UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO

UNIDAD ACADEMICA DE LOS CICLOS PROFESIONAL Y DE POSTGRADO DEL COLEGIO DE CIENCIAS Y HUMANIDADES

ESTIMADORES DEL NUMERO DE CONDICION EN SISTEMAS ALGEBRAICOS LINEALES Y OTROS PROBLEMAS ASOCIADOS

T E S I S

QUE PARA OBTENER EL GRADO DE

MAESTRO EN CIENCIAS DE LA COMPUTACION

P R E S E N T A

JOSE GERMAN GONZALEZ SANTOS

Instituto de Investigaciones en Matemáticas Aplicadas y Sistemas

MEXICO, D. F. JULIO 1990

UNAM – Dirección General de Bibliotecas Tesis Digitales Restricciones de uso

CONTENIDO

INTRODUCCION

Uno de los temas importantes del Análisis Numérico es la identificación y tratamiento de problemas muy sensitivos a pequeñas perturbaciones de sus datos, conocidos en la literatura como **problemas mal condicionados**. Por ejemplo, la inversa de la matriz

$$A = \begin{bmatrix} 1 & 1 \\ 1 & 1+c \end{bmatrix},$$

es muy sensitiva a pequeñas variaciones del parámetro c, sus entradas varían de acuerdo a 1/c.

El punto de partida en el tratamiento de los problemas mal condicionados, es asignar a cada problema una medida de su condicionamiento, de manera que cuando esta cantidad sea grande indique que el problema es mal condicionado.

La teoría de la condición empleada en este trabajo y en los especializados del tema, fué propuesta originalmente por Rice [Ri1] en 1961 y consiste en asignar a cada problema cierto número positivo, conocido como el **número de condición**. La magnitud de este número está relacionada directamente con el mal condicionamiento del problema; cuando es grande indica que el problema es mal condicionado.

Desde el punto de vista práctico, que un problema sea mal condicionado, quiere decir que es difícil de resolver numéricamente, es decir, el resultado proporcionado por el método empleado puede diferir considerablemente de la solución exacta del problema original. Por supuesto que el resultado no depende solamente del mal condicionamiento del problema sino también de la inestabilidad del método. Un problema puede parecer mal condicionado si es resuelto por medio de un método inestable.

El número de condición da solamente una cota superior sobre los efectos de las perturbaciones, y puede suceder que para ciertos problemas la cota sea muy cruda e inclusive sin utilidad práctica, pues no es posible que todas las propiedades de un problema queden representadas por un número, sin embargo en muchos casos el número de condición da una buena idea del comportamiento numérico del problema.

El presente tabajo se ocupa de problemas clásicos mal condicionados acerca de polinomios y matrices. En el primer capítulo se presentan algunos ejemplos de problemas mal condicionados y resultados de la teoría de la

condición necesarios en los capítulos posteriores.

A pesar de que los polinomios son las funciones más conocidas y empleadas en casi todas las áreas del Análisis Numérico, sus propiedades numéricas son poco conocidas. En el capítulo 3 se abordan tres problemas básicos relacionados con polinomios: la evaluación de polinomios, representación de polinomios y generación de polinomios ortogonales. Sin pretender hacer un tratamiento exhaustivo, se obtiene el número de condición de cada uno de estos problemas y se exponen algunos algoritmos adecuados para resolverlos.

La segunda parte del trabajo está relacionada con el problema $Ax = b$. En el capítulo 2 se presentan algunos resultados de la teoría de normas y del análisis de sensibilidad, quedando establecidas las cotas de error para el problema $Ax = b$ en términos de las perturbaciones de los datos del problema. En estas relaciones queda clara la importancia que juega el número de condición del problema de inversión, $K(A) = \|A\|\|A^{-1}\|$. El lector familiarizado con estos resultados puede omitir el capítulo.

Las matrices mal condicionadas pueden aparecer en diversas aplicaciones, por ejemplo, en Reología, determinar la rapidez de corte del esfuerzo cortante $f(\tau) = \dot{\gamma}(\tau)$ de un fluido viscoelástico, en un viscosímetro de cilindros concéntricos, requiere de la solución de una ecuación integral

$$\int_{k^2\tau_b}^{\tau_b} h(\tau)f(\tau)d\tau = \Omega(\tau_b). \tag{1}$$

Ninguno de los métodos existentes para resolver (1), es adecuado para núcleos arbitrarios y cuando la función Ω es solamente conocida modestamente, como sucede en una situación experimental. El éxito en la solución de la ecuación, depende de la precisión de Ω y de la forma de la función de peso $h(\tau)$. Algunos de los métodos se reducen a resolver un sistema algebraico lineal, donde la matriz resulta generalmente mal condicionada.

En la solución numérica de ecuaciones diferenciales parciales por medio de diferencias finitas, también aparecen matrices mal condicionadas, que además son poco densas, definidas positivas y con estructura.

Los métodos de estimación del número de condición son tratados en el capítulo 4, y en el 5 se comparan extensamente, utilizando diferentes conjuntos de matrices de prueba.

Calcular el número de condición de una matriz es una operación costosa, debido a que es necesario contar con la inversa de la matriz, y en algunos

casos esta operación es más cara que resolver el problema original. Esto hace interesante e importante la búsqueda de métodos de estimación, donde el estimador aproxime adecuadamente al valor real, con el mínimo número de operaciones.

Los métodos de estimación, para matrices generales parten del hecho de que la matriz ya se encuentra en forma triangular, o bien ha sido factorizada de acuerdo a alguna descomposición, la cual tiene factores triangulares. La relación entre el número de condición de la matriz original y el número de condición de sus factores, depende de la matriz y del método de factorización empleado. Los primeros métodos tratados son para matrices triangulares, y algunos de ellos pueden emplearse para estimar K(A) cuando A es una matriz de orden nxn y no singular. Existen métodos más adecuados para cierto tipo de matrices, por ejemplo, para matrices tridiagonales, poco densas sin estructura, positivas, etc.

Los métodos de estimación del número de condición los podemos clasificar en dos grupos: El primero comprende los métodos que dependen solamente de la magnitud de los elementos de la matriz y son obtenidos a partir de desigualdades de matrices. Y en el segundo grupo tenemos los que provienen de algoritmos heurísticos o probabilísticos motivados por la definición de la norma subordinada.

Por facilidad de cálculo, la mayoría de los códigos existentes estiman el número de condición para la norma $\| \|_\infty$ o $\| \|_1$, y solamente para cierto tipo de matrices se tienen estimadores en otras normas. Hasta el momento, no existe el mejor método de estimación del número de condición, pues siempre es posible encontrar ejemplos donde uno o varios de ellos fallen.

En el capítulo 4 presentamos también un nuevo método para la estimación del número de condición de matrices triangulares, el método DIV-MOD es una variante del método divide y venceras y proporciona sistemáticamente mejores resultados que los obtenidos por otros métodos y sólo comparables con los obtenidos por el método vía optimización convexa.

La mayoría de los métodos incluidos en el trabajo, fueron programados en lenguaje Fortran y otros en C; usando también subprogramas de los paquetes matemáticos IMSL, LINPACK, FORSYTHE y MATLAB. Además, el análisis de ciertos problemas se realizó con la ayuda de los paquetes matemáticos REDUCE y DERIVE.

CAPITULO 1

PROBLEMAS MAL CONDICIONADOS Y NUMERO DE CONDICION.

1.1 Problemas mal condicionados
1.2 Número de condición
1.3 Algoritmos estables

Al resolver un problema por medio de una computadora digital, es necesario representar los datos del problema de alguna manera. En general, podemos decir que la representación es aproximada; de modo que el problema a resolver ya no es el problema original, sino uno ligeramente diferente. La pregunta que surge de inmediato es ¿ qué tanto difiere el resultado del nuevo problema con respecto al original ?. Esto se conoce como la **sensitividad del problema.**

En la literatura se reportan problemas muy sensitivos a variaciones en sus datos, éstos son conocidos como **problemas mal condicionados.** A continuación se dan tres ejemplos de este tipo de problemas.

1.1 Problemas mal condicionados.

Primer ejemplo: Determinar la intersección de dos rectas casi paralelas. El modelo matemático para este problema se obtiene al introducir un sistema de coordenadas y escribir la ecuación para cada una de las rectas:

$$\begin{aligned} y &= a_1 x + b_1 \\ y &= a_2 x + b_2, \end{aligned} \tag{1.1}$$

donde $a_2 = a_1 + \epsilon$, con ϵ pequeño. El punto de intersección P de las dos rectas está dado por:

$$P = 1/\epsilon\,(b_1 - b_2,\ a_2 b_1 - a_1 b_2).$$

Si el coeficiente a_1 es alterado, en una cantidad δ, el punto de intersección del sistema perturbado es

$$P(\delta) = \frac{1}{(a_1 - \delta)} (b_1 - b_2, a_2 b_1 - (a_1 + \delta) b_2).$$

Cuando la perturbación δ es del orden de ϵ, la distancia entre P y P(δ),

$$\|P - P(\delta)\| = O\left(\frac{\delta}{\epsilon\,(\epsilon - \delta)}\right),$$

es muy grande.

Segundo ejemplo, [He2]: Determinar las raíces de un polinomio. Los datos son los coeficientes del polinomio o algunas otras cantidades que definan al polinomio univocamente. La salida es un conjunto de números complejos, aceptados como los ceros del polinomio. Por ejemplo, el polinomio

$$P(z) = z^{10} - 10z^9 + 45z^8 - 120z^7 + 210z^6 - 252z^5 + 210z^4 - 120z^3 + 45z^2 - 10z + 1,$$

es igual a $P(z) = (z - 1)^{10}$ y tiene a 1 como raíz con multiplicidad 10. Si el coeficiente del término constante cambia de 1 a $1 - 10^{-10}$, las raíces del nuevo polinomio son:

$$1 + 10^{-1} e^{2\pi k i/10}, \quad k = 0, 1, 2, \ldots 9.$$

Una perturbación de 10^{-10} en uno de los coeficientes causa que los ceros del polinomio cambien en una cantidad que es 10^9 veces el cambio del coeficiente constante.

Tercer ejemplo: Evaluación de una función. La importancia de este problema reside en que un amplio número de problemas pueden verse como un mapeo entre dos espacios. Como un caso particular puede considerarse la función entre dos espacios de dimensión finita $f : S \subseteq \mathbb{R}^m \longrightarrow \mathbb{R}^n$. Las m componentes del argumento x de f son los datos que determinan el problema y las n componentes de f(x) son la respuesta.

El problema numérico es calcular una aproximación a f(x) dado x. La naturaleza de la función puede limitar la precisión obtenida en su evaluación. Por ejemplo, la función

$$f(x) = \frac{1}{\sqrt{1 - x^2}} \qquad \text{para } x \in (-1,1),$$

es continua en todo el dominio y como es diferenciable, su sensibilidad puede estimarse por medio de su derivada:

$$f(x + \delta x) - f(x) \approx f'(x)\,\delta x$$

$$\approx \frac{x}{(1 - x^2)^{3/2}}\delta x.$$

La sensitividad de la función, como lo establece la relación anterior, es mayor en los extremos del intervalo. Por ejemplo, en la tabla (1.2) se presentan las variaciones de la función cerca del punto $x = 1 - 2^{-20}$.

$x = 1-2^{-20}$	$f(x\pm\delta x)$	$f(x\pm\delta x)-f(x)$
x - 5ε	568.013	-156.064
x - 4ε	591.206	-132.870
x - 3ε	617.495	-106.582
x - 2ε	647.634	-76.442
x - ε	682.666	-41.410
x	724.077	0.0
x + ε	774.071	49.993
1 - 2ε	836.092	112.015
1 - 3ε	915.893	191.816
1 - 4ε	1024.000	299.922
1 - 5ε	1182.413	458.335

Tabla (1.1) Variación de la función $f = 1/(1-x^2)^{1/2}$ cerca de $x = 1 - 2^{-20}$. ($\varepsilon = 1.92093E-7$)

De acuerdo con la tabla anterior, una ligera variación del punto de evaluación, $x = 1 - 2^{-20}$, produce cambios drásticos en el valor de la función; la perturbación de la función es 10^8 veces mayor a la ocurrida en el punto de evaluación.

1.2 Número de condición.

Una manera de cuantificar la sensitividad de un problema es asignarle un número, el número de condición, de manera que cuando este número sea relativamente grande advierta que el problema es mal condicionado. La definición de número de condición, adoptada en este trabajo y en la mayoría de la literatura, se debe a Rice [Ri1].

Definición.1.1. Sean X y Y dos espacios lineales normados y f una función f: D ⊆ X → Y, con D un dominio abierto. Sea x ∈ D fijo e y = f(x), y se supone que ninguno de los dos son ceros en sus repectivos espacios. La **sensitividad de la función f en x**, respecto a cambios relativos en x, será medida por el **número de condición relativo**

$$K_r(f, x) = \lim_{\delta \to 0} \sup_{\|h\| = \delta} \left\{ \frac{\|f(x+h) - f(x)\|}{\|f(x)\|} \Big/ \frac{\|h\|}{\|x\|} \right\}, \tag{1.2}$$

suponiendo que el límite existe.

Otra definición, empleada con menos frecuencia, es la la de **número de condición absoluto**

$$K_a(f, x) = \lim_{\delta \to 0} \sup_{\|h\| = \delta} \left\{ \frac{\|f(x+h) - f(x)\|}{\|f(x)\|} \right\}, \tag{1.3}$$

suponiendo que el límite existe.

El número de condición da solamente una cota superior sobre los efectos de las perturbaciones, y puede suceder que para ciertos problemas la cota sea muy cruda e inclusive sin utilidad práctica, pues no es posible que todas las propiedades de un problema queden representadas por un solo número. Sin embargo en muchos casos el número de condición da una buena idea del comportamiento numérico del problema.

Si un algoritmo estable es empleado para resolver un problema bien condicionado P(d), entonces la solución exacta del problema perturbado $P(d^*)$, y la solución numérica del problema exacto $P^*(d)$, están cercanas, además, como el problema es bien condicionado y d y d^* están cercanas, entonces las respuestas de P(d) y $P(d^*)$ también lo están. En otras palabras, si un algoritmo estable es empleado para resolver el problema P(d), entonces las soluciones exacta y númerica están cercanas.

Cuando un algoritmo estable se emplea para resolver un problema mal condicionado, no existe garantía de que la solución numérica y la exacta estén cercanas; sin embargo, como el algoritmo es estable, la solución de $P(d^*)$ y $P^*(d)$ están cercanas y por lo tanto, la diferencia entre las respuestas de P(d) y $P^*(d)$ debe ser aproximadamente igual a la diferencia entre las

respuestas P(d) y P(d^*).

El número de condición empleado en este trabajo es el número de condición relativo y será denotado por la letra K omitiendo el subíndice que indica relativo.

Si la función f tiene derivada de Fréchet $\partial f/\partial x$ en x^*, entonces (1.2) se transforma en

$$K(f, x^*) = \frac{\|x^*\|}{\|y^*\|} \| [\frac{\partial f}{\partial x}]_{x^*} \| \qquad (y^* = f(x^*)). \tag{1.4}$$

Un ejemplo particular de la relación anterior es cuando $X = \mathbb{R}^n$ y $Y = \mathbb{R}^m$. La derivada de Fréchet, como es bien conocido, es un mapeo lineal definido por el Jacobiano de f. La norma de vectores es cualquier norma y la de matrices es la norma inducida.

Con el empleo de la regla de la cadena para la derivada de Fréchet, es posible demostrar que el número de condición de la composición de funciones, gof, cumple la relación siguiente:

$$K(gof, x) \leq K(g, y)\, K(f, x). \tag{1.5}$$

Esta relación establece que, si la composición de funciones es mal condicionada, entonces al menos una de las funciones es mal condicionada. Esto presenta alguna utilidad práctica, ya que en algunas situaciones, es más fácil tratar con un problema alternativo que sea la composición del problema original y de algún otro. Para determinar que el problema original es mal condicionado basta demostrar que la composición es mal condicionada y que uno de los problemas que forman la composición, es bien condicionado. Una aplicación de este resultado se encuentra en el capítulo 3.

Si la función f de la definición (1.1) es lineal, entonces

$$\sup_{\|h\|=\delta} \frac{\|f(x + h) - f(x)\|}{\|h\|} = \sup_{\|h\|=\delta} \frac{\|f(h)\|}{\|h\|},$$

es independiente de x y δ, e igual a la norma de f. Por lo tanto la relación (1.2) se reduce a

$$K(f, x) = \frac{\|x\|}{\|y\|} \|f\|. \tag{1.6}$$

Si f es invertible, y se obtiene el supremo sobre f(X), de la relación anterior, resulta

(1.7) $$K(f) = \sup_y K(f,x) = \sup_y \frac{\|f^{-1}(y)\|}{\|y\|}\|f\| = \|f^{-1}\|\,\|f\|$$

El número de la derecha, se conoce usualmente como el número de condición del mapeo lineal f.

A continuación, se aplica la definición del número de condición a tres problemas, que aparecen frecuentemente en análisis numérico.

Problema 1: Evaluación de una función. Si $f:\mathbb{R} \longrightarrow \mathbb{R}$ es una función derivable, entonces de acuerdo a (1.4) el número de condición relativo está dado por

$$K(f,x) = \frac{|f'(x)|}{|f(x)|}|x|.$$

Por ejemplo, para la función de la tabla (1.1), el número de condición relativo

$$K(f,x) = \frac{x^2}{(1 - x^2)},$$

es grande para valores cercanos a ±1; en particular, si $x = 1-2^{20}$, el número de condición es igual a $5.24\text{x}10^6$. En otras palabras, la precisión esperada en la evaluación de la función está acotada por $5.24\text{x}10^6$ veces la precisión del punto de evaluación, es decir,

$$\frac{\|f(x+h)-f(x)\|}{\|f(x)\|} \leq 5.24\text{x}10^6\,\frac{\|h\|}{\|x\|}.$$

Problema 2: Cálculo del producto interior de dos vectores; $f(x,y) = \langle x,y\rangle$, con x, y $\in \mathbb{R}^n$. El número de condición de esta función puede obtenerse directamente de la definición (1.2).

$$K(f,(x,y)) = \lim_{\delta\to 0}\ \sup_{\|h\|=\delta} \left\{ \frac{\|f((x,y)+(h_x,h_y)) - f(x,y)\|}{\|f(x,y)\|} \Big/ \frac{\|h_x,h_y\|}{\|(x,y)\|} \right\},$$

donde $h = (h_x, h_y)$. Una manera natural de definir la norma de un elemento de $\mathbb{R}^n \times \mathbb{R}^n$ es

$$\|(x,y)\|_p = (\|x\|_p^p + \|y\|_p^p)^{1/p}, \text{ con } p \geq 1.$$

Si (p=2) entonces

$$(1.8)\qquad K(f,\ (x,y)) = \lim_{\delta\to 0} \frac{(\|x\|^2+\|y\|^2)^{1/2}}{|\langle x,y\rangle|} \sup_{\|h\|=\delta} \frac{|\langle x,h_y\rangle+\langle y,h_x\rangle+\langle h_x,h_y\rangle|}{\delta}.$$

Como

$$|\langle x,h_y\rangle + \langle y,h_x\rangle + \langle h_x,h_y\rangle| \le |\langle x,h_y\rangle| + |\langle y,h_x\rangle| + |\langle h_x,h_y\rangle|$$

$$\le \|x\|\|h_y\| + \|y\|\|h_x\| + \|h_x\|\|h_y\|,$$

y eligiendo a $h_x = \dfrac{x}{(\|x\|^2+\|y\|^2)^{1/2}}\delta$, y a $h_y = \dfrac{y}{(\|x\|^2+\|y\|^2)^{1/2}}\delta$,

entonces el miembro derecho de la relación (1.8) se transforma en

$$\lim_{\delta\to 0} \frac{(\|x\|^2+\|y\|^2)^{1/2}}{|\langle x,y\rangle|} \left(\frac{2\ \|x\|\ \|y\|}{(\|x\|^2+\|y\|^2)^{1/2}} + \frac{\|x\|\|y\|}{(\|x\|^2+\|y\|^2)}\,\delta\right).$$

Por lo tanto

$$K(f,\ (x,y)) = \frac{2}{|\cos(\phi)|}, \text{ donde } \phi = \cos^{-1}(\langle x,y\rangle/\|x\|\|y\|).$$

Por lo tanto, el número de condición, del problema de calcular el producto interno, depende sólo del ángulo que forman los vectores y no de la norma de x o y; para vectores casi perpendiculares, K(f, (x,y)) puede ser grande.

Problema 3. La variación de una raiz simple de un polinomio. Dado el polinomio

$$p(x) = a_0 + a_1 x + a_2 x^2 + \dots + a_n x^n$$

de grado n, interesa conocer como varía la raiz simple z_k, al perturbar ligeramente los coeficientes del polinomio p(x), es decir, que tanto difiere z_k de la raiz k-ésima del polinomio perturbado

$$p_\varepsilon(x) = p(x) + \varepsilon q(x),$$

donde $q(x) \not\equiv 0$ es un polinomio arbitrario y ε es pequeño. Los ceros de $p_\varepsilon(x)$ se denotaran por $z_j(\varepsilon)$, j = 1, 2, ..., n, repetidos de acuerdo a su multiplicidad; $z_j(0)$ denota los n ceros de p(x).

