Su frecuencia natural (péndulo) se puede valorar de forma intuitiva**



Desviación máxima

Desviación máxima

Otros problemas de ingeniería ligados a los valores propios: resonancia
Los puentes deben analizarse desde el punto de vista dinámico correctamente para evitar catástrofes

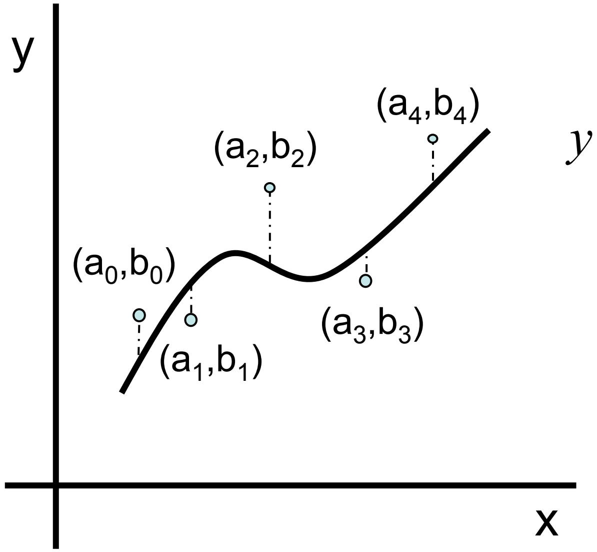
Puente de Tacoma (Seattle. Washington. 1935)
http://www.microsiervos.com/archivo/mundoreal/caida-puente-colgante-tacoma.html

Puente del Milenio (Londres. U.K. 2000)
http://www.microsiervos.com/archivo/ciencia/problema-resonancia-puente-milenio.html

El Puente del Milenio volvió a abrir en el 2002, después de que los ingenieros lo ajustaran con 91 amortiguadores para absorber las oscilaciones laterales y verticales.

Caso de estudio 3
Problemas de Mínimos Cuadrados

Ejemplo: Ajuste de una curva a una nube de puntos



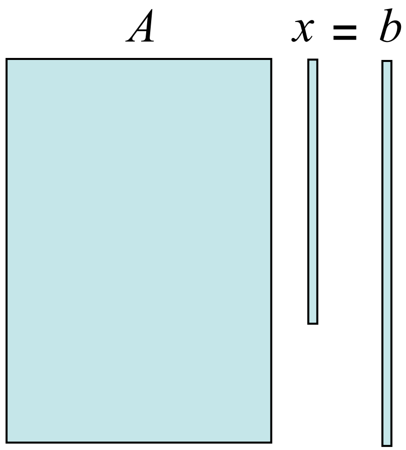
$$y = x_0 + x_1 x + x_2 x^2 + x_3 x^3$$

Imponiendo la condición de que la curva pase por los puntos conocidos se tiene

$$Para\ i = 0,1,\ldots,m-1$$

$$b_i = \sum_{j=0}^{n-1} x_j a_i^{\,j}$$

En forma matricial



con $A \in \Re^{m \times n}, x \in \Re^{n}, b \in \Re^{m}$

$$\text{y} \quad a_{\text{ij}} = a_i^{\,j},\; x = \begin{bmatrix} x_0 & x_1 & \ldots & x_{n-1} \end{bmatrix}^T \in \Re^n, b = \begin{bmatrix} b_0 & b_1 & \ldots & b_{m-1} \end{bmatrix}^T \in \Re^m$$

$$\left\| Ax - b \right\|_2 = \sqrt{(Ax-b)_0^{\;2} + (Ax-b)_1^{\;2} + \ldots + (Ax-b)_{m-1}^{\;\;2}}$$

El ajuste de la curva por mínimos cuadrados es equivalente a la resolución de un sistema de ecuaciones sobredeterminado:

$$\text{Resolver } Ax_{LS} = b \Leftrightarrow \boxed{\text{Encontrar } x_{LS} \in \Re^n \text{ tal que } \left\| Ax_{LS} - b \right\|_2 = \min_{x \in \Re^n} \left\| Ax - b \right\|_2}$$