Es bien sabido que las raices de un polinomio son funciones continuas de sus coeficientes, Henrici en [Hei, Vol.1, pag. 281]; por lo tanto $z_j(\varepsilon)$ es una función continua de ε.

Para obtener el número de condición del problema es necesario tener una idea de la variación de $z_j(\varepsilon)$ cerca de cero. De la teoría de funciones de variable compleja, se sabe que $z_j(\varepsilon)$ puede expresarse como una serie de potencias,

$$z_j(\varepsilon) = z_j + \sum_{k=1}^{\infty} b_k \varepsilon^k.$$

Tomando el primer término de la serie anterior, se obtiene la aproximación de primer orden

(1.9) $$z_j(\varepsilon) \approx z_j + b_1 \varepsilon,$$

donde $b_1 = z_j'(0)$. Para obtener una expresión para b_1, se parte de la relación que debe cumplir la raíz $z_j(\varepsilon)$, del polinomio perturbado

$$p(z_j(\varepsilon)) + \varepsilon q(z_j(\varepsilon)) = 0, \quad \text{para todo } \varepsilon.$$

Diferenciando la relación anterior, se obtiene una expresión para $z_j'(\varepsilon)$

$$p'(z_j(\varepsilon))z_j'(\varepsilon) + q'(z_j(\varepsilon))z_j'(\varepsilon) + q(z_j(\varepsilon)) = 0,$$

(1.10) $$z_j'(\varepsilon) = \frac{-q(z_j(\varepsilon))}{p'(z_j(\varepsilon)) + \varepsilon q'(z_j(\varepsilon))}.$$

Substituyendo $b_1 = z_j'(0)$ en (1.9), se obtiene una expresión explícita para la aproximación a $z_j(\varepsilon)$,

(1.11) $$z_j(\varepsilon) \approx z_j - \frac{q(z_j)}{p'(z_j)} \varepsilon$$

El número de condición para el problema de determinar una raíz simple, z_j, de $p(x)$ es igual a

$$K(z_j, \varepsilon) = \frac{|q(z_j)|}{|z_j p'(z_j) - q(z_j)\varepsilon|} |\varepsilon|.$$

Por ejemplo, si $p(x) = \prod_{i=1}^{10}(x - i)$ y $q(x) = x^9$, entonces el cambio más drástico se registra en la raíz $z_8 = 8$, donde

(1.12) $$z_8(\varepsilon) = 8 - 13315.25\varepsilon.$$

Esto significa que la raíz $z_8 = 8$, es mal condicionada a pequeñas perturbaciones del coeficiente a_9. En la tabla (1.2) se presentan las raíces del polinomio $p(x) + \varepsilon q(x)$, obtenidas con una aritmética de 20 dígitos, y la

aproximación (1.12)

ε	Raiz de p(x)+εq(x)	Estimación (1.12)
-32τ	8.02543	8.02539
-16τ	8.0127	8.01269
-8τ	8.00635	8.00634
-4τ	8.00317	8.00317
-2τ	8.00158	8.00158
0	8.0	8.0
2τ	7.99841	7.99841
4τ	7.99682	7.99682
8τ	7.99365	7.99365
16τ	7.9873	7.9873
32τ	7.9746	7.9746

Tabla (1.2). Comparación entre la relación (1.12) y las raices de p(x)+εq(x), obtenidas con una aritmética de 20 digitos de precisión. τ = 2^(-23).

De acuerdo con la tabla anterior, la relación (1.12) es una buena aproximación local de la función $z_j(\varepsilon)$, para valores pequeños de ε.

1.3 Algoritmos estables

Desde el punto de vista práctico, que un problema sea mal condicionado quiere decir que es difícil de resolver numéricamente; es decir, el resultado obtenido por el algoritmo puede diferir considerablemente del valor exacto. Esta diferencia puede no deberse exclusivamente a la naturaleza del problema, sino también al algoritmo. Si el algoritmo es estable, entonces el resultado obtenido no es afectado en forma significativa por los errores de redondeo. Cuando el algoritmo es inestable la acumulación de los errores de redondeo puede crecer rápidamente y obtenerse un valor que difiera considerablemente del valor exacto. La técnica general para determinar la estabilidad numérica de un algoritmo es conocida como análisis retrospectivo, introducida por Wilkinson en [Wi4] con el propósito de analizar algunos algoritmos en algebra lineal. En el análisis retrospectivo se trata de demostrar que el resultado numérico, $y' = y + \Delta y$, de un algoritmo para calcular $y = \phi(x)$, puede ser escrito de la forma $y' = \phi(x + \Delta x)$, es decir, que el resultado es la solución exacta de un problema con los datos perturbados. Si $\|\Delta x\|$ es pequeño se dice

que el algoritmo es estable.

Por ejemplo, el algoritmo de substitución hacia atrás (o hacia adelante) para resolver un sistema algebraico lineal $Tx = b$, donde $T \in \mathbb{R}^{n\times n}$ es una matriz triangular no singular, es estable. Wilkinson en [Wi3, pag.100] demuestra que si u es la unidad de redondeo de la computadora, n el orden de la matriz y $nu < 0.1$ entonces la solución numérica, y, del sistema $Tx = b$ satisface

$$(T + E)\, y = b,$$

donde

$$|e_{ij}| \le (|i - j| + 2)cu|t_{ij}|, \quad 1 \le i,j \le n,$$

y c es una constante del orden de la unidad.

Un ejemplo de algoritmo inestable es la evaluación de la función Bessel

$$J_m(x) = (\tfrac{1}{2}x)^m \sum_{k=0}^{\infty} \frac{(-\frac{1}{4}x^2)^k}{k!(m+k)!} \qquad m \ge 0,$$

por medio de la relación de recurrencia

$$J_{m+1}(x) = \frac{2m}{x} J_m(x) - J_{m-1}(x) \qquad m \ge 1, \tag{1.13}$$

con $J_0(x)$ y $J_1(x)$ conocidos.

En la tabla (1.3) se presentan los valores calculados de $J_m(1)$, cuando $J_0(1)$ y $J_1(1)$ son empleados con una precisión de 6 dígitos y los valores subsecuentes de $J_m(1)$ son obtenidos por medio de la relación de recurrencia anterior.

m	$J_m(1)$ Calculado	$J_m(1)$ Exacto
0	7.6519E-1	7.6519E-1
1	4.4005E-1	4.4005E-1
2	1.1490E-1	1.1490E-1
3	1.9566E-2	1.9563E-2
4	2.4432E-3	2.4766E-3
5	3.7941E-4	2.4975E-4
6	1.3009E-3	2.0938E-5
7	1.5231E-2	1.5023E-6
8	2.1194E-1	9.4223E-8
9	3.3758	5.2492E-9
10	6.0552E+1	2.6306E-10

Tabla (1.3). Evaluación de la función de Bessel por medio de un algoritmo inestable

La serie que define la función de Bessel converge rápidamente a cero y además el número de condición de la función

$$K(J_m, x) = |m - |x| \frac{|J_{m+1}(x)|}{|J_m(x)|}|,$$

no es grande para valores de x cercanos a 1. Por lo tanto el algoritmo (1.13) es numéricamente *inestable* para evaluar $J_m(x)$, aún para valores moderados de m.

La evaluación de la función de Bessel puede realizarse, con mejor éxito, si se emplean fracciones continuas para aproximarla, o bien, evaluar la relación de recurrencia con aritmética de multiple precisión o racional.

CAPITULO 2

CONDICION NUMERICA DEL PROBLEMA AX = B

2.1 Análisis de sensibilidad sin el concepto de norma
2.2 Norma de matrices y vectores
2.3 Análisis de sensibilidad del problema Ax = b
2.4 Problemas efectivamente bién condicionados

Determinar como se modifica la solución del sistema algebraico

(2.1) $$A x = b,$$

al perturbar los datos del problema ligeramente, es uno de los problemas más importantes del algebra lineal numérica. Los datos del problema son: la matriz no singular A y el vector b.

Un estudio sistemático de este problema aparece cuando se introduce el concepto de norma, pues permite establecer relaciones simples del error de la solución del sistema algebraico (2.1).

2.1. Análisis de sensibilidad sin el concepto de norma.

Ben Noble, en [Noll], hace un tratamiento del análisis de error sin el concepto de norma. Para ésto propone cuantificar el cambio que sufre la solución de (2.1) con respecto a cada dato del problema.

En el caso general, es decir, cuando la matriz A y el vector b son perturbados, se obtiene el sistema perturbado

(2.2) $$(A + \delta A)\, y = (b + \delta b),$$

donde $\delta A = (\delta_{ij})$, $\delta b = (\delta b_i)$ e y es la solución de nuevo sistema. De (2.1) y (2.2) se obtiene

(2.3) $$y - x = A^{-1}(\delta b - \delta A y).$$

Si el único dato que sufre cambio es la entrada b_k, entonces la relación (2.3) se transforma en:

$$y - x = A^{-1}\,\delta b,$$

o de manera equivalente

$$\delta x_j = y_j - x_j = \alpha_{jk}\delta b_k, \qquad j = 1, 2, \ldots, n$$

donde α_{jk} es el elemento (j,k) de la inversa de A y n es el orden de la matriz. De acuerdo con la relación anterior, el efecto de la perturbación de b_k en la solución del sistema, depende de la magnitud de la k-ésima columna de A^{-1}; en particular, para la entrada x_j el factor de magnificación es α_{jk}.

El error relativo de x_j respecto a pequeñas variaciones relativas de la entrada b_k, satisface la relación

$$\frac{|y_j - x_j|}{|x_j|} = K_{jk}\,\frac{|\delta b_k|}{|b_k|},$$

donde

$$K_{jk} = \frac{|\alpha_{jk} b_k|}{|x_j|} = \frac{|A_{kj} b_k|}{|x_j \Delta|}. \tag{2.4}$$

$\Delta = \det A$ y A_{kj} es el cofactor de a_{jk}. Los factores K_{jk} son conocidos como los **números de condición para cambios relativos en x_j causados por cambios relativos en b_k.**

La relación (2.4) es valida sólo cuando x_j y b_k son diferentes de cero. Si el número de condición, K_{jk}, es grande, quiere decir que la entrada x_j de la solución es muy sensible a perturbaciones de b_k.

Por ejemplo, la matriz del sistema algebraico

$$\begin{bmatrix} 1 & 1/2 & 1/3 \\ 1/2 & 1/3 & 1/4 \\ 1/3 & 1/4 & 1/5 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 2 \end{bmatrix},$$

tiene como inversa la matriz

$$\begin{bmatrix} 9 & -36 & 30 \\ -36 & 192 & -180 \\ 30 & -180 & 180 \end{bmatrix}.$$

La solución del sistema es:

$$x_1 = 69, \qquad x_2 = -396 \quad y \quad x_3 = 390\,.$$

La solución exacta para el sistema perturbado, $A\,y = b + \delta b$, es

$$y_1 = 98b_1 - 368b_2 + 308b_3 + 69,$$
$$y_2 = -368b_1 + 1928b_2 - 1808b_3 - 396,$$
$$y_3 = 308b_1 - 1808b_2 + 1808b_3 + 390.$$

Los números de condición K_{jk} para $j = 1,2$ y 3, $k = 1,2$ y 3, son:

$$K_{11} = 3/23,\ K_{12} = 0,\ K_{13} = 20/23,$$

$$K_{21} = 1/11,\ K_{22} = 0,\ K_{23} = 10/11,$$

$$K_{31} = 1/13,\ K_{32} = 0,\ K_{33} = 12/13.$$

De acuerdo a este tipo de análisis, la entrada x_3 es la más afectada por perturbaciones en b_3.

Por otra parte, si el único dato del problema que cambia es la entrada (p,q) de la matriz A (cambia de a_{pq} en $a_{pq} + \delta_{pq}$), la expresión (2.3) se transforma en

$$y - x = A^{-1}\delta A y$$

es decir

$$\delta x_j = \alpha_{jp}\delta a_{pq} y_j = \alpha_{jp}\delta a_{pq}(x_j + \delta x_j),$$
$$\text{para } j = 1, 2, \ldots n.$$

Si la perturbación δa_{pq} es pequeña, el témino $\delta a_{pq}\delta x_j$ puede omitirse de la relación anterior, para obtener

$$\delta x_j = y_j - x_j = \alpha_{jp}\delta a_{pq} x_j.$$

El error relativo de x_j, producido por cambios en la entrada a_{pq} de la matriz A, está dado por

$$\frac{|y_j - x_j|}{|x_j|} = K_{j,pq}\frac{|\delta a_{pq}|}{|a_{pq}|},$$

donde

$$K_{j,pq} = \frac{|\alpha_{jp} a_{pq} x_q|}{|x_j|} = \frac{|A_{pj} a_{pj}|}{|\Delta|}\frac{|a_{pq} x_q|}{|a_{pj} x_j|}.$$

Los números $K_{j,pq}$, son los números de condición relativos a cambios de x_j respecto a pequeños cambios relativos de a_{pq}.

Estos números, al igual que los anteriores, deben de interpretarse como factores de amplificación de las perturbaciones relativas, ocurridas a cada dato de problema y tratadas en forma independiente.

Por ejemplo, si la matriz del sistema algebraico del ejemplo anterior, es perturbada en la entradas que a continuación se indican

$$\begin{bmatrix} 1 & 1/2 & 1/3+\varepsilon \\ 1/2 & 1/3+\varepsilon & 1/4 \\ 1/3+\varepsilon & 1/4 & 1/5+\delta \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 2 \end{bmatrix},$$

entonces la solución exacta de este sistema perturbado es:

$$y_1 = \frac{69 - 2448\varepsilon - 4320\varepsilon^2 + 7308\delta + 2160\varepsilon\delta}{1 + 252\varepsilon - 2160\varepsilon^2 - 2160\varepsilon^3 + 1808\delta + 2160\varepsilon\delta}$$

$$y_2 = \frac{-396 + 2880\varepsilon - 1088}{1 + 252\varepsilon - 2160\varepsilon^2 - 2160\varepsilon^3 + 1808\delta + 2160\varepsilon\delta}$$

$$y_3 = \frac{390 + 2880\varepsilon - 2160\varepsilon^2}{1 + 252\varepsilon - 2160\varepsilon^2 - 2160\varepsilon^3 + 1808\delta + 2160\varepsilon\delta}$$

Los números de condición $K_{j,pq}$ para los valores de p y q donde se perturbó la matriz son:

$K_{1,13} = 390/23$	$K_{2,13} = 130/11$	$K_{3,13} = 10$
$K_{1,22} = 1584/23$	$K_{2,22} = 64$	$K_{3,22} = 792/13$
$K_{1,31} = 10$	$K_{2,31} = 115/11$	$K_{3,31} = 138/13$
$K_{1,33} = 780/23$	$K_{2,33} = 390/11$	$K_{3,33} = 36$

Con este enfoque, si todos los números de condición K_{jk} y $K_{j,pq}$ son pequeños, entonces se dice que un sistema algebraico del tipo (2.1), es bien condicionado. Caundo algunos de ellos son grandes, se dice que el sistema es mal condicionado. El concepto de pequeño o grande es relativo y depende de la aplicación.

Los números de condición son pequeños bajo este punto de vista, cuando el det(A) no es mucho más pequeño que los términos individuales de la suma siguiente:

$$\Delta = \det(A) = \sum_{i=1}^{n} A_{ij} a_{ij},$$

o bién cuando el lado derecho b se elige de manera que las entradas del vector solución no son pequeñas.

Este tratamiento es impráctico pues requiere del cálculo de un gran número de números de condición para un solo sistema algebraico.

En algún tiempo se pensó que el determinante estaba en estrecha relación con la condición del sistema, sin embargo, por una parte, existen sistemas bien comportados donde el determinante es pequeño y por otra; sistemas donde la matriz es mal comportada y el determinante no es pequeño. Un ejemplo de la primera observación es una matriz diagonal con entradas igual a 10^{-6}, el determinante de la matriz es de 10^{-6n}, y sin embargo es fácil de obtener su solución. La matriz siguiente tiene un determinante igual 1 y sin embargo es muy mal condicionada.

(2.5)
$$\begin{bmatrix} 1 & & & & \\ -1 & 1 & & & \\ -1 & -1 & 1 & & \\ \vdots & \vdots & \vdots & \ddots & \\ \vdots & \vdots & \vdots & & \ddots \\ -1 & -1 & -1 & \cdots & 1 \end{bmatrix}$$

Otra caracterización, propuesta por Von Newmann y Goldstine, [voi], es considerar el cociente

(2.6)
$$\frac{\max_i |\lambda_i|}{\min_i |\lambda_i|},$$

donde λ_i son los valores propios de la matriz del sistema. Valores grandes de este cociente indicarían que el sistema es mal comportado. Sin embargo la matriz definida en (2.5) es un contraejemplo a este intento, el cociente (2.6) es igual a 1.

Otra idea seductora es que, si la matriz es mal condicionada entonces, un elemento pequeño aparecerá en la diagonal de la matriz triangular superior, durante la eliminación Gaussiana con pivoteo parcial. Si la transpuesta de la matriz (2.5) es el factor triangular superior, obtenido en una factorización, entonces esta matriz, también, es un contraejemplo; claramente no tiene un elemento pequeño en la diagonal y sin embargo es mal condicionada.

A diferencia del tratamiento de Noble las tres últimas caracterizaciones

no proporcionan cotas sobre el error obtenido en la solución.

2.2. Normas de vectores y matrices.

El análisis de error y sensibilidad del problema (2.1), se hace actualmente a partir de las normas de matrices y vectores. El propósito de esta sección es presentar algunos resultados de la teoría de normas necesarios para los capítulos posteriores.

Las normas de vectores empleadas serán las p-normas, definidas como:

$$\|x\|_p = \left[\sum_{j=1}^{n} |x_j|^p \right]^{1/p} \quad \text{para} \quad 1 \le p \le \infty,$$

Las normas de matrices más empleadas en análisis numérico son: la norma de Frobenius

$$\|A\|_F = \left[\sum_{i=1}^{n} \sum_{j=1}^{n} |a_{ij}|^2 \right]^{1/2}, \tag{2.7}$$

y las normas inducidas por las p-normas de vectores

$$\|A\|_p = \max_x \frac{\|Ax\|_p}{\|x\|_p}. \tag{2.8}$$

La normas empleadas frecuentemente en el estudio y estimación del número de condición son $\| \ \|_1$, $\| \ \|_2$, $\| \ \|_\infty$ y la de Frobenius. La primera y la tercera tienen preferencia en los cálculos numéricos, pues sólo requieren de operaciones elementales para su evaluación, mientras que las otras dos presentan facilidades en la demostración de algunos resultados.

Las normas anteriores tienen las propiedades siguientes, las cuales también son válidas para matrices rectangulares m x n. Una buena referencia para las demostraciones es [Tol].

(I) $$\|A\|_1 = \max_{1 \le j \le n} \sum_{i=1}^{m} |a_{ij}|.$$

La norma 1 de una matriz es igual al máximo de las sumas de los valores absolutos por columna.

(II) $$\|A\|_2 = \sqrt{\lambda_{max}(A^*A)} = \sigma_{max}(A).$$

La norma 2 es igual a la raíz cuadrada positiva del valor propio más grande de la matriz A^*A (o AA^*), también conocido como el valor singular más grande de A. A^* es en el caso complejo, la transpuesta conjugada de A y en el caso real, es la transpuesta de A. En la literatura esta norma también se conoce como la norma espectral de A.

(III) $$\|A\|_\infty = \max_{1 \le i \le m} \sum_{j=1}^{n} |a_{ij}|.$$

Esta propiedad es similar a la de la norma 1, sólo que el máximo se toma sobre la suma de los valores absolutos de los renglones.

Otra relación importante, que relaciona la norma espectral y la norma de Frobenious es

(IV) $$\|A\|_F = [\text{Traza}(A^*A)]^{1/2} = [\text{Traza}(AA^*)]^{1/2}$$

$$= \left(\sum_{i=1}^{r} \sigma_i^2(A) \right)^{1/2}, \quad r = \text{rango de A.}$$

Estas normas cumplen, además, dos propiedades adicionales que facilitan el análisis de error de los sistemas algebraicos lineales. La primera, conocida como **consistencia de las normas de matrices** es

(2.9) $$\|AB\|_1 \le \|A\|_2 \|B\|_3,$$

donde $\|\|_1$, $\|\|_2$ y $\|\|_3$ son normas inducidas p o la norma de F. Segunda, la **consistencia de la norma de la matriz respecto a una norma p de vectores**:

(2.10) $$\|Ax\|_v \le \|A\|_1 \|x\|_v \quad \text{para toda x y toda A,}$$

donde $\|\|_1$ al igual que en el caso anterior, es una norma inducida o la norma F. Las propiedades (2.9) y (2.10) no necesariamente se cumplen para cualquier norma, por ejemplo, en la norma máxima

$$\|A\|_M = \max_{i,j} |a_{ij}|$$

las matrices $A = B = \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}$ no la satisfacen, pues $\|AB\|_M = 2$, mientras $\|A\|_M \cdot \|B\|_M = 1$.

Es bien conocido que dos normas en un espacio lineal de dimensión finita son equivalentes, es decir, para cualquier par de normas de matrices $\|\|_p$ y $\|\|_q$ existen constantes positivas $c_1(p, q)$ y $c_2(p, q)$ tales que

(2.11) $$c_1\|A\|_p \le \|A\|_q \le c_2\|A\|_p .$$

Zielke, en [Zi2], proporciona las constantes c_1 y c_2 para varias normas de matrices rectangulares y que coinciden con los resultados ya conocidos para matrices cuadradas. En la tabla (2.1) se presentan las constantes para el caso de las cuatro normas de interés.

Las constantes arriva mencionadas, son las mejores posibles, es decir, para la desigualdad (2.11) y cualquier par (p, q) existe una matriz A, donde la igualdad se cumple. Algunas desigualdades importantes pueden obtenerse de la tabla, por ejemplo, para estimar la norma espectral de A pueden emplearse alguna de las relaciones siguientes.

(2.12) $$\frac{1}{\sqrt{n}}\|A\|_1 \le \|A\|_2 \le \sqrt{n}\,\|A\|_1$$

(2.13) $$\frac{1}{\sqrt{n}}\|A\|_\infty \le \|A\|_2 \le \sqrt{n}\,\|A\|_\infty$$

(2.14) $$\frac{1}{\sqrt{n}}\|A\|_F \le \|A\|_2 \le \|A\|_F$$

(2.15) $$\frac{1}{\sqrt{n}}(\|A\|_1\|A\|_\infty)^{1/2} \le \|A\|_2 \le (\|A\|_1\|A\|_\infty)^{1/2}$$

	C_1			
$\|A\|_q$	$\|A\|_p = \|A\|_1$	$\|A\|_2$	$\|A\|_\infty$	$\|A\|_F$
$\|A\|_1$	-	$\frac{1}{\sqrt{n}}$	$\frac{1}{n}$	$\frac{1}{\sqrt{n}}$
$\|A\|_2$	$\frac{1}{\sqrt{m}}$	-	$\frac{1}{\sqrt{n}}$	$\frac{1}{\sqrt{r}}$
$\|A\|_\infty$	$\frac{1}{m}$	$\frac{1}{\sqrt{m}}$	-	$\frac{1}{\sqrt{m}}$
$\|A\|_F$	$\frac{1}{\sqrt{m}}$	1	$\frac{1}{\sqrt{n}}$	-
	C_2			
$\|A\|_1$	-	$\sqrt{m}$	m	$\sqrt{m}$
$\|A\|_2$	$\sqrt{n}$	-	$\sqrt{m}$	1
$\|A\|_\infty$	n	$\sqrt{n}$	-	$\sqrt{n}$
$\|A\|_F$	$\sqrt{n}$	$\sqrt{r}$	$\sqrt{m}$	-
r = rango de la matriz A.				

Tabla (2.1). Constantes $c_1(p,q)$ y $c_2(p,q)$ para la equivalencia entre las normas 1, 2, F e ∞ de matrices nxm.

Ejemplo. Sea A la inversa de la matriz de Hilbert de orden 3,

$$A = \begin{bmatrix} 9 & -36 & 30 \\ -36 & 192 & -180 \\ 30 & -180 & 180 \end{bmatrix}.$$

Las diferentes normas de la matriz son $\|A\|_1 = \|A\|_\infty = 408$, $\|A\|_2 = 372.1156$ y $\|A\|_F = 372.2056$. Con las desigualdades (2.12), (2.13), (2.14) y (2.15) se obtiene

$$235.5589 \leq \|A\|_2 \leq 408,$$

$$235.5589 \leq \|A\|_2 \leq 408,$$

$$214.8930 \leq \|A\|_2 \leq 372.2056,$$

$$235.5589 \leq \|A\|_2 \leq 408.$$

Las cotas obtenidas a partir de la tabla 2.1, pueden resultar muy holgadas y en algunos casos imprácticas, sin embargo, es la única forma simple de estimarla.

Previo al análisis de sensitividad de (2.1) se presentan algunos resultados básicos de la teoría de normas.

El primero de ellos establece que la norma de la inversa de la matriz identidad perturbada, no se aleja mucho de 1 si la perturbación es pequeña.

Teorema 2.1. Sea E una matriz de orden n tal que $\|E\| < 1$, entonces I- E es no singular y

$$\|(I-E)^{-1}\| \leq (1 - \|E\|)^{-1}.$$

Dem: sea $x \neq 0$ entonces

$$\begin{aligned}\|(I-E)(x)\| &= \|x - Ex\| \geq \|x\| - \|Ex\| \\ &\geq \|x\| - \|E\|\,\|x\| \\ &\geq (1 - \|E\|)\|x\| > 0.\end{aligned}$$

Como $1 - \|E\| > 0$ y $x \neq 0$, entonces $(I-E)x \neq 0$ y por lo tanto E es no singular. Por otra parte, como

$$I = (I-E)(I-E)^{-1},$$

se tiene

$$\begin{aligned}1 = \|(I-E)(I-E)^{-1}\| &= \|(I-E)^{-1} - E(I-E)^{-1}\| \\ &\geq \|(I-E)^{-1}\| - \|E(I-E)^{-1}\| \\ &\geq \|(I-E)^{-1}\| - \|E\|\,\|(I-E)^{-1}\|,\end{aligned}$$

entonces $\|(I-E)^{-1}\| \leq (1 - \|E\|)^{-1}$ □

En algunos libros de texto, se pasa un detalle por alto, en la demostración anterior; no necesariamente la norma de la matriz identidad es 1. Por ejemplo, la norma de Frobenious de la matriz identidad es $(n)^{1/2}$. Este resultado es válido solamente para las normas inducidas.

Colorario 2.1. Si $\|E\| < 1$ entonces

$$\| I - (I - E)^{-1}\| \leq \frac{\|E\|}{1 - \|E\|}$$

Dem: Del teorema anterior se tiene que la matriz (I-E) es no singular,

por lo tanto

$$(I - E)(I - E)^{-1} = I,$$

$$(I - E)^{-1} = I + E(I - E)^{-1},$$

$$I - (I - E)^{-1} = -E(I - E)^{-1},$$

$$\|I - (I - E)^{-1}\| \leq \|E\| \, \|(I - E)^{-1}\|.$$

Aplicando el resultado del teorema anterior se obtiene

$$\|I - (I - E)^{-1}\| \leq \frac{\|E\|}{1 - \|E\|} \qquad \square$$

Teorema 2.2. Sea A una matriz no singular y $\|A^{-1}E\| < 1$, entonces $A + E$ es no singular y $(A + E)^{-1}$ puede ser escrita de la forma:

$$(A + E)^{-1} = (I + F)A^{-1}$$

donde: $\|F\| \leq \dfrac{\|A^{-1}E\|}{1 - \|A^{-1}E\|}$ y además

$$\frac{\|A^{-1} - (A + E)^{-1}\|}{\|A^{-1}\|} \leq \frac{\|A^{-1}E\|}{1 - \|A^{-1}E\|}$$

Dem: del teorema 2.1 se tiene que $I - A^{-1}E$ es no singular y por lo tanto, $A + E = A(I - A^{-1}E)$ es también no singular, además

$$\begin{aligned}(A + E)^{-1} &= (A\,(I + A^{-1}E))^{-1} \\ &= (I + A^{-1}E)^{-1}A^{-1} \\ &= (I + (-I + (I + A^{-1}E)^{-1}))A^{-1}.\end{aligned}$$

Si se define $F = -I + (I - A^{-1}E)^{-1}$ y se aplica el corolario (2.1) a la matriz A^{-1}, se obtiene

$$\|F\| = \|I - (I + A^{-1}E)^{-1}\| \leq \frac{\|A^{-1}E\|}{1 - \|A^{-1}E\|}.$$

Ahora como $(A + E)^{-1} = (I + F)A^{-1}$

$$= A^{-1} + A^{-1}F,$$

$$\|A^{-1} - (A + E)^{-1}\| = \|A^{-1}F\|$$

$$\leq \|A\| \, \|F\|,$$

entonces

$$\frac{\|A^{-1} - (A+E)^{-1}\|}{\|A^{-1}\|} \leq \|F\| \qquad \square$$

Colorario 2.2. Si $K(A) = \|A\|\,\|A^{-1}\|$ y además $\|A^{-1}\|\,\|E\| < 1$, entonces.

$$\|F\| \leq \frac{K(A)\frac{\|E\|}{\|A\|}}{1 - K(A)\frac{\|E\|}{\|A\|}},$$

y

$$\frac{\|A^{-1} - (A+E)^{-1}\|}{\|A^{-1}\|} \leq \frac{K(A)\frac{\|E\|}{\|A\|}}{1 - K(A)\frac{\|E\|}{\|A\|}}$$

Dem: del teorema anterior se tiene que

$$\|F\| - \|F\|\,\|A^{-1}E\| \leq \|A^{-1}E\|,$$

$$\|F\| \leq \|A^{-1}E\| + \|F\|\,\|A^{-1}E\|$$

$$\leq \|A\|\,\|E\| + \|F\|\,\|A^{-1}\|\,\|E\|,$$

$$\|F\|(1 - \|A^{-1}\|\,\|E\|) \leq \|A^{-1}\|\,\|E\|,$$

$$\|F\| \leq \frac{\|A^{-1}\|\,\|E\|}{1 - \|A^{-1}\|\,\|E\|},$$

$$\|F\| \leq \frac{K(A)\frac{\|E\|}{\|A\|}}{1 - K(A)\frac{\|E\|}{\|A\|}}. \quad \square$$

La demostración de la segunda parte del corolario es similar a la demostración de la segunda parte del teorema 2.2.

2.3 Análisis de sensibilidad del problema $Ax = b$.

De acuerdo al análisis retrospectivo de Wilkinson [Wi3], resolver el sistema algebraico (2.1) por medio de una computadora digital, con aritmética de punto flotante, es equivalente a resolver exactamente el sistema perturbado

(2.16) $\qquad (A + \delta A)\,y = b + \delta b,$

donde las perturbaciones dependen, principalmente, del método y la computadora empleada. Determinar la magnitud de las perturbaciones es el cometido del análisis de error.

Este tipo de análisis, aunque importante, queda fuera del alcance de este trabajo, sólo se aborda el análisis de sensibilidad de la solución respecto a pequeñas perturbaciones de los datos del problema; para el problema (2.1) son la matriz A y el vector b.

El análisis se sensibilidad para el problema (2.1), se dividió en tres casos:

Caso I. Las perturbaciones afectan solamente el lado derecho de (2.1). La relación (2.16) se transforma en

(2.17) $$Ay = b + \delta b.$$

Substituyendo (2.1) en (2.17), se obtiene

$$y - x = A\delta b.$$

Por lo tanto

$$\|y - x\| = \|A^{-1}\delta b\|.$$

Aplicando (2.10), se obtiene una cota del error en término de la norma de la inversa de A,

$$\|y - x\| \leq \|A^{-1}\| \, \|\delta b\|.$$

Finalmente, aplicando la desigualdad $1/\|x\| \leq \|A\| / \|b\|$ se obtiene,

(2.18) $$\frac{\|y - x\|}{\|x\|} \leq \|A\| \, \|A^{-1}\| \, \frac{\|\delta b\|}{\|b\|},$$

una cota del error relativo de la solución del sistema $Ax = b$, al perturbar ligeramente el lado derecho. La cantidad $K(A) = \|A\| \, \|A^{-1}\|$, se conoce como el número de condición de la matriz A. El número de condición, en este caso, da el factor de magnificación de la perturbación. Cuando este número es relativamente grande se dice que el sistema es mal condicionado.

Por ejemplo, si K(A) es del orden de 10^6 y la perturbación relativa de b es del orden de la precisión de la computadora, por ejemplo 10^{-7}, entonces la solución del sistema tendrá una precisión de, a lo más, 2 dígitos de precisión.

Caso II. Las pertubaciones sólo afectan a la matriz A. El sistema perturbado es

(2.19) $$(A + \delta A)\, y = b,$$
$$(A + \delta A)\,(y - x) = b - (A + \delta A)x,$$

substituyendo (2.1) en (2.19) se obtiene

$$(A + \delta A)\,(y - x) = -\delta A\, x,$$

$$y - x = (A + \delta A)^{-1}(-\delta A\, x).$$

Por lo tanto

$$\|y - x\| = \|(A + \delta A)^{-1}(-\delta A\, x)\|.$$

Si $\|A^{-1}\delta A\| < 1$ y aplicando la propiedad (2.10) y el teorema (2.2) resulta

$$\|y - x\| \le \|A^{-1}\|(1 + \|F\|)\,\|\delta A\|\,\|x\|,$$

$$\|y - x\| \le \frac{\|A^{-1}\|}{1 - \|A^{-1}\delta A\|}\,\|\delta A\|\,\|x\|.$$

Dividiendo cada miembro de la última desigualdad, por $\|x\|$, se obtiene una cota del error relativo de la solución de (2.19),

$$\frac{\|x - y\|}{\|x\|} \le \frac{K(A)}{1 - \|A^{-1}\delta A\|}\left(\frac{\|\delta A\|}{\|A\|}\right). \tag{2.20}$$

Caso III. Caso general. Las perturbaciones afectan tanto a la matriz A como al vector b. De la relación (2.16) se tiene que

$$(A + \delta A)\, y = b + \delta b,$$

$$(A + \delta A)\, y - (A + \delta A)x = b + \delta b - (A + \delta A)x,$$

$$(A + \delta A)(x - y) = \delta b - \delta A x,$$

$$(x - y) = (A + \delta A)^{-1}(\delta b - \delta A x).$$

Por lo tanto

$$\|x - y\| = \|(A + \delta A)^{-1}(\delta b - \delta A x)\|.$$

De la propiedad de consistencia y del teorema (2.2) se obtiene

$$\|x - y\| \le \|I + F\|\|A^{-1}\|(\|\delta b\| + \|\delta A\|\|x\|)$$

$$\le \frac{\|A^{-1}\|}{1 - \|A^{-1}\delta A\|}(\|\delta b\| + \|\delta A\|\|x\|).$$

Finalmente, de la desigualdad $1/\|x\| \le \|A\| / \|b\|$ se obtiene una cota del error relativo de la solución

$$\frac{\|x - y\|}{\|x\|} \le \frac{K(A)}{1 - \|A^{-1}\delta A\|}\left(\frac{\|\delta b\|}{\|b\|} + \frac{\|\delta A\|}{\|A\|}\right). \tag{2.21}$$

Además si $\|A^{-1}\|\|\delta A\| < 1$, entonces la relación (2.21) se transforma en

$$\frac{\|x - y\|}{\|x\|} \leq \frac{K(A)}{1 - K(A)\frac{\|\delta A\|}{\|A\|}} \left(\frac{\|\delta b\|}{\|b\|} + \frac{\|\delta A\|}{\|A\|} \right) \tag{2.22}$$

2.4 Sistemas efectivamente bien condicionados.

En la sección anterior se demostró que la sensibilidad de la solución, del problema $Ax = b$, está relacionada estrechamente con la magnitud del número de condición del problema de inversión, $K(A) = \|A\|\|A^{-1}\|$.

Existen, sin embargo, algunos sistemas donde las cotas de error pueden sobreestimar considerablemente la sensibilidad real de la solución del sistema, por ejemplo, en sistemas algebraicos lineales, donde A es la matriz de Vandermonde; con nodos $x_1 > x_2 > \ldots > x_n > 0$ y el lado derecho con la propiedad: $b_i = (-1)^i b_i \neq 0$, para $i = 1, 2, \ldots, n$. La sensibilidad de la solución, para este tipo de sistemas, no crece exponencialmente como lo establece el número de condición; más aún, Higham demuestra en [Hi5] que, si el algoritmo de Björck-Pereyra [Bj1] es empleado para resolver este tipo de sistemas, el error relativo producido en la componentes diferentes de cero del vector solución x, es independiente de K(A).

Tony Chan en [Ch1] llama a estos problemas; **problemas efectivamente bien condicionados**, y caracteriza la sensibilidad de la solución no sólo en términos de A sino también del lado derecho.

A partir de la descomposición en valores singulares, puede demostrarse que si $A x = b$ es el sistema exacto y $(A + \delta A) y = b + \delta b$ el perturbado, entonces

$$\frac{\|\delta x\|}{\|x\|} \leq \frac{\sigma_{n+1-k}}{\sigma_n} \left(\frac{\|P_k b\|}{\|b\|} \right)^{-1} \frac{\|\delta b\|}{\|b\|} \quad \text{para } 1 \leq k \leq n, \tag{2.23}$$

en donde, $P_k = U_k U_k^t$, $1 \leq k \leq n$, es el operador proyección, con $U_k = [u_{n+1-k}, \ldots, u_n] \in \mathbb{R}^{n \times k}$. Cuando $k = n$ se tiene una desigualdad ya conocida.

La relación (2.23) establece que la sensitividad de la solución no depende sólo de los tamaños relativos de los valores singulares, sino también de la proyección de b sobre los valores singulares izquierdos de A. Si b tiene una fuerte componente en el subespacio definido por U_k, entonces la sensibilidad de la solución puede disminuir por el factor $\|b\|/\|P_k b\|$.

Para matrices de Vandermonde con nodos positivos y diferentes, como los

sistemas mencionados anteriormente, sus vectores singulares u_k y v_k tienen $k - 1$ cambios de signo; en particular, el vector singular izquierdo v_n tiene los mismos cambios de signo que el lado derecho propuesto.

Este y otros ejemplos sugieren que la definición de número de condición, para el problema (2.1), no necesariamente es la más adecuada para medir la sensibilidad real del problema. Una alternativa es retomar la definición original de Rice, (1.3), y hacer supuestos sobre el tipo de perturbaciones, que pueden presentar los datos e introducir en el análisis de sensibilidad la información sobre la estructura de la matriz. Con este tratamiento pueden obtenerse cotas de error más aproximadas.

Una posibilidad es emplear el número de condición local K(A, x), definido como

(2.24) $$K(A, x) = (\|Ax\|/\|x\|)\|A^{-1}\|.$$

Claramente, K(A) es el supremo del número de condición local, es decir,

$$K(A) = \|A^{-1}\| \sup_x (\|Ax\|/\|x\|) = \|A^{-1}\|\|A\|.$$

Si $\|Ax\|$ y $\|x\|$ son del mismo orden de magnitud entonces, K(A, x) es esencialmente $\|A^{-1}\|$, además el número de condición local puede ser moderado aunque K(A) sea grande.

Skeel, en [Sk1], considera el sistema $Ax = b$, con A una matriz cuadrada no singular sujeta a perturbaciones de la forma $A \to A + E$, $|E| \le \epsilon|A|$, y $b \to b + d$, $|d| \le \epsilon|b|$, donde $|.|$ denota la operación de remplazar cada elemento de un vector o una matriz por su valor absoluto. Para perturbaciones que sólo afectan la matriz A, Skeel introduce el número de condición

(2.25) $$\text{cond}(A, x) = \lim_{\epsilon \to 0} \sup_{|E| \le \epsilon|A|} \frac{\|\delta x\|_\infty}{\|x\|_\infty},$$

donde $(A + E)(x + \delta x) = b$. Y demuestra que

$$\text{cond}(A, x) = \frac{\||A^{-1}||A||x|\|_\infty}{\|x\|_\infty}.$$

El valor máximo de cond(A, x) = cond(A, e)= $\||A^{-1}||A|\|_\infty$. Una de las propiedades más importantes de este número, es la invariancia bajo escalamientos de renglones, de hecho, si $D = \text{diag}(d_i)$ es una matriz diagonal se cumple la relación

$$|(DA)^{-1}||DA| = |A^{-1}||D^{-1}||D||A| = |A^{-1}||A|$$

Este número de condición es empleado por Higham en [Hi6] para predecir el comportamiento del error hacia adelante en la solución del sistema triangular $Tx = b$; concluye que la solución de sistemas triangulares depende del lado derecho, y puede ser muy aproximada, sin importar el tamaño del número de condición de la matriz. Por ejemplo, si T es una matrix triangular superior con

$$|t_{ii}| \geq |t_{ij}| \text{ para } j > i,$$

entonces la matriz triangular $W = |T^{-1}||T|$ satisface $w_{ij} \leq 2^{j-1}$ para $j > i$, es decir, cond(T) está acotado por un valor fijo n, sin importar el valor de $K(T) = \|T^{-1}\|\|T\|$.

De este capítulo se concluye que el número de condición de la matriz A, $K(A) = \|A\|\|A^{-1}\|$, está relacionado estrechamente con el factor de magnificación de la incertidumbre presente en los datos del sistema -matriz A y vector b-. Sin embargo, las cotas de error obtenidas en (2.3), pueden sobreestimar considerablemente la sensitividad real de la solución para ciertos sistemas. La estimación puede mejorar cuando se toma en cuenta el lado derecho del sistema o el tipo de perturbaciones que pueden presentarse en los datos.

CAPITULO 3

EL NUMERO DE CONDICION DE POLINOMIOS

3.1 Evaluación de polinomios
3.2 Representación de polinomios
3.3 Generación de polinomios ortogonales.

Los polinomios son empleados como un medio para aproximar en casi todas las áreas del análisis numérico, y proveen de una herramienta matemática para el desarrollo de métodos en la teoría de aproximaciones, integración numérica y solución numérica de ecuaciones diferenciales e integrales.