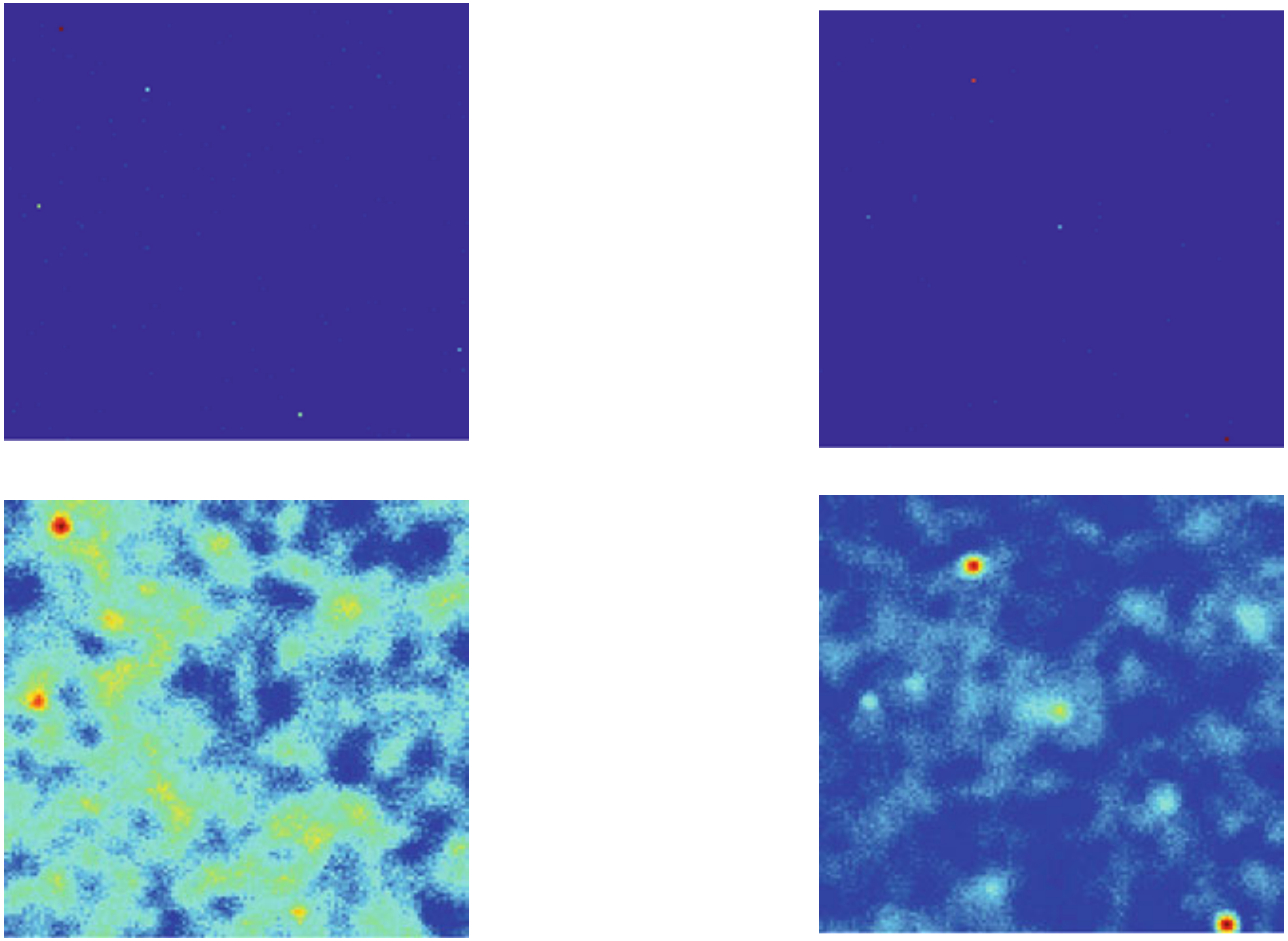
Problema Lineal de Mínimos Cuadrados

Tópicos

- Problema lineal y no lineal de mínimos cuadrados
- Resolución de sistemas de ecuaciones sobredeterminados
- Descomposición QR

Caso de estudio 4:
Ruido debido a la radiación cósmica

- The **Cosmic Microwave Background (CMB)** is a diffuse radiation which is contaminated by the radiation emitted by point sources. The precise knowledge of CMB fluctuations can lead to a better knowledge of the chemistry at the early stages of the Universe. In this work, we present an efficient algorithm, with a high degree of parallelism, which can improve, from the computational point of view, the classical approaches for detecting point sources in Cosmic Microwave Background maps. **High performance computing libraries and parallel computing techniques have allowed to construct a portable, fast and numerically stable algorithm.** To check the performance of the new method, we have carried out several simulations resembling the observational data collected by the Low Frequency Instrument of the Planck satellite. The sources are detected in their real positions.



Modelado del problema

Incógnitas

$$d(x,y) = \sum_{\alpha=1}^{n} a_\alpha b(x - x_\alpha, y - y_\alpha) + (f * b)(x,y) + n(x,y)$$

Radiación observada

Radiación de las fuentes

Efecto de la CMB

Ruido

Tras ciertas reordenaciones

$$d = \phi a + c, \text{ con } \phi \in \Re^{N\times n}, N >> n$$

Información

Incógnitas

Ruido Gaussiano

Imponiendo las condiciones adecuadas para minimizar el ruido y haciendo un filtrado para calcular el valor de *n*, se llega al siguiente

Problema matricial

$$\phi \in N \times n, N >> n \,;\, \xi \in N \times N, \text{Def. Positiva.}$$

$$Ma = e \quad \text{con} \quad M = \phi^T \xi^{-1} \phi \qquad e = \phi^T \xi^{-1} d$$

Solución clásica

1. Calcular *M*
2. Calcular *e*
3. Resolver el sistema

Dificultades y soluciones

- Evitar operaciones numéricamente inestables como el cálculo de la inversa
- Utilizar moderadamente la memoria
- Conseguir un tiempo de ejecución razonable
- Organizar el algoritmo de tal forma que se puedan usar Librerías de Altas Prestaciones

Solución utilizando Librerías de Altas Prestaciones

Sea $\xi = L * L^T$, con L triangular inferior (Descomposición de Cholesky)

$$e = \phi^T \xi^{-1} d = \phi^T (L * L^T)^{-1} d = \phi^T L^{-T} L^{-1} d = \phi^T \overbrace{(L^{-T} \underbrace{(L^{-1} d)}_{c_1})}^{c_2} = \phi^T c_2$$

$$M = \phi^T \xi^{-1} \phi = \phi^T (L * L^T)^{-1} \phi = (L^{-1}\phi)^T (L^{-1}\phi) = Z^T Z = (QR)^T (QR) = R^T (Q^T Q) R = R^T R$$

con $\mathrm{Z} = QR$, (Desc.QR de Z; Q ortogonal: $Q^T Q = I$)

Algoritmo

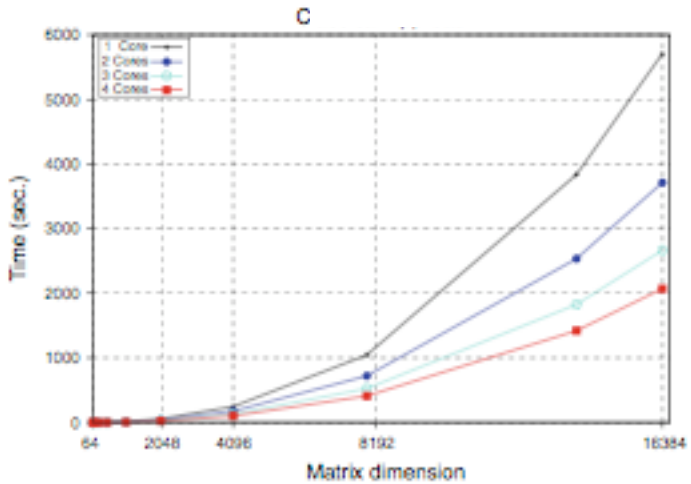
$$\text{Resolver } Ma = e \Leftrightarrow R^T Ra = e \Leftrightarrow R^T y = e \text{ con } Ra = y$$