Los polinomios son, sin duda, las funciones más empleadas en el análisis numérico y existe un gran número de trabajos donde se estudian las propiedades de estas funciones, inclusive algunos de los problemas clásicos de polinomios ya tienen más de un siglo de haberse planteado. Sin embargo en contraste con el amplio número de propiedades matemáticas conocidas, está el reducido conocimiento de sus propiedades numéricas; no fué sino hasta mediados de este siglo que empezaron a estudiarse las dificultades numéricas que se presentan en algunos problemas relacionados con polinomios.

En este capítulo se abordan tres problemas mal condicionados relacionados con polinomios: evaluación de polinomios, representación de polinomios y generación de polinomios ortogonales. Además se presentan algunos algoritmos estables para resolverlos.

3.1 Evaluación de polinomios.

Wilkinson [Wil], a principios de los años cincuenta encontró, al realizar un experimento numérico, que evaluar un polinomio, inclusive de grado modesto, puede resultar un proceso difícil, es decir, que el resultado obtenido puede diferir considerablemente del valor exacto.

Un ejemplo de esta observación, es el polinomio

$$P(x) = \prod_{i=1}^{10} (x - i), \tag{3.1}$$

el cual tiene todas sus raices reales y diferentes, además no están cercanas entre si. El experimento consiste en generar una tabulación del polinomio P(x), expresado en forma de potencias, en [0, 10] con pasos de 0.1. Si el experimento se reliza en una computadora IBM-PC o compatible, las cantidades son representadas en punto flotante de precisión sencilla (23 bits de mantisa), la aritmética se realiza a 64 bits,[1] y además si se emplea, en la evaluación del polinomio, la regla de Horner o también conocida como multiplicación anidada, entonces los valores calculados del polinomio, en los ceros del polinomio, son los que aparecen en la tabla (3.1).

x	P(x)
1.000000	0.317413
2.000000	-2.432587
2.999999	0.865173
3.999998	-10.735942
4.999998	137.355286
5.999997	-107.856201
6.999996	381.536987
7.999995	877.128662
8.999998	-909.637024
10.000002	-706.004761

Tabla 3.1. Valor del polinomio $P(x) = \prod_{i=1}^{10} (x - i)$ en $x = \sum_{i=1}^{n} .1$, con n = 10, 20, ..., 100

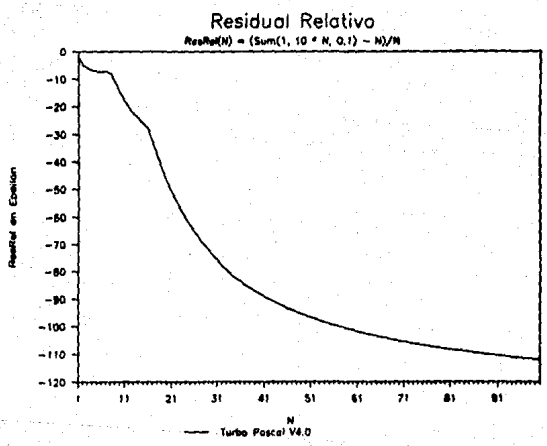
El número real 0.1 tiene una representación binaria periódica, por lo tanto, su representación de punto flotante es aproximada. La representación de punto flotante de 0.1, denotada por float(0.1), depende de la aritmética empleada, por ejemplo, en Turbo C (V2.0) y Fortan (V3.2), float(0.1) > 0.1. y en Turbo-Pascal con aritmética rápida, float(0.1) < 0.1

[1] Este tipo de aritmética, es compartido por varios paquetes de programación para las computadoras IBM y compatibles (XT y AT), y corresponde a la del coprocesador aritmético Intel-8087 o su emulador. Ambos realizan la aritmética a 63 bits de mantisa, 15 bits de exponente y un bit de signo para la mantisa. Ver [Int].

Al Sumar float(0.1) reiteradamente, se van acumulando los errores, de manera que $n_f = \sum_{i=1}^{10*n} float(0.1)$ puede diferir considerablemente de n. La diferencia entre n_f y n depende también de la aritmética, por ejemplo, el coprocesador Intel-8087 o emulador redondean el resultado de la suma antes de depositarlo en memoria. En las figuras (3.2) y (3.3) se presenta el residual relativo

$$ResRel(n) = \left(\sum_{i=1}^{n} float(0.1) - \frac{n}{10}\right) \Big/ \frac{n}{10},$$

para el problema de sumar float(0.1) iteradamente, con la aritmética rápida de Turbo-Pascal (V4.0) y la del coprocesador Intel-8087 respectivamente.



Figura(3.2) ResRel(n) con aritmética Turbo-Pascal

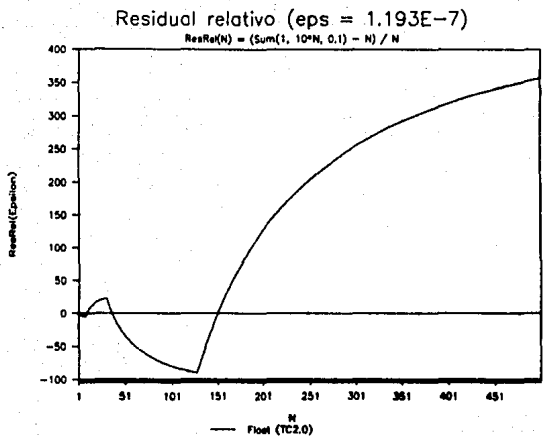


Figura 3.3. ResRel(n). Con aritmética del Coprocesador artimético Intel-8087 o emulador.

El experimento muestra dos aspectos importantes que se presentan al evaluar una función.

Primero: la precisión del punto de evaluación; las graficas (3.2) y (3.3), revelan que el punto de avaluación puede estar lejos del valor real; en este caso, se debe a los errores acumulados al sumar reiteradamente float(0.1).

Segundo: Para justificar el comportamiento observado al evaluar el polinomio, basta hacer un análisis de error simple, pero suficiente.

La evaluación del polinomio

$$p(x) = a_0 + a_1x + a_2x^2 + \ldots x^n$$

en $x = z$ se realiza por medio de la regla de Horner. Los pasos del algoritmo son

$$s_n = 1,$$

$$s_{n-1} = z * s_n + a_{n-1},$$

$$\ldots$$

$$s_0 = z * s_1 + a_0,$$

y $$p(z) = s_0.$$

Si los cálculos se realizan en forma exacta, excepto, en el k-ésimo paso, donde el error introducido es c (en la aritmética en que se realizó el experimento, c corresponde al error de redondeo de la aritmética), entonces, la solución calculada s_k^* satisface la relación:

$$s_n^* = 1$$

$$s_p^* = z * s_{p+1}^* + a_p \qquad , p \neq k$$

y $$s_k^* = z * s_{k+1}^* + a_k + c.$$

Claramente

$$p^*(z) = p(z) + c * x^k.$$

El valor calculado y el exacto son iguales, hasta antes del paso k. El error, presente en el paso k, puede interpretarse como el evaluar exactamente un polinomio que difiere de p en su k-ésimo coeficiente. Aunque esta observación es trivial, tiene importancia práctica, pues muestra que los errores producidos en los cálculos son equivalentes, en su efecto, a perturbar los coeficientes originales del polinomio y realizar el cálculo en forma exacta.

Para realizar el análisis de error completo, debe considerarse que en cada paso del algoritmo, puede presentarse un error.

$$s_n^* = 1$$

$$s_{n-1}^* = z * s_n^* + a_{n-1} + c_{n-1}$$

$$\cdots$$

$$s_0^* = z * s_1^* + a_0 + c_0$$

$$p^*(z) = s_0^*$$

Siguiendo un razonamiento similar, puede demostrarse que:

$$p^*(z) = p(z) + q(z)$$

donde

$$q(z) = c_0 + c_1 * z + \cdots + c_{n-1} * z^{n-1}.$$

De esta manera, evaluar un polinomio por medio de una computadora, es equivalente a evaluar exactamente un polinomio con los coeficientes perturbados. La magnitud de la perturbación $e_k \approx s_k(1 + c_f)$, donde c_f es el epsilón de la aritmética empleada (en este ejemplo $c \approx 1.192093 \times 10^{-7}$).

De acuerdo con la definición 1.1, el problema de evaluar este polinomio cerca de sus raíces, es un problema mal condicionado, pues

$$K(P, x) = |x| \sum_{i=1}^{10} \frac{1}{|x - i|},$$

es grande para valores cercanos a las raíces.

Existen varias alternativas para evaluar adecuadamente polinomios mal condicionados. Una posibilidad es emplear aritmética de multiple precisión (por ejemplo, ACRITH, [JAi]) o bien aritmética exacta (paquetes: DERIVE y REDUCE, [Rai]). Alguna de estas opciones es conveniente cuando se requiere un reducido número de evaluaciones del polinomio; pues el tiempo empleado por el algoritmo es considerablemente mayor.

Otra disyuntiva es emplear algoritmos especializados; que aprovechen ciertas propiedades de los polinomios. Como por ejemplo, si el polinomio puede escribirse como un producto de dos o más polinomios, entonces, la acumulación de los errores puede reducirse al realizar la evaluación de cada factor por separado.

Una posibilidad más es cambiar la representación del polinomio, es decir, expresarlo como una combinación lineal de una base de polinomios diferente a donde el problema es mal condicionado. El trabajo adicional, que sólo se realiza una sola vez, es transformar el polinomio original a la nueva base.

Rice en [Ri2] da algunas de las representaciones más comunes, su calidad numérica en relación al problema de la evaluación y su complejidad. Para ejemplificar esta opción, considérese la representación de un polinomio en la base de los polinomios de Newton. El polinomio de Newton de orden n es construido a partir de n números reales $c_1, c_2, \cdots c_n$.

$$N_0(x) = 1$$

$$N_1(x) = N_0(x)(x - c_1) = (x - c_1)$$

$$N_2(x) = N_1(x)(x - c_2) = (x - c_1)(x - c_2)$$

$$\cdots$$

$$N_n(x) = N_{n-1}(x)(x - c_n) = (x - c_1)(x - c_2) \ldots (x - c_n)$$

Dado un polinomio mónico, p(x), éste puede escribirse como una combinación lineal de estos polinomios, es decir:

(3.2) $$p(x) = b_0N_0 + b_1N_1(x) + \ldots + b_nN_n(x)$$

Cuando $c_1 = c_2 = \ldots = c_n$, (3.2) se conoce como la representación de p(x) con desplazamiento. Si los c_i's son las raíces de p(x) la representación se conoce como el producto de raíces.

En la tabla (3.2) se presenta la evaluación, en diferentes representaciones, del polinomio definido en (3.1). Los puntos de evaluación son los mismos que los empleados en la tabla (3.1).

X	"Exacto"	Horner	Desp.(c=7)	Raíces
1.000000	0.043259	0.317413	-0.000000	-0.043259
2.000000	0.009613	-2.432587	-0.000000	0.009613
2.999999	0.007210	0.865173	-0.035370	0.007210
3.999998	-0.007210	-10.735942	0.005974	-0.007210
4.999998	0.006866	137.355286	0.006302	0.006866
5.999997	-0.009613	-107.856201	-0.009537	-0.009613
6.999996	0.018539	381.536987	0.018539	0.018539
7.999995	-0.052872	877.128662	-0.052932	-0.052871
8.999998	0.076904	-909.637024	0.078674	0.076904
10.000002	0.692139	-706.004761	0.683900	0.692142

Tabla 3.2. Evaluación del polinomio definido en (3.1) en diferentes representaciones.

La segunda columna etiquetada como "Exacto" corresponde a la evaluación de la aproximación lineal del polinomio cerca de cada una de las raíces, es decir,

$$p(i \pm \varepsilon) \approx m_i(i \pm \varepsilon), \text{ para } i = 1, 2, \ldots, 10.$$

Existen otras bases de polinomios que pueden emplearse para evaluar un polinomio; algunas se tratarán en la sección siguiente. Para la elección de una base debe considerarse: la dificultad para expresar el polinomio en la nueva base y el número de operaciones requeridas para la evaluación en la nueva representación. El primer aspecto, aunque importante, sólo se realiza una vez, para un polinomio dado, mientras que el número de evaluaciones, del mismo polinomio, es regularmente grande.

El número de operaciones requeridas en la evaluación, depende de la representación; por ejemplo, para la base de potencias con el método de Horner, para la representación de potencias desplazadas y para la representación de producto de raíces, se requieren n multiplicaciones y n sumas. La evaluación vía los polinomios de Newton se realiza con 2n sumas y n productos.

3.2 Representación de polinomios.

Los polinomios son sin duda las funciones más empleadas en el análisis numérico y en cada aplicación debe seleccionarse la base más conveniente. La evaluación de polinomios, como se vio en la sección anterior, presenta mejores propiedades numéricas en ciertas bases de polinomios.

En esta sección se obtienen expresiones del número de condición para el problema de representación de polinomios. Este número proporciona una medida de cuanto se modifica el polinomio, en cierto intervalo, al perturbar ligeramente sus coeficientes, o de manera equivalente, como se modifican los coeficientes al perturbar ligerante el polinomio. Las bases que trataremos son: las de potencias, las ortogonales y las de Lagrange.

El problema de representación de polinomios puede enunciarse de la manera siguiente:

Sea $\mathbb{P}_{n-1}$ la clase de polinomios (reales) de grado menor o igual a n - 1, y $P_0, P_1, \dots P_{n-1}$ una base de $\mathbb{P}_{n-1}$. Para cualquier $P \in \mathbb{P}_{n-1}$ se denotará por $a_0, a_1, \dots, a_{n-1}$ los coeficientes del polinomio $P \in \mathbb{P}_{n-1}$ con respecto a la base $P_0, P_1, \dots P_{n-1}$. El problema es determinar el número de condición del mapeo lineal:

$$M_n : \mathbb{R}^n \longrightarrow \mathbb{P}_{n-1}[a, b]$$

$$(a_0, a_1, \dots a_{n-1}) \longrightarrow P(x) = \sum_{i=0}^{n-1} a_i * P_i(x).$$

Las normas empleadas, en $\mathbb{R}^n$ y $\mathbb{P}_{n-1}$, son $\|a\|_\infty = \max_i |a_i|$ y $\|P\|_\infty = \max_{a \le x \le b} |P(x)|$ respectivamente. Como M_n es un mapeo lineal su número de condición es $K(M_n) = \|M_n\|_\infty \|M_n^{-1}\|_\infty$.

3.2.1. Condición de polinomios ortogonales.

El condicionamiento de un problema está relacionado estrechamente con su representación. Una elección desafortunada de la representación, redituará en la dificultad para resolver el problema. Por ejemplo, el empleo de la base de potencias, en el problema de mínimos cuadrados, puede conducir a un sistema lineal mal condicionado. Una forma de superar el problema es cambiar a una base donde el condicionamiento del problema sea menor.

Esta sección está dedicada a la representación de polinomios por medio de polinomios ortogonales.

La parte sencilla de la estimación del número de condición es calcular $\|M_n\|$. Supongase que $\|a\|_\infty = 1$, entonces

$$|M_n(a)| = \left|\sum_{k=0}^{n-1} a_k \cdot P_k(x)\right| \le \sum_{k=0}^{n-1} |a_k| \cdot |P_k(x)|$$

$$\le \|a\|_\infty \cdot \sum_{k=0}^{n-1} |P_k(x)|.$$

Por lo tanto:

$$\|M_n\| = \max_{a \ne 0} \frac{|M_n a|}{\|a\|} \le \max_{a \ne 0} \sum_{k=0}^{n-1} |P_k(x)|. \tag{3.3}$$

Cálculo de $\|M_n^{-1}\|$. Por definición $M^{-1}(P) = a$, es decir, el mapeo inverso asocia a un polinomio sus coeficientes. Por ortogonalidad, (A.7), de la familia P_k, $k = 0, 1, \ldots$ se tiene que

$$a_k = \frac{1}{h_k} \int_a^b P(x)\, P_k(x)\, w(X)\, dx \qquad k = 0, 2, \ldots n-1,$$

$$|a_k| \le \frac{1}{h_k} \int_a^b |P(x)|\, |P_k(x)|\, w(x)\, dx \qquad k = 0, 2, \ldots n-1.$$

De aplicar la desigualdad de Schwartz, a la última desigualdad, resulta

$$|a_k| \le \frac{1}{h_k}\left[\int_a^b |P(x)|^2 w(x) dx \int_a^b |P_k(x)|^2 w(x)\, dx\right]^{1/2}.$$

Como la integral del lado derecho corresponde a la definición de h_k, (A.7), entonces la desigualdad anterior se transforma en

$$|a_k| \le \frac{1}{h_k}\left[\int_a^b |P(x)|^2 w(x)\, dx\, h_k\right]^{1/2},$$

$$\le \frac{1}{h_k}\left[\, \|P(x)\|_\infty^2 \int_a^b w(x)\, dx\, h_k \,\right]^{1/2},$$

$$\le \frac{1}{h_k}\, \|P\|_\infty\, (\mu_0 h_k)^{1/2} = \left[\frac{\mu_0}{h_k}\right]^{1/2} \|P\|_\infty,$$

donde $\mu_0 = \int_a^b w(x)\, dx$. Por lo tanto la expresión para la norma del mapeo inverso es

$$\|M_n^{-1}\|_\infty = \max_{0\le k\le n-1} \frac{|a_k|}{\|P\|_\infty} \le \max_{0\le k\le n-1} \left[\frac{\mu_0}{h_k}\right]^{1/2} \tag{3.4}$$

De (3.3) y (3.4) se obtiene una cota superior para el número de condición de estas representaciones.

$$K(M_n) = \|M_n\|\, \|M_n^{-1}\|_\infty \le \max_{a\le x\le b} \sum_{k=0}^{n-1} |P_k(x)| \max_{0\le k\le n-1} \left[\frac{\mu_0}{h_k}\right]^{1/2} \tag{3.5}$$

Al aplicar (3.5) a los polinomios de Chebyshev y de Legendre, (A.9) y (A.10), puede obtenerse una cota de su número de condición.

Polinomios de Chebyshev:

$$K(M_n) \le \max_{-1\le x\le 1} \sum_{k=0}^{n-1} |T_n(x)|\, (2)^{1/2} \le (2)^{1/2} n$$

Polinomios de Legendre:

$$K(M_n) \le n\, (2n-1)^{1/2}$$

3.2.2. Condición de polinomios en forma de potencias.

$P_k(x) = x^{k-1}$, $k = 1, 2, \ldots, n$.

El análisis de esta base presenta cierta dificultad y las cotas que se conocen para el número de condición son, en algunos casos, muy holgadas.

Sin perdida de generalidad, puede suponerse que los extremos del intervalo [a,b], satisfacen $|a| \le b$.

Cálculo de $\|M_n\|_\infty$.

$$\|M_n\| = \sup_{\|x\|=1} \max_{a\le x\le b} \left|\sum_{k=0}^{n-1} a_k x^{k-1}\right| = \sum_{k=0}^{n-1} b^{k-1}$$

Por lo tanto:

$$\|M_n\| = \frac{b^n - 1}{b - 1} \tag{3.5}$$

Cálculo de $\|M_n^{-1}\|$.

$$\|M_n^{-1}\| = \sup_{\|p\|\le 1} \max_{a\le x\le b} \frac{|p^{(k-1)}(0)|}{(k-1)!}$$

$$= \max_{a\le x\le b} \sup_{\|p\|\le 1} \frac{|p^{(k-1)}(0)|}{(k-1)!}.$$

Esta última relación puede reescribirse como:

$$\|M_n^{-1}\| = \max_{1\le k\le n} \|\lambda_k\|$$

Donde λ_k son funcionales lineales $\lambda_k : P_{n-1}[a,b] \longrightarrow R$, definidas por:

$$\lambda_k P = \frac{p^{(k-1)}(0)}{(k-1)!}.$$

Una de propiedades importantes de los polinomios de Chebyshev, (A9), es la extremalidad del polinomio de Chebyshev normalizado, $\tilde{T}_n$; entre todos los polinomios con el mismo término principal, $\tilde{T}_n$ tiene norma mínima en [-1, 1]. En 1878, Zolotarev, [Ca1], resolvió el problema correspondiente cuando dos coeficientes principales son dados. Este problema puede interpretarse como determinar la mejor aproximación por polinomios de grado menor o igual a n - 2 a funciones de la forma $x^n - \sigma x^{n-1}$, para un σ dado. Los polinomios con esta propiedad son conocidos como los polinomios de Zolotarev. Formalmente el problema planteado inicialmente por Zolotarev fué: determinar

$$\min_{q} \max_{-1\le x\le 1} |x^n - \sigma x^{n-1} + a_{n-2}x^{n-2} + \ldots + a_0|,$$

como una función de n y del parámetro σ, y encontrar el polinomio extremal. De acuerdo con [Ha2] el polinomio de Zolotarev de grado n y parámetro σ puede escribirse como:

$$Z_{n,\sigma} = sT_n + T_{n-1} + q^*_{n-2}, \tag{3.6}$$

donde T_n y T_{n-1} son polinomios de Chebyshev, s es una constante y q^*_{n-2} es el único polinomio de grado menor o igual a n-2 el cual minimiza:

$$\|Z_{n,\sigma}\|_\infty = \max_{-1\le x\le 1} |Z_{n,\sigma}(x)|.$$

En particular, si $s = 0$ el problema de minimización es resuelto con $q^{*}_{n-2} = 0$. Una relación explícita de los polinomios de Zolotarev aparece en [Ri4, pag.79].