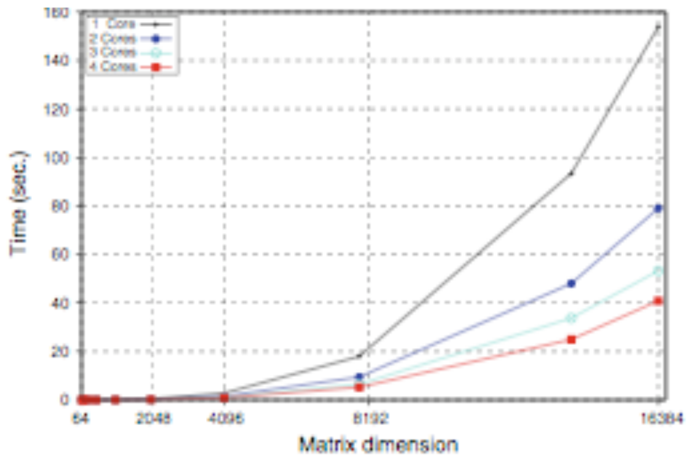
1. Calcular la Descomposición de Cholesky: $\xi = L * L^T$ LAPACK
2. Resolver el sistema de ecuaciones triangular inferior $Lc_1 = \mathrm{d}$ BLAS2
3. Resolver el sistema de ecuaciones triangular superior $L^T c_2 = c_1$ BLAS2
4. Calcular el producto matriz-vector $e = \phi^T c_2$ BLAS2
5. Resolver el sistema de ecuaciones triangular inferior con multiples términos independientes $LZ = \phi$ BLAS3
6. Calcular la *R* de una Descomposición QR: $QR = Z$ LAPACK
7. Resolver el sistema de ecuaciones triangular superior $R^T y = e$ BLAS2
8. Resolver el sistema de ecuaciones triangular inferior $Ra = y$ BLAS2

Prestaciones

- The testbed system used in the experimentation is composed by one Intel Xeon E5530 Quad-Core processors (4 cores) running at 2.40 GHz with Ubuntu Linux distro (10.04.2 LTS) as operating system. The high performance implementation of LAPACK was provided by Intel MKL (*Mathematical Kernel Library, version 10.3). Finally, experiments reported in this section employ ieee 754 double precision arithmetic.*
- OpenMP is used when possible.

Execution time for the classical and CMB algorithms



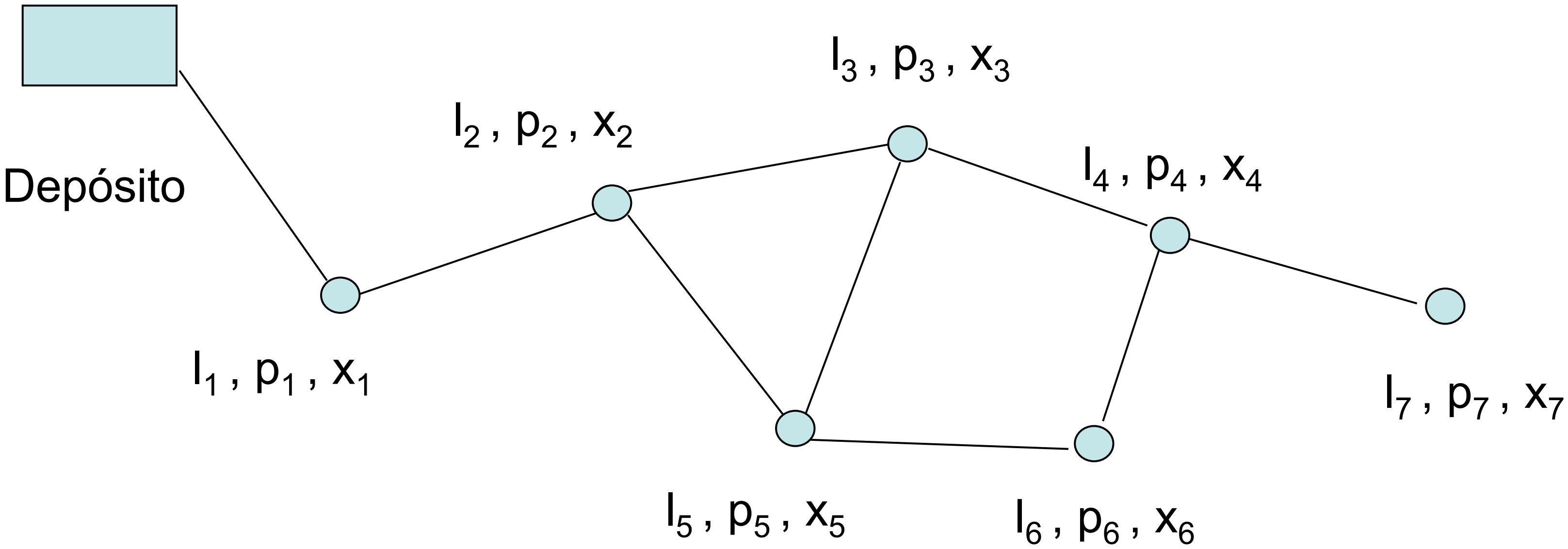


Time (s) for $N = 2^{14}$

Algorithm/cores	1	2	3	4
Classical	$5.71e + 03$	$3.71e + 03$	$2.67e + 03$	$2.07e + 03$
CMB	$1.54e + 02$	$7.90e + 01$	$5.32e + 01$	$4.09e + 01$