El problema de calcular la norma de los funcionales λ_k está relacionado con el problema de Zolotarev . El polinomio extremal para el funcional λ_k es un polinomio de Zolotarev de grado n - 1 (Se está trabajando en el es espacio $P_{n-1}[a, b]$).

Desafortunadamente cuando el intervalo [a,b] es arbitrario, el valor del parámetro σ no es fácil de expresar y puede ser diferente para cada k. El caso que resulta fácil de abordar, es cuando $\sigma = 0$, en tal situación, el problema se reduce al intervalo [-w, w] y el polinomio se Zolotarev se transforma en la suma de un par de polinomios de Chebyshev, [Ga9]. Con esta simplificación puede obtenerse la norma de los funcionales λ_k:

$$\|\lambda_k\| = |T^{(k-1)}_{n-1}[-w, w](0) + T^{(k+1)}_{n-2}[-w, w](0)|/(k-1)!,$$

$$k = 1, 2, \ldots n,$$

donde $T_n[-w, w](x) = T_n(x/w)$. Así, la norma del mapeo inverso resulta:

$$\|M_n^{-1}\|_\infty = \max_{1 \le k \le n} \|\lambda_k\|_\infty = \|a_{T_{n-1}}[-w, w] + a_{T_{n-2}}[-w, w]\|_\infty , \tag{3.7}$$

donde a_{T_n} son los coeficientes del polinomios $T_n(x)$. La relación del número de condición del mapeo M_n para el intervalo [-w, w], se obtiene a partir de (3.5) y (3.7).

$$K_\infty(M_n) = \frac{w^n - 1}{w - 1} \max(\|a_{T_{n-1}}(x/w)\|_\infty, \|a_{T_{n-2}}(x/w)\|_\infty), \tag{3.8}$$

donde el factor $\frac{w^n - 1}{w - 1}$ toma el valor n cuando w = 1. Aplicando, (A.9), una aproximación asintótica de los polinomios de Chebyshev y la relación (3.8), se obtienen expresiones explícitas para el número de condición en el caso simétrico; cuando $w < 1$, $w = 1$ y $w > 1$.

$K_\infty(M_n)$	Intervalo [-w, w]
$(1 + \sqrt{1+w^2})^n$	$w > 1$
$(1 + \sqrt{2})^n$	$w = 1$
$[(1 + \sqrt{1+w^2})/w]^n$	$0 < w < 1$

En forma similar, cuando el intervalo de interés es [0, w], el número de condición para la función M_n también depende del valor de w. Relaciones para este caso fueron establecidas por Gautschi en [Ga9]:

$K_\infty(M_n)$	Intervalo [0, w]
$(2 + w + 2\sqrt{1 + w})^n$	$w > 1$
$(3 + 2\sqrt{2})^n$	$w = 1$
$[(2 + w + 2\sqrt{1 + w})/w]^n$	$0 < w < 1$

Finalmente, para obtener una estimación del número de condición en el caso general, Gautschi en [Gal0], emplea interpolación para obtener una cota superior de la norma de M_n^{-1}. Los puntos de interpolación, $s = (s_1, s_2, \ldots, s_n)$, son distintos y pertenecen al intervalo [a,b]. La relación obtenida es:

$$\|M_n^{-1}\|_\infty \le n\|V_n^{-1}(s)\|_\infty ,$$

donde $V_n(s)$ es la matriz de Vandermonde de orden n construida a partir del conjunto de nodos s. El problema consiste en minimizar $\|V_n^{-1}\|$ sobre todos los posible conjuntos de nodos s, para obtener una cota superior lo más cercana a $\|M_n^{-1}\|$.

Si los puntos de interpolación son elegidos como $s_\nu = \tau\cos\theta_\nu$, $\nu = 1, 2, \ldots, n$, con $\tau > 0$ y $\cos\theta_\nu$ son las raíces del polinomio de Chebyshev de orden n, $T_n(x)$, entonces [Gal0]:

$$K_\infty(M_n) \le \frac{3^{3/4}}{4(\sqrt{2} - 1)} \frac{2 + b - a}{2 + b + a} \frac{b^n - 1}{b - 1}\left(1 + \frac{b + a}{2}\right)^n \left|T_n\left(\frac{2i}{b - a}\right)\right|$$

Si los nodos de interpolación son elegidos como $s_\nu = \tau(1+ \cos\theta_\nu)$, $\nu = 1, 2, \ldots, n.$, con $\tau > 0$, entonces

$$K_\infty(M_n) \le \frac{b - a}{2(1 - |a|)\sqrt{1 - b - a}} \frac{b^n - 1}{b - 1}(1 + |a|)^n T_n\left(\frac{2}{b - a} + 1\right)$$

De acuerdo con [Gal0], la primer relación proporciona una mejor cota que la segunda cuando el intervalo [a, b] es cercano a un simétrico. En otro caso, la segunda relación proporciona una mejor cota que la primera.

3.2.4. Condición de las bases de Lagrange:

La sección de representación de polinomios concluye con las bases Lagrangianas, que presentan buen condicionamiento, inclusive en algunos casos mejor que el proporcionado por polinomios ortogonales. Los polinomios de Lagrange están definidos a partir de n nodos diferentes: $x_1, x_2, \ldots, x_n$ en el intervalo [a,b].

$$\ell_j(x) = \prod_{\substack{k=1 \\ k \neq j}}^{n} \frac{x - x_k}{x_j - x_k}, \quad k = 1, 2, \ldots n$$

A diferencia de las anteriores bases, los polinomios de Lagrange tienen el mismo grado, n - 1. La fórmula de interpolación de Lagrange está dada por:

$$P(x) = \sum_{k=1}^{n} y_k \ell_k(x),$$

donde los valores y_k, k = 1, 2, ..., n, son los valores de la función en los puntos x_k.

Para obtener una cota del número de condición de esta representación, se supone que: $\|y\|_\infty = \max_k |y_k| = 1$.

Cálculo de $\|M_n\|$. Como

$$|M_n(y)| \leq \sum_{k=1}^{n} |y_k| \, |\ell_k(x)| \leq \|y\|_\infty \sum_{k=1}^{n} |\ell_k(x)|,$$

entonces

$$\|M_n\|_\infty \leq \|\lambda_n(x)\| = \max_{a \leq x \leq b} |\lambda_n(x)|. \tag{3.9}$$

Donde $\lambda_n(x) = \sum_{k=1}^{n} |\ell_i(x)|$ es la función de Lebesgue. deBoor en [del] da un procedimiento para demostrar la igualdad en (3.9).

De la relación (3.9) y del hecho que $\|M_n^{-1}\|_\infty = 1$, el número de condición, en la norma uniforme, para las bases de Lagrange resulta:

$$K_\infty(M_n) = \|\lambda_n\|_\infty$$

Ejemplos: A continuación se proporcionan cotas de $\|\lambda_n\|_\infty$, para diferentes distribuciones de nodos.

1) **Interpolación uniformemente espaciada:** deBoor en [del] proporciona una cota inferior del número de condición para las base de Lagrange, cuando los nodos son elegidos uniformemente,

$$\|\lambda_n\| \geq k * e^{n/2},$$

donde k es una constante. Un problema donde la cota crece al aumentar n, es la interpolación de la función de Runge. En 1901, Runge descubrió las dificultades de la interpolación. El trató de interpolar la función

$$f(x) = \frac{1}{1 + 25x^2},$$

en el intervalo [-1, 1], con nodos igualmente espaciados, y descubrió que el polinomio interpolante $p_n(x)$ tiende a infinito al aumentar el grado del polinomio; $p_n(x)$ diverge en el intervalo $0.726 \leq |x| < 1$.

2) **Cuando los $x_i = 1, 2, \ldots n$ son los ceros del polinomio de Chebyshev de grado n [del].**

$$\|\lambda_n\| \leq (2 / \pi) \log(n) + 4$$

Si los nodos de interpolación de la función de Runge son elegidos cercanos a las raíces del polinomio de Chebyshev, el problema con la función desaparece. El polinomio resultante $p_n(x)$ converge a f(x) para x en [-1, 1], como $n \to \infty$, [Fo2].

3) **Si los $x_i = 1, 2, \ldots, n$ son nodos arbitrarios (distintos)** [Ga9], entonces:

$$\|\lambda_n\| > \ln(n) / 8\pi^{1/2}$$

4) **Cuando los $x_i = 1, 2, \ldots,$ son los puntos de Chebyshev expandidos** [del]:

$$x_i = (a + b - (a - b) \frac{\cos((2i - 1)\pi/(2n))}{\cos(\pi/(2n))})/ 2$$

entonces

$$(2/\pi)\ln(n) + 0.5 \leq \|\lambda_n\| \leq (2/\pi)\ln(n) + 0.73$$

3.3 Generación de polinomios ortogonales.

El problema a tratar en esta sección, es la construcción de polinomios ortogonales a partir de los momentos de potencias. El problema ha recibido

poca atención en la literatura, a pesar de que los polinomios ortogonales estan relacionados estrechamente con el análisis aplicado (integración numérica, mínimos cuadrados, expansión de series, fracciones continuas, etc). En opinión de Gautschi, [Gai1], ésto se debe a dos razones: primera, la mayoría del trabajo práctico, donde se emplean polinomios ortogonales, se realiza con polinomios ortogonales conocidos y sus aspectos constructivos son bien conocidos. Segunda, aún en el caso de funciones de peso generales, el problema tiene una solución matemática directa; es bien sabido como expresar o como calcular polinomios ortogonales en términos de los momentos.

La sección inicia con la estimación del número de condición del problema de generación de polinomio ortogonales; la estimación se realiza de manera indirecta debido a la dificultad de realizarla directamente. De acuerdo con el análisis de sensibilidad, el problema es mal condicionado.

En la segunda parte de la sección, se presentan dos algoritmos para la generación de polinomios ortogonales; el primero es el método clásico, conocido como el método de momentos; el segundo, con mejores caracteristicas numéricas, es una variante del primer método y fué propuesto por J. C. Wheeler en [Whi].

3.3.1. El número de condición del problema.

Sea w(x) una función dada (función de peso) definida sobre un intervalo finito [a, b] no negativa, integrable con

(3.10) $$\int_a^b w(x)\,dx > 0 \quad y$$

donde los momentos

(3.11) $$\mu_k = \int_a^b x^k w(x)\,dx\,, \quad k = 0, 1, 2, \ldots 2n-1$$

existen. El problema es determinar los coeficientes a_k y b_k, para k = 0, 1, ..., n, que aparecen en la relación recursiva de tres términos, que deben satisfacer los polinomios buscados.

Es conocida la dificulad para conocer directamente el número de condición del problema. Gautschi en [Ga3] propone una forma indirecta para estimarlo. La idea es bastante simple y está apoyada en la desigualdad (1.6). Si un problema P es la composición de los problemas Q y R y el problema P es mal condicionado

entonces al menos alguno de los problemas (Q ó R) es mal condicionado.

El problema P, para este propósito, es la construcción de las fórmulas de cuadratura de Gauss-Christoffel; dada una función de peso w(x), definida sobre un intervalo finito [a, b], considérese la sucesión de reglas de cuadratura

$$\int_a^b f(x)\, w(x)\, dx \approx \sum_{k=1}^{n} \lambda_k^{(n)} f(\xi_k^{(n)}), \quad n = 0, 1, \ldots \tag{3.12}$$

Cada una de las reglas anteriores es llamada una fórmula de cuadratura de Gauss-Christoffel si tiene el máximo grado de exactitud, es decir, si la función f es un polinomio de grado a lo más 2n-1, entonces la relación (3.12) es una igualdad . Christoffel demostró que si w(x) es una función no negativa, integrable, cumple (3.10) y todos sus momentos existen, entonces, tales cuadraturas existen, son únicas y además $\xi_k^{(n)} \in (a, b)$ y $\lambda_k^{(n)} > 0$.

El procedimiento tradicional para obtener las fórmulas de cuadratura, consiste en construir una familia de polinomios ortogonales $\{p_i\}$ asociados con la función de peso w(x). Los $\xi_k^{(n)}$ son los ceros de p_n y los $\lambda_k^{(n)}$ pueden obtenerse a partir de los polinomios ortogonales, empleando alguno de los métodos existentes. De acuerdo con este procedimiento, el problema R es generar los polinomios ortogonales a partir de los momentos de potencias, y el problema Q es la generación de la regla de cuadratura, a partir de los polinomios ortogonales.

Golub y Welsh [Go2] encontraron que una regla de cuadratura puede obtenerse al calcular los valores propios y las primeras componentes de los vectores propios ortonormalizados de la matriz de Jacobi. La cual se obtiene a partir de la relación recursiva de tres términos que satisface la familia de polinomios ortogonales $\{p_i\}$:

$$p_{-1}(x) = 0 \qquad p_0(x) = 1$$

$$p_{k+1}(x) = x\, p_k(x) - a_k p_k - b_k p_{k-1}$$

La relación anterior puede escribirse en la forma matricial siguiente

$$x \begin{bmatrix} p_0(x) \\ p_1(x) \\ \vdots \\ p_{n-1}(x) \end{bmatrix} = \begin{bmatrix} a_0 & -1 & & & \\ b_1 & a_1 & -1 & & \\ & \ddots & \ddots & \ddots & \\ & & & b_{n-1} & a_{n-1} \end{bmatrix} \begin{bmatrix} p_0(x) \\ p_1(x) \\ \vdots \\ p_{n-1}(x) \end{bmatrix} - \begin{bmatrix} 0 \\ 0 \\ \vdots \\ p_n(x) \end{bmatrix}$$

o en forma equivalente:

(3.13) $$x\, p(x) = Tp(x) - p_n(x)\, e_n,$$

donde T es la matriz tridiagonal y $e_n = (0, 0, \ldots, 1)^t$.

De la relación (3.13), se desprende el primer resultado importante: ξ_i es una raíz del polinomio p(x) si y solamente si

$$Tp(\xi_i) = \xi_i p(\xi_i),$$

es decir, si ξ_i es un valor propio de la matriz T.

La matriz T puede ser transformada a una matriz simétrica, por medio de una transformación de similaridad [Go2]. La matriz obtenida es conocida, en la literatura, como la matriz de Jacobi asociada a la familia de polinomios ortogonales $\{p_i\}$, $(J = DTD^{-1})$.

Los pesos λ_i que aparecen en las fórmulas de cuadratura, se obtienen a partir de las primeras entradas de los vectores propios normalizados, q_i, de la matriz de Jacobi. Aplicando la identidad de Christoffel-Darboux, se demuesta en [Go2] que

(3.14) $$\lambda_i = q_{1i}^2\, \mu_0 \quad , i = 1, 2, \ldots, n.$$

Los valores propios ξ_i, vistos como raíces del polinomio ortogonal $p_n(x)$, son bien condicionados, y además el número de condición de la matriz de Jacobi J_n respecto al problema de determinar los vectores propios, es típicamente del orden n^2 para polinomios ortogonales definidos sobre un intervalo finito, [Ga5]. Por lo anterior el problema R es bien condicionado.

La estimación del número de condición del problema P es equivalente a determinar el número de condición del sistema algebraico no lineal, que resulta de pedir la máxima exactitud de la regla de cuadratura, es decir, que la relación (3.12) es una igualdad exacta si f es un polinomio de grado menor o igual a 2n - 1. El sistema resultante es

(3.15) $$\sum_{r=1}^{n} \lambda_r^{(n)} \left(\xi_r^{(n)}\right)^k = \mu_k \,, \quad k = 0, 1, \ldots, 2n-1.$$

Por simplicidad, se omite el subíndice (n). Para ejemplificar la relación anterior, sea n = 2 y $\mu_0 = 2$, $\mu_1 = 0$, $\mu_2 = 2/3$, $\mu_3 = 0$, los momentos correspondientes a la función de peso w(x) = 1 definida en [-1, 1]. La relación (3.15) se transforma en

$$\lambda_1\xi_1^k + \lambda_2\xi_2^k = \mu_k \ , \ k = 0, 1, 2 \text{ y } 3,$$

o en forma equivalente:

$$\lambda_1 + \lambda_2 = 2$$

$$\lambda_1\xi_1 + \lambda_2\xi_2 = 0$$

$$\lambda_1\xi_1^2 + \lambda_2\xi_2^2 = 2/3$$

$$\lambda_1\xi_1^3 + \lambda_2\xi_2^3 = 0.$$

El sistema tiene solución única $\lambda_1 = \lambda_2 = 1$ y $\xi_2 = -\xi_1 = \frac{\sqrt{3}}{3}$ y la regla

$$\int_a^b f(x)\, w(x)\, dx \approx f(-\frac{\sqrt{3}}{3}) + f(\frac{\sqrt{3}}{3})$$

tiene una precisión semejante a la regla de Simpson, sólo que con dos puntos únicamente, y es la bien conocida fórmula de Gauss.

El sistema (3.15) puede escribirse en forma compacta como

(3.16) $$F(y) = x$$

donde $x^t = (\mu_0, \mu_1, \ldots, \mu_{2n-1})$, y $= (\lambda_1, \lambda_2, \ldots, \lambda_n, \xi_1, \ldots, \xi_n)$, $F^t = (F_1, F_2, \ldots, F_{2n})$ y

(3.17) $$F_k(y) = \sum_{i=1}^{n} \lambda_i \xi_i^{k-1}, \quad k = 1, 2, \ldots, 2n.$$

De acuerdo al capítulo 1, el número de condición de una función $F:\mathbb{R}^n \to \mathbb{R}^m$ derivable está dado por

(3.18) $$K_n(F) = \frac{\|x_0\|}{\|y_0\|} \|[F_y(y_0)]^{-1}\|,$$

donde y_0 es la solución de $F(y) = x_0$ y $F_y(y)$ denota el Jacobiano de la función F. La norma de matrices empleada en la relación (3.18) es una norma subordinada a la norma vectorial elegida para x (y). El Jacobiano de la función F puede expresarse como el producto de la matriz U_{2n} y la matriz diagonal D, donde la matriz U_{2n} es la matriz de Vandermonde confluente de orden 2n

$$U_{2n}(\xi_1, \xi_2, \ldots, \xi_2) = \begin{bmatrix} 1 & 1 & \ldots & 1 & 0 & \ldots & 0 \\ \xi_1 & \xi_2 & \ldots & \xi_n & 1 & \ldots & 1 \\ \xi_1^2 & \xi_2^2 & \ldots & \xi_n^2 & 2\xi_1 & \ldots & 2\xi_n \\ \vdots & \vdots & & \vdots & \vdots & \ldots & \vdots \\ \xi_1^{2n-1} & \xi_2^{2n-1} & \ldots & \xi_n^{2n-1} & (2n-1)\xi_1^{2n-2} & & (2n-1)\xi_n^{2n-2} \end{bmatrix}$$

y D es la matriz

$$D = \begin{bmatrix} 1 & & & & & & \\ & 1 & & & & & \\ & & \ddots & & & & \\ & & & 1 & & & \\ & & & & \lambda_1 & & \\ & & & & & \lambda_2 & \\ & & & & & & \ddots \\ & & & & & & & \lambda_n \end{bmatrix}$$

De la relación (3.18) se tiene

$$K_n(F) = \frac{\|x_0\|}{\|y_0\|} \|D^{-1}U_{2n}^{-1}\|. \tag{3.19}$$

Gautschi en [Ga1], [Ga2] y [Ga8] presenta algunas de las propiedades de la matriz U_{2n}, en particular, proporciona cotas para la norma infinito de la inversa de U_{2n} (Ver apéndice A). Además si se restringe el análisis al intervalo (0, 1) y $w(x) \geq 0$, puede demostrarse, [Ga3], que

$$K_n(F) > \min(\mu_0, 1/\mu_0) \max_{1 \leq r \leq n} \left\{ (1 + \xi_r) \prod_{k \neq r} \left(\frac{1 + \xi_k}{\xi_r - \xi_k} \right)^2 \right\}. \tag{3.20}$$

La segunda parte de la relación (3.20) corresponde a la norma de la inversa de la matriz U_{2n} (Ver. Apéndice A).

La cota inferior para $K_n(F)$ puede resular muy grande y por lo tanto, el problema de determinar los números de Christoffel $(\lambda_1, \lambda_2, \ldots, \lambda_n, \xi_1, \ldots, \xi_n)$ a partir de los momentos de potencias, puede ser mal condicionado.

Resumiendo. La construcción de las fórmulas de cuadratura de Gauss-Christoffel puede verse como la composición de los problemas Q y R, (P = Q.R). El problema R es la generación de polinomios ortogonales y el problema

Q es la determinación de los valores y vectores propios de la matriz de Jacobi.

Como el problema P es mal condicionado, entonces alguno de los problemas que forman la composición también lo es. Como el problema Q es bien condicionado, la única posibilidad es que el problema R -la generación de polinomios ortogonales- sea mal condicionado.

En las dos secciones siguientes se presentan dos métodos para generar polinomios ortogonales a partir de los momentos de potencias. El primero es el método clásico, conocido como el método de momentos, el cual, desde el punto de vista numérico, puede presentar crecimiento exponencial del error de redondeo.

El segundo procedimiento tiene mejores características numéricas y emplea una familia de polinomios ortogonales, conocida de antemano, y los momentos de potencias iniciales, para obtener los momentos modificados. Los polinomios ortogonales se generan a partir de estos nuevos momentos.

3.3.2 El método de momentos.

El método parte del conjunto de momentos de potencia: $\mu_0, \mu_1, \ldots, \mu_{2n-1}$ y obtiene los coeficientes a_i, b_i, $i = 0, 1, 2, \ldots, n-1$. de la relación recursiva de tres términos que deben satisfacer los polinomios buscados. De acuerdo a (A.8), la relación de recurrencia puede reescribirse como:

$$f_{k+1}(x) = (x - a_k)f_k(x) - b_k f_{k-1}(x), \qquad f_{-1} = 0,\ f_0 = 1 \tag{3.21}$$

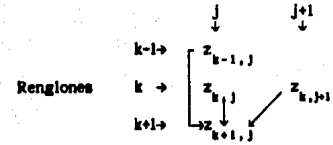
El presente algoritmo es una ligera variación del método atibuido a Chebyshev.

Sean $f_0, f_1, \ldots, f_{2n-1}$ los polinomios ortogonales a determinar, con $f_0 = 0$ y $f_1 = 1$. Sea Z la matriz con elementos $z_{i,j} = \langle f_i, x^j \rangle = \int_a^b f_i(x)x^j w(x)dx$, para $i = -1, 0, 1, \ldots n-1$, y $j = 0, 1, 2, \ldots, 2n-1$; claramente el renglón -1 de esta matriz, tiene sólo entradas 0 y el renglón 0, consiste de los momentos $\mu_0, \mu_1, \ldots \mu_{2n-1}$, además $z_{i,j} = 0$ para $i > j$.

De la definición de producto interno y de (3.21), se obtiene que los elementos de la matriz Z satisfacen la relación siguiente:

$$z_{k+1,\,j} = z_{k,j+1} - a_k z_{k,j} - b_k z_{k-1,j} \tag{3.22}$$

Esta relación es fácil de recordar por medio del diagrama siguiente:



Después de haber calculado los primeros k renglones, los coeficientes a_k y b_k son determinados a partir de las relaciones

(3.23) $$z_{k+1,k-1} = 0 \Rightarrow b_k = z_{k,k} / z_{k-1,k-1} \quad (b_0 = \mu_0)$$

(3.24) $$z_{k-1,k} = 0 \Rightarrow a_k = z_{k,k+1} / z_{k,k} - z_{k-1,k} / z_{k-1,k-1}.$$

Ejemplo: Sean $\mu_0 = 2$, $\mu_1 = 0$, $\mu_2 = 2/3$, $\mu_3 = 0$ los momentos correspondientes a la función de peso $w(x) = 1$ en el intervalo $[-1, 1]$. La matriz Z inicial resulta:

(-1)	0	0	0	0
(0)	μ_0	μ_1	μ_2	μ_3

De acuerdo a las relaciones (3.23) y (3.24), los valores de los coeficientes $b_0 = 2$, $a_0 = \mu_1 / \mu_0 = 0$. El paso siguiente es aplicar la relación de recurrencia (3.22) cuando $k = 0$. La matriz Z resultante es:

(-1)	0	0	0	0
(0)	μ_0	μ_1	μ_2	μ_3
(1)		$\mu_2 - a_0\mu_1$	$\mu_3 - a_0\mu_2$	0

,

y los valores de $b_1 = \mu_2 / \mu_0 = 1/3$, $a_1 = \mu_3 / \mu_2 = 0$.

Los polinomios ortogonales resultantes son:

$$f_{-1}(x) = 0, \qquad f_0(x) = 1,$$

$$f_1(x) = x\, f_0(x) - 2 f_{-1}(x) = x$$

$$f_2(x) = x\, f_1(x) - 1/3\, f_0(x) = x^2 - 1/3$$

En la programación de este método, es suficiente con conservar en memoria los dos últimos renglones calculados de la matriz Z. El código del programa aparece en el Apendice (B.2).

Wheeler [Whi] empleó este método para construir reglas de cuadratura, las cuales fueron empleadas en la estimación de promedios. Cuando el número de

momentos fué pequeño, los resultados fueron adecuados. Las dificultades se presentaron cuando el número de momentos fué grande, la precisión de la computadora fué limitada, o bien, la función tenía cambios bruscos.

3.3.3 El método de momentos modificados.

El mal condicionamiento de la transformación de los momentos de potencias a los coeficientes que aparecen en la relación recursiva de tres términos, se debe a que la función x^n "recupera" a w(x) solamente cerca de 1, para n grande. Una mejor "codificación" de w(x) fué propuesta por Sack y Donovan [Sa2] e involucra los momentos *modificados*:

$$\nu_k = \int_a^b p_k(x)\, w(x)\, dx\ ,\quad k = 0,\ 1,\ \ldots$$

donde (p_i) es una familia apropiada de polinomios ortogonales, regularmente, ortogonales con respecto a una medida clásica. La primera parte del algoritmo, consiste en obtener los momentos *modificados*, a partir de los momentos de potencias iniciales. Sean $\mu_0,\ \mu_1,\ \ldots,\ \mu_{2n-1}$ lo momentos de potencias iniciales, $p_0,\ p_1,\ \ldots,\ p_{2n-1}$ la familia de polinomios ortogonales adecuada e Y la matriz con elementos $y_{k,j} = \langle p_k, x^j \rangle$. El primer renglón esta formado por los momentos de potencias y el renglón -1 por ceros. Los elementos de Y satisfacen una relación similar a (3.22):

$$y_{-1,1} = 0, \quad y_{0,1} = \mu_1$$

$$y_{k+1,j} = y_{k,j+1} - \alpha_k y_{k,j} - \beta_k y_{k-1,j}, \tag{3.25}$$

donde α_k y β_k corresponden a los coeficientes de la relación de recurrrencia, que satisfacen los polinomios (p_i). El primer paso del método, es obtener los momentos modificados; éstos se obtienen de la aplicación reiterada de la relación (3.25). Los momentos modificados van quedando en la primera columna de Y.

Por ejemplo, sean $\mu_0,\ \mu_1,\ \mu_2$ y μ_3 como en el ejemplo anterior y (T_i^-) la familia de polinomios de Chebyshev normalizados. La relación de recurrencia de esta familia es

$$T_{-1}^-(x) = 0, \quad T_0^-(x) = 1$$

$$T^{\sim}_{k+1}(x) = x\,T^{\sim}_{k}(x) - 1/2T^{\sim}_{k-1}(x).$$

La matriz Y resultante es

(-1)	0	0	0	0
(0)	2	0	2/3	0
(1)	0	2/3	0	
(2)	-1/3	0		
(3)	0.			

El segundo paso del método consiste en aplicar un procedimiento similar al de Chebyshev (método de *momentos*), a los momentos modificados ν_0, ν_1, ..., ν_{2n-1} para obtener los coeficientes de la relación de recurrencia de los polinomios ortogonales (q_i), i = 1,2, ..., 2n-1, asociados a la función de peso w(x). Los coeficientes a determinar α_i y β_i, i = 0, 1, ... n-1, deben cumplir la relación:

$$q_{-1}(x) = 0, \quad q_0(x) = 1$$

$$q_{k+1}(x) = xq_k(x) - \alpha_k q_k(x) - \beta_k\, q_{k-1}(x) \tag{3.26}$$

En esta etapa se construye la matriz W, con entradas $w_{k,l} = \langle q_k, p_m \rangle$, k = -1, 0, 1, ..., 2n - 1 y m = -1, 0, ..., n. Esta matriz tiene en su primer renglón y columna (-1) igual a cero, el renglón 1 contiene los momentos modificados ν_1, ν_2, ..., ν_{2n-1}, la columna 1 y además

$$w_{km} = 0 \text{ para } m < k. \tag{3.27}$$

De las relaciones (3.20) y (3.26), puede demostrarse la relación siguiente:

$$w_{k+1,m} = w_{k,m+1} - (\alpha_k - a_k)w_{k,m} - \beta_k w_{k-1,m} + b_k w_{k,m-1}$$

$$k = 0, 1, \ldots, n-2,$$

$$m = k, k+1, \ldots, 2n-k-1 \tag{3.28}$$

Después de calcular los k primeros renglones de W, pueden obtenerse relaciones para determinar α_k y β_k. De la propiedad (3.27) y la relación (3.28) se obtiene:

$$w_{k+1,k-1} = 0 \;\Rightarrow\; \beta_k = w_{k,k} \,/\, z_{k-1,k-1}$$

$$w_{k+1,k} = 0 \quad \Rightarrow \quad \alpha_k = a_k + w_{k,k+1} / w_{k,k} - w_{k-1,k-1} / w_{k-1,k-1}$$

Para la programación del método se emplearon, al igual que el método de Chebyshev, dos arrregios de longitud 2n. El primer arreglo es inicializado a cero y corresponde al rengion -1 de la matriz W. El segundo arreglo contiene los momentos modificados. El listado del programa se encuentra en el apéndice (B.4)

CAPITULO 4

ESTIMADORES DEL NUMERO DE CONDICION DE MATRICES

4.1 Matrices triangulares
4.2 Matrices tridiagonales

Los códigos de programación relacionados con la solución del problema algebraico Ax = b, contemplan la estimación del número de condición del problema de inversión, $K(A) = \|A\|\|A^{-1}\|$. Calcular el número de condición es una operación costosa, debido a que es necesario contar con la inversa de A, y en general esta operación es más compleja que resolver el problema original. Esto hace interesante e importante la búsqueda de métodos para la estimación de este número.

La mayoría de los métodos de estimación para matrices densas y pequeñas parten del hecho que la matriz ya se encuentra factorizada y emplean los factores para estimar K(A). La primera sección está dedicada a los métodos de estimación de matrices triangulares. En la segunda se presentan algunos métodos de estimación para matrices tridiagonales.

4.1 Matrices Triangulares

En varias áreas de aplicación, la matriz dada ya se encuentra en una forma triangular o bien ha sido factorizada de acuerdo a alguna descomposición, la cual tiene un factor triangular. La relación entre el número de condición de la matriz original y los números de condición de los factores depende de la descomposición empleada.

En la práctica, un sistema denso de ecuaciones lineales es resuelto ordinariamente por eliminación Gaussiana con algún tipo de pivoteo. Esto proporciona matrices de permutación P y Q, una matriz triangular inferior L y una matriz triangular superior U tal que:

$$PAQ = LU,$$

donde las matrices P y Q son los intercambios de renglones y columnas necesarios para realizar el pivoteo. En el pivoteo parcial la matriz Q es la identidad. Con pivoteo parcial o total los elementos de L satisfacen:

$$|\ell_{ij}| \leq 1.$$

El pivoteo asegura, generalmente, el buen condicionamiento de la matriz L, y el mal condicionamiento de la matriz A es reflejado en el correspondiente mal condicionamiento de la matriz U. Cuando A es singular algún elemento u_{ii}, comunmente u_{nn}, es cero. Así el mal condicionamiento de A, se debe, con frecuencia, a un valor de u_{nn} pequeño. Por supuesto que A puede ser mal condicionada sin que ningún u_{ii} sea pequeño. Un ejemplo de este hecho es el sistema (2.5)

Un caso más simple es cuando tenemos una factorización de la forma:

$$A = QR,$$

donde Q es ortogonal y R es una matriz triangular superior. En este caso:

$$\|A\|_2 = \|R\|_2, \quad \|A^{-1}\| = \|R^{-1}\|_2, \quad K_2(A) = K_2(R),$$

donde K_2 denota el número de condición en la norma espectral. En la descomposición de Cholesky (con pivoteo) de una matriz Hermitiana positiva definida $A \in \mathbb{C}^{n,n}$, se tiene:

$$P^T AP = LL^*,$$

donde L es una matriz triangular inferior con elementos positivos y reales en la diagonal. Empleando propiedades de la norma espectral y de la de Frobenious [St1], puede demostrarse que:

$$K_2(A) = K_2(L)^2.$$

Los métodos para obtener una estimación del número de condición de matrices triangulares, podemos dividirlo en tres grupos; el primer grupo, comprende los métodos obtenidos a partir de las desigualdades de matrices que dependen solamente de la magnitud de los elementos de la matriz triangular. El segundo grupo está formado por los algoritmos heuristicos o probabilisticos, motivados por la definición de norma subordinada. Y el último grupo lo integra solamente el método vía optimización convexa.

Los resultados incluidos en esta sección están relacionados con las normas 1, 2, ∞ y la de Frobenius. Por facilidad de cálculo, la mayoría de los

códigos existentes estiman el número de condición para las normas $\|\ \|_1$ y $\|\ \|_\infty$; aunque también son importantes los métodos de estimación para la norma espectral, debido a la relación de esta norma con los valores propios de la matriz A^tA.

4.1.1 Matrices de comparación.

Sea $T = (t_{ij}) \in \mathbb{R}^{n\times n}$ una matriz triangular de orden n sobre el campo de los reales. Una cota inferior para la norma de la inversa de T es:

$$(\min\ |t_{ii}|)^{-1} \le \|T^{-1}\|_{1,2,\infty,F}.$$

La validez de esta desigualdad está basada en que el reciproco de los elementos de la diagonal de la matriz T son elementos de la diagonal de su inversa. Para obtener cotas superiores de $\|T^{-1}\|$ pueden emplearse las matrices de comparación siguientes:

I) $M(T) = (m_{ij})$

$$m_{ij} = \begin{cases} |t_{ii}| & \text{si } i = j \\ -|t_{ij}| & \text{si } i \neq j\ . \end{cases}$$

Estas matrices fueron introducidas por Ostrowski [Os1] en 1937, en el estudio de la convergencia de procesos iterativos en matrices. La matrices M, son útiles, también, en el análisis espectral de ciertas matrices. Una caracterización de estas matrices aparece en [Po1].

II) $W(T) = (w_{ij})$

$$w_{ij} = \begin{cases} |t_{ii}| & \text{si } i = j \\ -\alpha_i & \text{si } i \neq j \end{cases}$$

$$\alpha_i = \max_{i\neq j} |t_{ij}|.$$

III) Si A es una matriz con elementos en la diagonal diferentes de cero, la matriz $Z(A) = (z_{ij})$, está definida como:

$$z_{ij} = \begin{cases} \beta & \text{si } i = j \\ -\alpha\beta & \text{si } i \neq j \end{cases}$$

$$\alpha = \max_{i<j} \frac{|t_{ij}|}{t_{ii}} \qquad \beta = \min_i |t_{ii}|.$$

Es fácil demostrar que si T es una matriz triangular no singular, entonces las matrices M(T), W(T) y Z(T) tienen inversa y todos sus elementos son no negativos. En general, este resultado no se cumple, por ejemplo, la matriz

$$A = \begin{bmatrix} 1 & -1 \\ 1 & 1 \end{bmatrix}$$

es no singular, sin embargo $\det(M(A)) = \det(W(A)) = \det(Z(A)) = 0$. Y aunque la inversa exista, no necesariamente sus elementos son no negativos. Por ejemplo, la inversa de las matrices $M(H_3)$, $W(H_3)$ y $Z(H_3)$, donde H_3 es la matriz de Hilbert de orden 3, tienen elementos negativos

$$M^{-1}(H_3) = -\frac{1}{359}\begin{bmatrix} 9 & 396 & 510 \\ 396 & 192 & 900 \\ 510 & 900 & 180 \end{bmatrix},$$

$$W^{-1}(H_3) = -\frac{1}{67}\begin{bmatrix} 18 & -48 & -57 \\ -48 & -6 & -135 \\ -50 & -90 & -15 \end{bmatrix},$$

$$Z^{-1}(H_3) = -\frac{1}{56}\begin{bmatrix} 30 & -75 & -75 \\ -75 & 30 & -75 \\ -75 & -75 & 30 \end{bmatrix}.$$

El resultado principal de esta sección es el lema siguiente:

Lema 4.1, [Hil]: Sea T una matriz triangular no singular, entonces

$$\|T^{-1}\|_p \leq \|M(T)^{-1}\|_p \leq \|W(T)^{-1}\|_p \leq \|Z(T)^{-1}\|_p, \quad p = 1, 2, \infty, F.$$

Este resultado puede emplearse directamente para estimar $K_p(T)$, con $p = 1, 2, \infty$ y F.

4.1.2. Normas de las inversas de las matrices de comparación.

Para las matrices M(T), W(T) y Z(T), existen procedimientos para calcular la normas 1, 2, ∞ y F de sus respectivas inversas, sin calcular la inversa en forma explícita.

Caso I. La norma 1 e infinito.

Higham en [Hil], presenta los algoritmos 4.1 y 4.2 para calcular la norma infinita de la inversa de M(T) y W(T) respectivamente. El cálculo de estas normas parte de la observación siguiente: si A es una matriz con elementos no negativos, entonces $\|A\|_\infty = \|Ae\|_\infty$, con $e = (1, 1, \ldots, 1)^t$. Por ejemplo, para la matriz M(T), es suficiente calcular $\|M(T)^{-1}e\|_\infty$.

Algoritmo 4.1. [Hil]. Calcula $\|M^{-1}(T)\|_\infty$ para T una matriz triangular superior de orden nxn. La variable r_M contiene el resultado.

```
z_n := 1/ |t_nn|
For i:= n - 1 Downto 1 Do
Begin
  s := 1 ;
  s := s + |t_ij| * z_j    (j := i+1,... n)
  z_i:= s / |t_ii|
End ;
r_M := ||z||_∞
```

Algoritmo 4.2.[Hil]. Calcula $\|W^{-1}(T)\|_\infty$ para T una matriz triangular superior de orden nxn. El resultado queda en la variable r_W.

```
z_n := 1 / |t_nn|
s := 0
For i := n - 1 Downto 1
Begin
  s := s + z_i+1
  α_i := 1 + max_i≤j≤n |t_ij|
  z_i := ( 1 + α_i * s) / |t_ii|
```

End ;

$r_w := \|z\|_\infty$

Lemiere demuestra, en [Lei], que la norma infinita de la inversa de la matriz Z(T) satisface la relación:

$$\|Z(T)^{-1}\|_\infty = \frac{(\alpha + 1)^{n-1}}{\beta}. \qquad (4.1)$$

Empleando los algoritmos anteriores y del hecho que $\|A\|_1 = \|A^t\|_\infty$, pueden obtenerse estimadores para $\|M(T)^{-1}\|_1$, $\|W(T)^{-1}\|_1$ y $\|Z(T)^{-1}\|_1$.

Ejemplo. Sea T la matriz triangular

$$T = \begin{bmatrix} 1 & -1 & -2k & 0 \\ 0 & 1 & k & -k \\ 0 & 0 & 1 & -1 \\ 0 & 0 & 0 & k \end{bmatrix} \quad \text{con } k \geq 1.$$

La norma de la inversa de T es igual a $\|T^{-1}\|_\infty = k + 4$. Aplicando el algoritmo 4.1 a

$$M(T) = \begin{bmatrix} 1 & -1 & -2k & 0 \\ 0 & 1 & -k & -k \\ 0 & 0 & 1 & -1 \\ 0 & 0 & 0 & k \end{bmatrix},$$

se obtiene $\|M(T)^{-1}\|_\infty = 3k + 6$. El estimador es tres veces mayor que el de la matriz original. Ahora aplicando el método 4.2 a la matriz

$$W(T) = \begin{bmatrix} 1 & -2k & -2k & -2k \\ 0 & 1 & -k & -k \\ 0 & 0 & 1 & -1 \\ 0 & 0 & 0 & k \end{bmatrix},$$

se obtiene $\|W(T)^{-1}\|_\infty = 2k^2 + 8k + 5$. Finalmente al aplicar la relación (4.1) a la matriz

$$Z(T) = \begin{bmatrix} 1 & -2k & -2k & -2k \\ 0 & 1 & -2k & -2k \\ 0 & & 1 & -2k \\ 0 & & & 1 \end{bmatrix},$$

se obtiene $\|Z(T)^{-1}\| = (2k + 2)^3$.

End ;

$r_w := \|z\|_\infty$

Lemiere demuestra, en [Lel], que la norma infinita de la inversa de la matriz Z(T) satisface la relación:

$$\|Z(T)^{-1}\|_\infty = \frac{(\alpha + 1)^{n-1}}{\beta}. \tag{4.1}$$

Empleando los algoritmos anteriores y del hecho que $\|A\|_1 = \|A^t\|_\infty$, pueden obtenerse estimadores para $\|M(T)^{-1}\|_1$, $\|W(T)^{-1}\|_1$ y $\|Z(T)^{-1}\|_1$.