Caso de estudio 5

- Minimización de fugas en una red de distribución de agua



l_i: flujo de pérdidas en un nodo
p_i: presión en un nodo
x_i: presión en la válvula reguladora

El problema de la minimización de fugas se puede formular de la siguiente manera:

$$\begin{aligned} \text{minimizar } & f(x) = \sum_{j=1}^{n} l_j \\ \text{sujeto a: } & F(x,u) = 0 \\ & g_j(x) = p_j^{\text{mín}} - p_j \leq 0, \qquad \forall j \in I_c, \end{aligned}$$

donde $x = \left(q^T, l^T, h^T\right)^T$ es el vector de variables hidráulicas del sistema, que incluye el vector de fugas l, y u es el vector de consignas de las válvulas, de n_v componentes. La restricción $F(x,u) = 0$ expresa las ecuaciones de equilibrio del análisis hidráulico estático, dadas por el sistema de ecuaciones (5.6), mientras que la segunda restricción impone presiones mínimas, $p_j^{\text{mín}}$, sobre un determinado conjunto de nodos denotado por I_c.

Podemos entonces definir $\tilde{f}(u) = f(x(u))$, $\tilde{g}_j(u) = g_j(x(u))$, y reformular el problema (5.7) en términos de las variables u

$$\begin{aligned} \text{minimizar } & \tilde{f}(u) = \sum_{j=1}^{nh} l_j \\ \text{sujeto a } & \tilde{g}_j(u) = p_j^{\text{mín}} - p_j \leq 0, \qquad \forall j \in I_c, \end{aligned}$$

•**Problemas de optimización**

$$\min_{x \in \Re^n} \phi(x_1, x_2, \ldots, x_n)$$

•**Problemas de optimización con restricciones**

$$\min_{x \in \Re^n} \varphi(x_1, x_2, \ldots, x_n)$$

$$\text{sujeto a } \; F(x) = 0 \;,\; G(x) < 0 \;,\; \ldots$$

•**Resolución de sistemas de ecuaciones no lineales**

$$\begin{cases} f_1(x_1, x_2, \ldots, x_n) = 0 \\ f_2(x_1, x_2, \ldots, x_n) = 0 \\ \ldots. \\ f_n(x_1, x_2, \ldots, x_n) = 0 \end{cases}$$

Caso de estudio 6

Resolución de Sistemas de Ecuaciones Lineales

Sistemas de ecuaciones lineales con matrices de Hilbert

Matriz de Hilbert de orden 4

$$A = \begin{bmatrix} 1 & 1/2 & 1/3 & 1/4 \\ 1/2 & 1/3 & 1/4 & 1/5 \\ 1/3 & 1/4 & 1/5 & 1/6 \\ 1/4 & 1/5 & 1/6 & 1/7 \end{bmatrix}$$

```
% Programa MATLAB
% Sistemas de ecuaciones con matrices de Hilbert

clear
clc

for n=4:14
        n
        A=hilb(n);
        b=zeros(n,1);
        for i=1:n
            b(i)=sum(A(i,:));
        end
        sol=ones(n,1);
        x=A\b;
        [sol x]
end
```

Resultados

n =4
ans =

1.0000	1.0000
1.0000	1.0000
1.0000	1.0000
1.0000	1.0000

n =8
ans =

1.0000	1.0000
1.0000	1.0000
1.0000	1.0000
1.0000	1.0000
1.0000	1.0000
1.0000	1.0000
1.0000	1.0000
1.0000	1.0000

n =9
ans =

1.0000	1.0000
1.0000	1.0000
1.0000	1.0000
1.0000	1.0000
1.0000	1.0000
1.0000	1.0000
1.0000	1.0000
1.0000	1.0000
1.0000	1.0000

n =10
ans =

1.0000	1.0000
1.0000	1.0000
1.0000	1.0000
1.0000	1.0000
1.0000	0.9999
1.0000	1.0002
1.0000	0.9997
1.0000	1.0003
1.0000	0.9999
1.0000	1.0000

n =11
ans =

1.0000	1.0000
1.0000	1.0000
1.0000	1.0000
1.0000	1.0001
1.0000	0.9997
1.0000	1.0012
1.0000	0.9973
1.0000	1.0036
1.0000	0.9970
1.0000	1.0014
1.0000	0.9997

n =12
ans =

1.0000	1.0000
1.0000	1.0000
1.0000	1.0000
1.0000	1.0004
1.0000	0.9965
1.0000	1.0161
1.0000	0.9532
1.0000	1.0881
1.0000	0.8933
1.0000	1.0804
1.0000	0.9658
1.0000	1.0063