Ejemplo. Sea T la matriz triangular

$$T = \begin{bmatrix} 1 & -1 & -2k & 0 \\ 0 & 1 & k & -k \\ 0 & 0 & 1 & -1 \\ 0 & 0 & 0 & k \end{bmatrix} \quad \text{con } k \geq 1.$$

La norma de la inversa de T es igual a $\|T^{-1}\|_\infty = k + 4$. Aplicando el algoritmo 4.1 a

$$M(T) = \begin{bmatrix} 1 & -1 & -2k & 0 \\ 0 & 1 & -k & -k \\ 0 & 0 & 1 & -1 \\ 0 & 0 & 0 & k \end{bmatrix},$$

se obtiene $\|M(T)^{-1}\|_\infty = 3k + 6$. El estimador es tres veces mayor que el de la matriz original. Ahora aplicando el método 4.2 a la matriz

$$W(T) = \begin{bmatrix} 1 & -2k & -2k & -2k \\ 0 & 1 & -k & -k \\ 0 & 0 & 1 & -1 \\ 0 & 0 & 0 & k \end{bmatrix},$$

se obtiene $\|W(T)^{-1}\|_\infty = 2k^2 + 8k + 5$. Finalmente al aplicar la relación (4.1) a la matriz

$$Z(T) = \begin{bmatrix} 1 & -2k & -2k & -2k \\ 0 & 1 & -2k & -2k \\ 0 & & 1 & -2k \\ 0 & & & 1 \end{bmatrix},$$

se obtiene $\|Z(T)^{-1}\| = (2k + 2)^3$.

Caso 2. Norma de Frobenius.

Karasalo en [Kal] da un método para calcular $\|W^{-1}(T)\|_F$. El procedimiento queda establecido en el lema siguiente:

Lema 4.2. Sea T una matriz triangular y consideremos la matriz de comparación

$$\begin{bmatrix} t_{11} & -t_1 & -t_1 & \cdots & -t_1 \\ & t_{22} & -t_2 & \cdots & -t_2 \\ & & & & \\ & & & & t_{nn} \end{bmatrix},$$

entonces

$$\|W(T)^{-1}\|_F = \sum_{i=1}^{n} \mu_i / t_{ii}^2,$$

donde μ_i está determinada por la relación de recurrencia:

$$\begin{cases} \mu_1 = 1 \\ \mu_i = (1 + C_{i-1})^2 \mu_{i-1} - 2C_{i-1}, \quad i = 2,3,..n \end{cases}$$

donde $C_i = t_i / t_{ii} \quad i = 1, 2, \ldots n - 1.$

Caso 3. Norma espectral.

No se conocen algoritmos para estimar la norma espectral de las inversas de las matrices de comparación. Una forma de acotar su valor es emplear los algoritmos 4.1, 4.2 o la relación (4.1) y emplear alguna desigualdad de normas, por ejemplo, la relación (2.12) o (2.15).

Ejemplo. Emplear la relación (2.15) para estimar $\|M(T)^{-1}\|_2$, $\|W(T)^{-1}\|_2$ y $\|Z(T)^{-1}\|_2$ con T igual a la del ejemplo anterior.

como $RM = (\|M(T)^{-1}\|_1 \|M(T)^{-1}\|_\infty)^{1/2} = (12k^2 + 27k + 6)^{1/2}$, entonces

$$\frac{1}{2} RM \leq \|M(T)^{-1}\|_2 \leq RM.$$

como $RW = (\|W(T)^{-1}\|_1 \|W(T)^{-1}\|_\infty)^{1/2} = (24k^2 + 10k + 1)^{1/2}$, entonces

$$\frac{1}{2} RW \leq \|W(T)^{-1}\|_2 \leq RW.$$

y finalmente, como $RZ = (\|Z(T)^{-1}\|_1 \|Z(T)^{-1}\|_\infty)^{1/2} = (2k + 2)^3$, entonces

$$\frac{1}{2} RZ \leq \|Z(T)^{-1}\|_2 \leq RZ.$$

4.1.3 Estimadores Heuristicos.

El grupo de estimadores heuristiticos del número de condición de una matriz triangular es amplio, y sólo se mencionan en esta sección aquellos que son empleados usualmente en los resolvedores de sistemas de ecuaciones lineales, o bien presentan caracteristicas favorables, que los hacen buenos candidatos para substituir a los ya existentes.

Estos métodos parten del hecho de que si A es una matriz no singular, entonces

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

es una cota inferior de la norma de la matriz inversa de A, para b elegida de cierta manera. Por supuesto que b puede elegirse en forma arbitraria y existen buenas posibilidades de obtener un buen estimador, sin embargo, puede mejorarse el estimador al hacer una elección adecuada de b.

Los métodos heuristicos emplean dos pasos para realizar la estimación.

Primer paso. Elegir un vector d de manera que la solución del sistema:

$$A^t y = d,$$

tenga norma grande.

Segundo paso. Resolver el sistema:

$$A x = y,$$

y tomar el cociente $\|y\| / \|x\|$ como un subestimador de $\|A^{-1}\|$.

La diferencia entre los métodos radica en la estrategia empleada para elegir el vector d.

La justificación de la heuristica está basada en la decomposición

en valores singulares[1] (DVS) de la matriz A. Sea $A \in \mathbb{R}^{n\times n}$ una matriz de orden n no singular, entonces existen las matrices ortogonales $U = [u_1, u_2, \ldots u_n]$, $V = [v_1, v_2, \ldots v_n]$ tal que:

$$U^t A V = \Sigma,$$

donde $\Sigma = \mathrm{diag}[\sigma_1, \sigma_2, \ldots, \sigma_n]$ es una matriz diagonal n x n y los elementos de la diagonal son los valores singulares de la matriz A con $\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_n > 0$. Los valores singulares de A son las raíces cuadradas positivas de los valores propios de la matriz simétrica definida positiva AA^t. Además

(4.2) $$\|A\|_2 = \sigma_1.$$

(4.3) $$\|A^{-1}\|_2 = 1/\sigma_n \text{ , de esta manera } K_2(A) = \sigma_1/\sigma_n.$$

(4.4) $$Av_i = \sigma_i u_i, \quad A^t u_i = \sigma_i v_i.$$

(4.5) $$\|A\|_F^2 = \sigma_1^2 + \sigma_2^2 + \ldots + \sigma_n^2.$$

Una formulación equivalente es considerar a la matriz A como una transformación lineal de $\mathbb{R}^n \longrightarrow \mathbb{R}^n$, entonces existen bases ortonormales $\{v_1, v_2, \ldots, v_n\}$ y $\{u_1, u_2, \ldots, u_n\}$ en $\mathbb{R}^n$ tal que $Av_i = \sigma_i u_i$, $i = 1, 2, \ldots, n$. Desde el punto de vista geométrico ésto quiere decir que A manda el sistema de coordenadas $\{v_1, v_2, \ldots, v_n\}$ al sistema $\{u_1, u_2, \ldots, u_n\}$ con contracciones y expansiones de magnitud σ_i, a lo largo de las coordenadas correspondientes. Una demostración geométrica de este resultado aparece en [Bli] y en [Goi] una demostración formal.

En la primera etapa de los métodos se resuelve el sistema triangular $A^t y = d$, con un lado derecho elegido de manera que $\|y\|_2$ sea grande. Expresando el

vector d en términos de la base ortogonal $\{v_i\}$ se tiene:

(4.6) $$d = \|d\|_2 \Sigma \alpha_i v_i \quad \text{con} \quad (\Sigma \alpha_i^2 = 1)$$

además $$y = \|d\|_2 \Sigma \alpha_i A^{-t} v_i.$$

Aplicando (4.4) a la última relación resulta

[1] Esta descomposición fué establecida por Sylver (1889) para matrices reales y cuadradas. El caso general es atribuido a Eckart y Young (1939)

(4.7) $$y = \|d\|_2 \Sigma(\alpha_i/\sigma_i) * u_i,$$

y de (4.6) y (4.7) se obtiene

(4.8) $$\frac{\|y\|_2}{\|d\|_2} = [\Sigma(\alpha_i/\sigma_i)^2]^{1/2}.$$

El miembro derecho de la igualdad (4.8), depende de los valores singulares de la matriz y de la elección de d. Si d es elegido de manera que tenga una fuerte componente de v_n, entonces la expresión de la derecha resulta un valor cercano a $\|A^{-1}\|_2 = 1/\sigma_n$, más aún cuando la matriz es mál condicionada. Sin embargo, ésto en la practica es difícil de lograr, pues no se conocen los vectores (v_i).

Para analizar el lado izquierdo puede admitirse que $\|d\|_2$ es constante y por lo tanto, el cociente depende exclusivamente de $\|y\|_2$. Dado un cierto valor de $\|y\|_2$, pueden existir varias maneras de combinar los reciprocos de los valores singulares para obtener el valor $\|y\|_2/\|d\|_2$. Cuando la matriz es mál condicionada y además se pide que $\|y\|_2$ sea grande, entonces es probable que el cociente sea una buena aproximación de $K_2(A)$.

Por ejemplo la matriz triangular superior

$$T = \begin{bmatrix} 1 & .98 & -2 \\ 0 & .1 & -.1 \\ 0 & 0 & 1 \end{bmatrix},$$

tiene una DVS igual a $U^tTV = \Sigma$, con

$$U = \begin{bmatrix} .3644 & -.6121 & .7018 \\ .3590 & -.6030 & -.7124 \\ -.8593 & -.5115 & .000 \end{bmatrix}, \quad V = \begin{bmatrix} .9419 & -.3320 & .0501 \\ .0471 & -.0169 & -.9987 \\ -.3324 & -.9431 & .0002 \end{bmatrix}$$

y $\Sigma = Diag[2.5848, .5424, .0713]$.

Si $d_i \in \{-1,1\}$, entonces existen 8 posibles lados derechos para resolver el sistema transpuesto, con $\|d\| = \sqrt{3}$. Para cada una de las alternativas, se calcularon: el vector y correspondiente, los α_i, para $i = 1, 2$ y 3, y el cociente $\|y\|_2 / \|b\|_2$. Los valores de estas cantidades aparecen en la tabla 4.1.

d	α_1	α_2	α_3	$\|y\|_2/\|d\|_2$
$d_1 = [-1,-1,-1]$	-.3791	.7459	.5475	1.84
$d_2 = [-1,-1, 1]$	-.7630	-.3431	.5479	0.83
$d_3 = [-1, 1,-1]$	-.3246	.7264	-.6057	11.46
$d_4 = [-1, 1, 1]$	-.7085	-.3625	-.6053	11.46
$d_5 = [1,-1,-1]$	.7085	.3625	.6053	11.46
$d_6 = [1,-1, 1]$	.3246	-.7264	.6057	11.46
$d_7 = [1, 1,-1]$	.7630	.3431	-.5479	0.83
$d_8 = [1, 1, 1]$	.3791	-.7459	-.5475	1.84

abla 4.1. Estimadores de $\|T^{-1}\|_2$ obtenidos en el primer paso de la heurística cuando $d_1 \in \{1,-1\}$.

Dos observaciones se desprenden de la tabla anterior: primera, el valor de $\|T^{-1}\|_2 = 14.02$ no se alcanza con ninguno de los lados derechos propuestos. Y segundo, elegir un vector d que asegure que $\|y\|_2$ sea grande no implica, en general, que α_3 sea grande. Obsérvese que tampoco es el más pequeño.

En el segundo paso del método se emplea la solución del sistema transpuesto para resolver el sistema $Tx = y$. Como $x = A^{-1}y$, entonces

$$x = \|d\|_2 \Sigma(\alpha_i/\sigma_i) A^{-1} u_i.$$

Aplicando (4.4) se obtiene

(4.9) $$x = \|d\|_2 \Sigma(\alpha_i/\sigma_i^2) v_i,$$

y de (4.7) y (4.9) resulta:

(4.10) $$\frac{\|x\|_2}{\|y\|_2} = \left[\frac{\Sigma(\alpha_i/\sigma_i^2)}{\Sigma(\alpha_i/\sigma_i)^2}\right]^{1/2} \leq \|A^{-1}\|_2.$$

Si d no tiene una componente de v_n que sea exageradamente pequeña, entonces el vector x es dominado completamente por su componente v_n. Con el proceso de dos pasos se tuvo un beneficio de σ_1^{-2}. Resultando un buen estimador para σ_n^{-1}.

En la tabla 4.2 se encuentran los cocientes $\|x\|_2/\|y\|_2$ respecto a la elección hecha en el primer paso (Tabla 4.1).

d	$\|y\|_2/\|d\|_2$	$\|x\|_2/\|y\|_2$
d_1, d_8	1.84	1.95
d_2, d_7	0.83	2.21
d_3, d_4, d_5, d_6	11.46	14.00

Tabla 4.2. Estimador de $\|T^{-1}\|_2$ después del segundo paso de la heuristica.

La heuristica cubierta en esta sección es general y puede aplicarse a cualquier tipo de matriz no singular. En la práctica, los métodos heuristicos emplean el cociente (4.10) como estimador del número de condición de matrices en diferentes normas.

4.1.3.1. DECOMP.

La lista de estimadores se inicia con tres estimadores clásicos propuestos originalmente por Cline, Moler, Stewart y Wilkinson [Cl1] y empleados en las rutinas DECOMP y SOLVE de Forsythe, Malcolm y Moler [Fo2] y en el paquete Linpack [Do1].

En el método DECOMP, la forma de resolver el sistema triangular $T^t y = d$, sugiere el procedimiento a seguir en la primera etapa del método. En el paso k-ésimo de la substitución hacia adelante del sistema:

$$k \rightarrow \begin{bmatrix} t_{11} & & & & \\ t_{12} & t_{22} & & & \\ \vdots & \vdots & \ddots & & \\ t_{1k} & t_{2k} & \cdots & t_{kk} & \\ \vdots & \vdots & & \vdots & \ddots \\ t_{1n} & t_{2n} & & t_{kn} & \cdots t_{nn} \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_k \\ \vdots \\ y_n \end{bmatrix} = \begin{bmatrix} d_1 \\ d_2 \\ \vdots \\ d_k \\ \vdots \\ d_n \end{bmatrix},$$

se tiene la relación

$$t_{kk} y_k = d_k - (t_{1k} y_1 + t_{2k} y_2 + \cdots + t_{k-1,k} y_{k-1}),$$

$$= d_k - P^k(k-1), \qquad (4.11)$$

donde $P^1(k) = t_{11}y_1 + t_{21}y_2 + \dots + t_{k,1}y_k$.

Si se elige $d_k = a \in (-1, 1)$, entonces en cada paso debe resolverse el problema:

$$\max \quad \phi_k(a) = |a - P^k(k-1)|$$
$$\text{sujeta a que} \quad a \in (-1, 1).$$

Para lograr este objetivo, basta elegir el signo de d_k igual al signo de $P^k(k-1)$ o bién positivo si es cero.

Esta estrategia es la más sencilla y esta implantada en los subprogramas DECOMP y SOLVE que aparecen en [Fo2]. Las rutinas originales tienen un error tipográfico en la sección correspondiente a la solución del sistema transpuesto $L^t x = y$. En el apéndice B) se incluye la rutina DECOMP sin error.

Por supuesto que la heuristica puede fallar, entre otras cosas, porque una vez seleccionado el signo de d_k el valor de y_k queda establecido y puede suceder que no sea la mejor elección, desde el punto de vista global. Dicho de otra manera, DECOMP sólo emplea información local para decidir el signo de d_k. Por ejemplo el sistema

$$\begin{bmatrix} 1 & 0 & k- & k \\ 0 & 1 & -k & k \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} d_1 \\ d_2 \\ d_3 \\ d_4 \end{bmatrix},$$

tiene un número de condición igual a $(1 + 2k)^2$. Aplicando el procedimiento de anterior se obtiene

$k = 1)$ DECOMP siempre elige el signo de d_1 positivo, asi que $d_1 = 1$ y $y_1 = 1$.

$k = 2)$ Como $y_2 = d_2$, el signo de d_2 se elige positivo y $y_2 = 1$.

$k = 3)$ Como $y_3 = d_3 - (ky_1 - ky_2) = d_3$. El signo de d_3 se elige positivo y el valor de $y_3 = 1$.

$k = 4)$ Finalmente $y_4 = d_4 - (-ky_1 + ky_2) = d_4$, por lo tanto el signo de d_4 es

también positivo. El vector resultante es:

$$d^t = (1, 1, 1, 1).$$

En el segundo paso del procedimiento, se resuelve el sistema $Tx = y$. El vector solución es $x^t = (1, 1, 1, 1)$ y el estimador de $\|T^{-1}\|_1$, es el cociente $\|x\|_1 / \|y\| = 1$.

El método falla al dar un estimador del número de condición ($2k + 1$) muy por debajo del valor exacto.

4.1.3.2 LINPACK.

Cline en [Cl3] generaliza el estimador anterior y la estrategia propuesta es similar, solo que para decidir el signo de d_k se toma en cuenta también lo que puede ocurrir a futuro en la solución del sistema. En el paso k se tiene la relación siguiente:

$$t_{kk} y_k = d_k - p^k(k-1)$$

y además para cada $r > k$ se han calculado las cantidades

$$p^r(k-1) = t_{1r} y_1 + t_{2r} y_2 + \dots + t_{k-1,r} y_{k-1},$$

que corresponden al producto de las primeras k-1 entradas del renglón r por las variables ya calculadas. d_k es elegida de manera que y_k y las cantidades $p^{k+1}(k-1), \dots, p^n(k-1)$ sean grandes. Si $b_k = a \in (-1, 1)$ entonces

$$\phi_k(a) = |y_k(a)| + \sum_{i=k+1}^{n} |p^i(k-1) + t_{ki} y_k(a)|,$$

con $y_k(a) = (d_k - p^k(k-1))/t_{kk}$, es la función a maximizar en cada paso de la substitución hacia adelante.

Por ejemplo, para la matriz del ejemplo anterior, la función $\phi_k(a)$ puede escribirse como $|y_k(a)| + S_{k+1}(a)$, donde $S_{k+1}(a)$ es la segunda expresión. Los valores de estas expresiones durante el primer paso del método son:

Paso (k)	$y_k(-1)$	$y_k(1)$	$S_{k+1}(-1)$	$S_{k+1}(1)$	b_k	y_k
1	1	1	0	0	1	1
2	-1	1	2k	0	-1	-1
3	-1-2k	1-2k	-2k	-2k	-1	-1-2k
4	-1+2k	1+2k	-	-	1	1+2k

En el segundo paso, se resuelve el sistema Tx = y, y se obtiene un estimador de $\|T^{-1}\|_1$: $\|x\|_1/\|y\| = k + 1$. El estimador del número de condición es igual a (k+1)(2k+1). El resultado obtenido por LINPACK, para este ejemplo, es considerablemente mejor que el obtenido por DECOMP y el costo adicional de esta estrategia es aproximadamente el doble de las operaciones requeridas por DECOMP y un arreglo de tamaño n.

El paquete LINPACK emplea este método para estimar $K_1(A)$, donde A es una matriz general (Subprograma SGECO) o una matriz triangular (Subprograma DTRCO). El algoritmo puede producir sobreflujo durante la división entre t_{kk} o fallar completamente si cualquier t_{kk} es cero. Para aminorar estas dificultades la matriz se mantiene escalada apropiadamente.

4.1.3.3. LINPACKM.

Una generalización del método anterior es ponderar las cantidades $P^1(k)$ por medio de cantidades no negativas, w_i. La función a maximizar en cada paso de la substitución hacia adelante resulta:

$$\phi_k(a) = |y_k(a)| + \sum_{i=k+1}^{n} w_i |p^1(k-1) + t_{ki} y_k(a)|.$$

En LINPACK los pesos son iguales a 1. Otra opción, mencionada en [Cli] es $w_i = 1/t_{ii}$.

4.1.3.4 RETROSPECTIVO.

El estimador retrospectivo, además de tomar en cuenta lo que puede ocurrir en pasos subsiguientes de la solución del sistema $T^t y = d$, puede modificar las entradas del vector d calculadas previamente, es decir, en el paso k-ésimo se estima el valor de d_k y se adecuan los valores previos de d_i,

$d_2, \ldots, d_{k-1}$ para lograr un efecto anticipado sobre $P^{k+1}(k), \ldots, P^n(k)$.

En esta sección se veran dos casos; cuando se emplea la norma espectral y la norma 1.

Caso I. Norma espectral.

En el paso k-ésimo, los valores de $d_1, d_2, \ldots, d_{k-1}$ ya son conocidos y satisfacen la relación

$$\sum_{i=1}^{k-1} d_i^2 = 1, \tag{4.12}$$

además ya se han calculado los valores hacia adelante $P^{k-1}(k-1), \ldots P^n(k-1)$.

Una manera de modificar las entradas $d_1, d_2, \ldots, d_{k-1}$, de manera que las nuevas entradas sigan teniendo la propiedad (4.12), es multiplicarlas por la cantidad s= sen(a), para algún a ∈ [0, 2π], y hacer la entrada d_k igual a c = cos(a). El sistema en el paso k-ésimo es

$$\begin{bmatrix} t_{11} & & & \\ t_{12} & t_{22} & & \\ \cdots & \cdots & & \\ t_{1,k-1} & t_{2,k-1} & \cdots & t_{k-1,k-1} \\ t_{1k} & t_{2k} & \cdots & t_{kk} \end{bmatrix} \begin{bmatrix} y'_1 \\ y'_2 \\ \cdots \\ y'_{k-1} \\ y'_k \end{bmatrix} = \begin{bmatrix} sd_1 \\ sd_2 \\ \cdots \\ sd_k \\ c \end{bmatrix},$$

donde las nuevas variables, y'_i (i = 1,2,.. k-1), difieren de las anteriores en solo una una constante:

$$y'_i = sy_i \qquad (i = 1,2,\ldots,k-1),$$

y la variable

$$y'_k = (c - \sum_{i=1}^{k-1} t_{ik} y'_i) / t_{kk} = (c - sP^k(k-1)) / t_{kk}.$$

Por lo tanto, establecer el valor de la k-ésima entrada del vector d se reduce a determinar el valor de a ∈ [0, 2π] que maximice la función:

$$\phi(a) = \sum_{i=1}^{k} (y'_i)^2 + \sum_{i=k+1}^{n} (P_a^i(k))^2, \tag{4.13}$$

donde $P_a^i(k) = sP^i(k-1) + t_{ki} y'_k$, para i = k+1, ..., n.