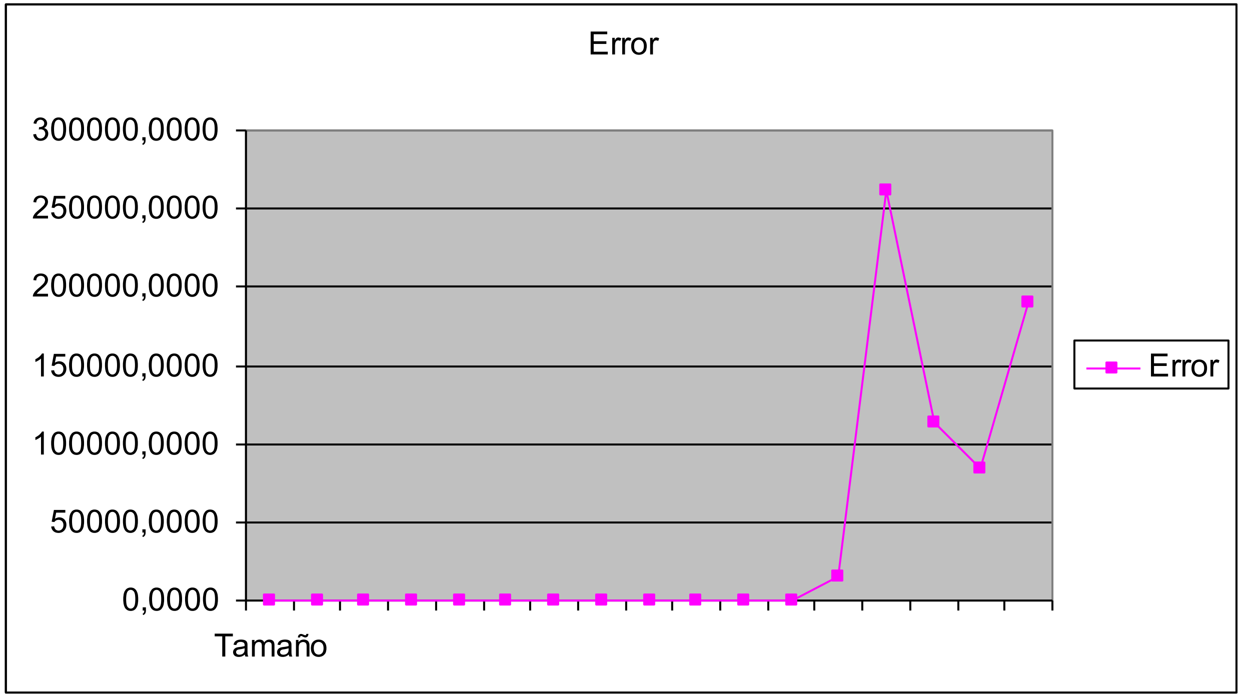
n =13
ans =

1.0000	1.0000
1.0000	1.0000
1.0000	1.0001
1.0000	0.9989
1.0000	1.0109
1.0000	0.9379
1.0000	1.2255
1.0000	0.4608
1.0000	1.8603
1.0000	0.0932
1.0000	1.6062
1.0000	0.7672
1.0000	1.0391

n =14
ans =

1.0000	1.0000
1.0000	1.0000
1.0000	1.0006
1.0000	0.9916
1.0000	1.0519
1.0000	0.8693
1.0000	0.8092
1.0000	3.3755
1.0000	-6.8166
1.0000	15.3612
1.0000	-15.2629
1.0000	12.2947
1.0000	-3.4253
1.0000	1.7506

Error cometido



$$Error = \sum_{i=1}^{n} \left(x_i - sol_i\right)^2$$

Conclusiones

Es necesario:

- Adquirir los conceptos básicos de computación matricial
- Generar algoritmos numéricamente estables para resolver problemas bien condicionados
- Estudiar los algoritmos matriciales básicos:
  - Descomposiciones directas: LU, LDL^T, Cholesky, QR, y sus aplicaciones
  - Descomposiciones iterativas: Cálculo de valores propios y singulares
- Saber utilizar las librerías donde están implementadas las descomposiciones matriciales