Substituyendo en (4.13) las definiciones de $P_a^i(k)$ y de y'_k se obtiene:

$$\phi(a) = s^2 \sum_{i=1}^{k-1} y_i^2 + ((c - sP^k(k-1))/t_{kk})^2 + \sum_{i=k+1}^{n} (sP^i(k-1) + t_{ki}y_k')^2$$

El parámetro a, puede determinarse a partir de la ecuación $\phi'(a) = 0$.

$$\phi'(a) = 2sc\sum_{i=1}^{k-1} y_i^2 + \frac{2}{t_{kk}^2}(c - sP^k(k-1))(-s - cP^k(k-1)) +$$

$$\sum_{i=k+1}^{n} (sP^i(k-1) + \frac{t_{ik}}{t_{kk}}(c - sP^k(k-1)))(cP^i(k-1) + \frac{t_{ik}}{t_{kk}}(-s - cP^k(k-1)))$$

Si $y = (y_1, y_2, \ldots, y_{k-1})^t$ y $P_i = P^i(k-1)$, entonces la última relación puede escribirse como

$$\phi'(a) = scy^ty + \frac{1}{t_{kk}^2}[cs(P_k^2 - 1) + (s^2 - c^2)P_k] +$$

$$\sum_{i=k+1}^{n}[scP_i^2 - \frac{P_i t_{ik}}{t_{kk}}(s^2 - c^2) - \frac{2scP_i t_{ik}P_k}{t_{kk}} +$$

$$\left[\frac{t_{ik}}{t_{kk}}\right](P_k(s^2 - c^2) + scP_k^2 + 1))].$$

Además si $t = (t_{k+1,k}, \ldots, t_{nk})^t$ y $P = (P_{k+1}, \ldots, P_n)^t$ la expresión anterior resulta

$$\phi'(a) = (s^2 - c^2)(P_k(t^tt + 1) - P^tP\, t_{kk}) +$$

$$sc(t_{kk}^2(y^ty + P^tP) + (P_k^2 - 1)(t^tt + 1) - 2P^tPt_{kk}P_k).$$

Finalmente, sea $\beta = t_{kk}^2(y^ty + P^tP) + (P_k^2 - 1)(t^tt + 1) - 2P^tPt_{kk}P_k$, y $\alpha = P_k(t^tt + 1) - P^tPt_{kk}$; la ecuación $\phi'(a) = 0$ se transforma en la ecuación de segundo grado:

(4.14) $$\alpha \tan^2(a) + \beta \tan(a) - \alpha = 0.$$

Si $r = \beta / 2\alpha$, entonces las raíces de esta ecuación están dadas por las expresiones

$$\mu_1 = r + (1 + r^2)^{1/2}, \quad \mu_2 = r - (1 + r^2)^{1/2}.$$

Los dos posibles pares seno-coseno que satisfacen la ecuación (4.14) son

$$s_1 = 1/(1 + \mu_1^2)^{1/2} \quad ; \quad c_1 = s_1\mu_1,$$

y

$$s_2 = 1/(1 + \mu_2^2)^{1/2} \quad ; \quad c_2 = s_2\mu_2.$$

Substituyendo cada uno de los pares anteriores, en la función $\phi(s)$ puede determinarse donde se alcanza el máximo.

Este método puede generalizarse empleando la misma idea que en LINPACKM.

Caso II. Norma 1.

En esta norma los expresiones resultan más simples y faciles de calcular. En el paso k-ésimo ya se han obtenido las entradas d_1, d_2, ..., d_{k-1} con

$$\sum_{i=1}^{k} |d_i| = 1,$$

y los valores hacia adelante $P^i(k-1)$ para i = k, k+1, ..., n. Al igual que en el caso anterior, las primeras k-1 entradas del lado derecho son multiplicadas por una cantidad $\lambda \in (0, 1)$ y la entrada d_k se hace igual a 1-λ. Los valores de las nuevas variables en términos de las anteriores, quedan establecidos por las relaciones:

$$y'_i = \lambda\, y_i \text{ , para } i = 1, 2, \ldots, k-1,$$

$$y'_k = (1 - \lambda - \lambda P^k(k-1)) / t_{k'k} \tag{4.15}$$

La función a maximizar, en el paso k, es

$$\phi(\lambda) = \lambda \sum_{i=1}^{k-1} |y_i| + |y'_k| + \sum_{i=k+1}^{n} |P^i(k-1) + t_{ik} y'_k| \tag{4.16}$$

sujeta a que $\lambda \in (0, 1)$

Para obtener el máximo de ϕ, basta evaluarla en 0 y 1 para decidir cual es el mejor valor. Una vez obtenido el valor adecuado, se actualizan las variables, usando (4.15). En esta heuristica, después de cada paso de la substitución hacia adelante, el lado derecho queda con una entrada igual a 1 y el resto son cero, en otras palabras, la solución del sistema $T^t y = d$ es una columna de T^{-t}.

Uno de los problemas que se presentan al escribir el código para éste y

los métodos anteriores, en norma 1, es realizar en forma eficiente las dos evaluaciones de la función ϕ, para decidir cual es el mejor valor del parámetro, y una vez obtenido, actualizar las variables y los valores posteriores $P^1(k)$.

Por ejemplo, en el método retrospectivo el valor de ϕ para $\lambda = 0$ y 1, está dado por las relaciones siguientes:

si $\lambda = 1$, entonces $\phi(1) = \sum_{i=1}^{k-1} |y_i| + y'_k + \sum_{i=k+1}^{n} |P^1(k-1) + t_{ik} y'_k|$,

con $y'_k = -P^k(k-1) / t_{kk}$;

si $\lambda = 0$, entonces $\phi(0) = y'_k + \sum_{i=k+1}^{n} |P^1(k-1) + t_{ik} y'_k|$,

con $y'_k = 1 / t_{kk}$.

La actualización de las variables es muy simple; si el valor seleccionado fué 1, las primeras k-1 entradas del vector y no se alteran, en otro caso, se inicializan a cero. Para la nueva variable no existe problema pués es sólo un valor. Las cantidades $P^1(k-1)$ son actualizas suponiendo de antemano que el valor óptimo se alcanza con $\lambda = 1$. Si la comparación de las evaluaciones da un resultado contrario, entonces a cada $P^1(k)$ para $i = k+1, \ldots, n$ se le adiciona la cantidad

$$\frac{t_{ik}}{t_{kk}} (1 + P^1(k-1)) \quad \text{para } i = k+1, \ldots, n.$$

El subprograma DTRCO de Linpack fué modificado para realizar la estimación de acuerdo a esta heurística. En el apéndice C.2) se encuentra el código que realiza esta estrategia.

4.1.3.5 Divide y Vencerás.

La lista de estimadores heurísticos termina con uno de los métodos que produce mejores resultados, aunque a un costo ligeramente mayor. El método fué propuesto por Cline, Conn y Van Loan en [Cl3] y ha tenido poca atención en la literatura. Cline, en un trabajo posterior [Cl2], comenta que los resultados experimentales con este método y el retrospectivo, en la norma 1, son similares a Linpack.

El método se fundamenta en que una vez conocidas las soluciones de los sistemas triangulares superiores (inferiores),

$$T_{11}y_1 = d_1 \quad y \quad T_{22}y_2 = d_2,$$

pueden emplearse en la solución del sistema triangular superior (inferior)

$$\begin{bmatrix} T_{11} & T_{12} \\ 0 & T_{22} \end{bmatrix} \begin{bmatrix} z_1 \\ z_2 \end{bmatrix} = \begin{bmatrix} cd_1 \\ sd_2 \end{bmatrix}. \tag{4.17}$$

La solución de (4.17) es

$$z_2 = T_{22}^{-1} s\, d_2 = sy_2,$$

$$z_1 = T_{11}cd_1 - T_{11}^{-1}T_{12}z_2 = cy_1 - w, \text{ con } T_{11}w = T_{12}z_2.$$

Si la matriz T_{11} es de orden k y T_{22} de orden r, entonces, para resolver el sistema (4.17) es necesario realizar una multiplicación de una matriz (kxr), por un vector ($T_{12}z_2$), resolver un sistema triangular de orden r, ($T_{11}w = T_{12}z_2$) y sumar dos vectores de tamaño k.

En la estrategia divide y vencerás, el segundo miembro de la igualdad (4.17), se forma a partir de los múltiplos de los términos derechos (d_1 y d_2) de los sistemas iniciales, de manera que la norma de la solución sea máxima. Si además, se pide que $\|d_1\| = \|d_2\| = 1$, entonces los pesos se eligen de manera que el lado derecho resultante también tenga norma 1.

El problema se reduce a determinar los valores de los pesos c y s del sistema

$$\begin{bmatrix} T_{11} & T_{12} \\ 0 & T_{22} \end{bmatrix} \begin{bmatrix} z_1 \\ z_2 \end{bmatrix} = \begin{bmatrix} cd_1 \\ sd_2 \end{bmatrix}, \tag{4.18}$$

tal que $\|z\| = \|z_1\| + \|z_2\|$ sea máxima y $\|(cd_1, sd_2)^t\| = 1$.

Al igual que en el estimador retrospectivo, determinar los valores c y s que maximicen la norma de la solución, depende de la norma empleada.

Caso I. Norma espectral.

Las cantidades c y s pueden ser igual a cos(a) y sen(a) respectivamente, para algún $a \in [0, 2\pi]$. El problema se convierte en optimizar la función

(4.19) $$\phi(a) = \|z_1\|_2 + \|z_2\|_2$$

con $a \in [0, 2\pi]$

donde $$z_2 = sy_2$$

(4.20) $$z_1 = cy_1 - sw, \text{ con } T_{11}w = T_{12}z_2$$

Substituyendo (4.20) en (4.19) se obtiene

$$\phi(a) = s^2\|y_2\|_2^2 + c^2\|y_1\|_2^2 + s^2\|w\|_2^2 - 2scw^t y_1.$$

De la ecuación $\phi'(a) = 0$, se obtienen dos pares seno-coseno que dan los puntos extremos de $\phi(a)$. La derivada de ϕ resulta

$$\phi'(a) = 2sc(\|y_2\|_2^2 + \|w\|_2^2 - \|y_1\|_2^2) + 2w^t y_1(c^2 - s^2).$$

Si $\alpha = \|y_2\|_2^2 + \|w\|_2^2 - \|y_1\|_2^2$ y $\beta = w^t y_1$, entonces

$$\phi'(a) = sc\alpha + c^2\beta - s^2\beta = 0$$

(4.21) $$= \alpha\tan(a) + \beta - \beta\tan^2(a) = 0,$$

es una ecuación cuadrática en tan(a). Haciendo $r = \alpha / 2\beta$, las raíces de (4.21) son:

$$\mu_1 = r + \sqrt{1 + r^2} \quad y \quad \mu_2 = r - \sqrt{1 + r^2}.$$

Los pares seno-coseno que se obtienen de los puntos extremos de ϕ son:

$$s_1 = \frac{1}{\sqrt{1+\mu_1^2}} \qquad c_1 = \mu_1 s_1,$$

y $$s_2 = \frac{1}{\sqrt{1+\mu_2^2}} \qquad c_2 = \mu_2 s_2.$$

Substituyendo cada uno de estos pares en ϕ, es posible determinar donde se alcanza el máximo.

Caso II. La norma 1.

En esta norma las cantidades c y s son igual a λ y a $1-\lambda$ respectivamente, con $\lambda \in (0, 1)$. La solución del sistema (4.18) es:

$$z_2 = (1-\lambda)y_2$$

(4.22) $z_1 = \lambda y_1 - w$, con $T_{11} w = T_{12} z_2$

y la función a optimizar es

$$\phi(\lambda) = \|z_1\|_1 + \|z_2\|_1$$

con $\lambda \in (0, 1)$.

Para evaluar la función ϕ, en los dos puntos de interes, en forma eficiente, deben tenerse las expresiones de ϕ en cada uno de estos puntos.

(4.22) $$\phi(0) = \|y_2\|_1 + \|w\|_1$$

$$\phi(1) = \|y_1\|_1$$

Si $\|y_2\|_1 \geq \|y_1\|_1$, puede evitarse el calcular $\|w\|_1$. Una vez determinado el valor de λ adecuado, se emplea para obtener la solución del sistema.

La forma de emplear el método divide y venceras, en un sistema dado de orden n, es primero resolver n sistemas de orden 1x1, donde las matrices de los sistemas son los elementos de la diagonal de la matriz T. Sean T_1, T_2, ... T_n éstos sistemas y $n = 2p + q$, con $q = 0$ ó 1. Para $i = 1, 2, \ldots, q$ se combinan los sistemas T_{2i-1} y T_{2i} para obtener el sistema T'_i. Si $q = 0$ entonces se pasa a la etapa siguiente con los sistemas T'_1, T'_2, ... T'_p. En otro caso, primero se combinan T'_p y T_n para obtener el sistema T''_p y entonces se pasa a la etapa siguiente con los sistemas T'_1, T'_2, ..., T''_p. El mismo procedimiento se aplica a los nuevos sistemas obtenidos en cada etapa hasta obtener el sistema inicial.

El método termina con un lado derecho igual a cero excepto en una sola entrada, donde tiene el valor 1.

4.1.3 El método DIV-MOD

Hemos encontrado que es suficiente aplicar el método divide y vencerás al sistema directo, para obtener resultados sistemáticamente mejores que los obtenidos con otros métodos y en un menor número de operaciones. La cantidad obtenida al aplicar el método divide y vencerás, al sistema directo triangular, es la norma 1 de alguna columna de A^{-1}.

Al resolver cada uno de los sistemas del tipo (4.18) que aparecen en la solución del sistema original, se está seleccionando la columna con mayor norma de entre las columnas de cada subsistema. Por ejemplo, si las columnas

seleccionadas en los subsistemas fueron r y s respectivamente

$$U^{-1} = \begin{bmatrix} & T_{11}^{-1} & T_{12}^{-1} & \\ & & T_{22}^{-1} & \end{bmatrix},$$

(r: T_{11}^{-1}; s: T_{12}^{-1}, T_{22}^{-1})

entonces de la relación (4.22), se tiene que $\phi(0)$ es la norma de la columa s, siendo $\|y_2\|_1$ la norma de la columna s del bloque T_{22}^{-1} y $w = T^{-1}T_{12}z_2$ es la norma de la parte de la columna s que está fuera del bloque.

Con esta variante que proponemos, hemos encontrado, en el caso triangular, un ahorro de n/2 operaciones, pues se evita resolver el sistema transpuesto.

Cuando la matriz triangular es de orden $n = 2^k$, el número de operaciones requeridas para llevar a cabo la estrategia DIV-MOD es de $3n^2 + 3n$.

En el siguiente capitulo se presentan varios ejemplos donde se aplica este método y en el apéndice C.3) el código correspondiente.

4.1.4 Estimador via Optimización convexa.

El método fué popuesto por Hager en [Hal], para estimar la norma 1 de la inversa de una matriz A; tratando el problema de calcular $\|A^{-1}\|_1$ como un problema de optimización convexa.

Para $B \in \mathbb{R}^{n\times n}$, $\|B\|_1$ es el máximo global de la función convexa

(4.23) $$F(x) = \|Bx\|_1$$

sobre el conjunto convexo

(4.24) $$S = \{ x \mid \|x\|_1 \leq 1 \}$$

La función F es convexa, simétrica, más aún, lineal por pedazos, y el conjunto S define un poliedro. De la teoría de convexidad o simplemte de la definición de la norma 1, se sabe que la función F alcanza su máximo en un vértice de S.

Como la función F no es diferenciable en todo el conjunto S, *no* pueden aplicarse métodos donde se requieren la primera o segunda derivada. El método propuesto por Hager emplea algunos resultados del analisis no diferenciable y las propiedades de F y S para obtener, en el mejor de los casos, el máximo global de F en S.

Para explicar el algoritmo es necesario introducir algunas definiciones y resultados del análisis sin derivadas.

Definición 4.1. Sea $K \subseteq \mathbb{R}^n$. K es un conjunto convexo si para todo x, y $\in$ K se tiene $x_\lambda \in K$ donde

$$x_\lambda = (1 - \lambda)x + \lambda y \quad \forall\ \lambda \in [0,1].$$

Ejemplo. El conjunto definido en (4.24) es convexo. Sean x , y $\in$ S y $\lambda \in [0,1]$, entonces por la desigualdad del triángulo se tiene

$$\begin{aligned} \|(1-\lambda)x + \lambda y\|_1 &\leq |1-\lambda|\|x\|_1 + |\lambda|\|y\|_1 = (1-\lambda)\|x\|_1 + \lambda\|y\|_1 \\ &\leq (1-\lambda) + \lambda = 1. \end{aligned}$$

Definición 4.2. Sea $K \subseteq \mathbb{R}^n$ un conjunto convexo y $f:\mathbb{R}^n \to \mathbb{R}$. La función f es convexa si para todo x, y $\in$ K se tiene

(4.25) $$f(x_\lambda) \leq (1-\lambda)\, f(x) + \lambda\, f(y) \quad \forall\ \lambda \in [0,1].$$

La desigualdad (4.25) expresa el hecho que la grafica de una función convexa siempre está por debajo de la cuerda que une a los puntos (x, f(x)), (y, f(y)).

Si K es un conjunto abierto y f(x) es diferenciable en K, una definición equivalente de convexidad es que para todo x, y $\in$ K se tiene que

(4.26) $$f(y) \geq f(x) + \nabla f(x)^t(y - x).$$

Ejemplo. La función definida en (4.23) es una función convexa sobre el conjunto S.

Sean x, y $\in$ S entonces

$$\begin{aligned} F(x_\lambda) = \|Bx_\lambda\|_1 &= \|B((1-\lambda)x + \lambda y)\|_1 \\ &= \|(1-\lambda)Bx + \lambda By\|_1 \\ &\leq (1-\lambda)\|Bx\| + \lambda\|By\| \\ &= (1-\lambda)F(x) + \lambda F(y). \end{aligned}$$

Definición 4.3. Sea f una función de $K \subseteq \mathbb{R}^n$ en $\mathbb{R}$ con K convexo y $x \in \text{int}(K)$.

(a) Sea $a \in K$. Se dice que a es un subgradiente de f en x si

(4.27) $$f(y) \geq f(x) + a^t(y-x) \qquad \forall\ y \in K$$

(b) El conjunto de todos los subgradientes de f en x es llamado el

subdiferencial de f en x y es denotado por $\partial f(x)$.
(c) Cuando $\partial f(x) \neq \emptyset$ se dice que f es subdiferenciable en x.

Ejemplo. Sea $f: \mathbb{R} \to \mathbb{R}$ con $f(x) = |x|$. Claramente esta función no es diferenciable en 0, pero si es subdiferenciable en este punto. De la relación (4.27) se tiene

$$\frac{|y|}{y} a \leq 1 \quad \Leftrightarrow \quad a \in [-1, 1]$$

Este ejemplo puede generalizarse de la manera siguiente; si $f:(a, b) \to \mathbb{R}$ es convexa, entonces es subdiferenciable en todo punto $c \in (a, b)$ y además $\partial f(c) = [f'_-(c), f'_+(c)]$, donde $f'_-(c)$ y $f'_+(c)$ son la derivada izquierda y derecha de f en c respectivamente. En general se tiene que si f es una función convexa de $\mathbb{R}^n \to \mathbb{R}$ entonces esta es subdiferenciable [Val].

Otra propiedad importante de la generalización de derivada es que si la función f es diferenciable en x entonces $\partial f(x)$ tiene sólo un elemento y coincide con el $\nabla f(x)$.

El método propuesto por Hager y perfeccionado por Higham en [Hi3], trata de maximizar la función $F(x) = \|Bx\|_1$ en S. El método parte de un cierto punto del S denotado por x. Si F es diferenciable en x la desigualdad (4.26) asienta que la manera de encontrar un $y^* \in S$ donde la función sea mayor es maximizando la cantidad

$$\nabla F(x)^t(y - x).$$

Como $|\nabla F(x)^t y| \leq \|\nabla F(x)\|_\infty \|y\|_1 \leq \|\nabla F(x)\|_\infty$, y^* puede elegirse igual a un vértice de S, $(\pm e^1, i = 1, 2, \ldots, n)$ donde $\|\nabla F(x)\|_\infty = |\nabla F(x)_j|$. Por otra parte si $\|\nabla F(x)\|_\infty \leq \nabla F(x)^t x$ entonces x es un máximo local de F sobre S.

Como $F(x) = \|Bx\|_1$, el gradiente de F es igual a $\nabla F(x) = B^t\zeta$, donde

$$\begin{cases} \zeta_i = 1 & \text{Si } \sum_{j=1}^{n} b_{ij} x_j > 0 \\ \zeta_i = -1 & \text{Si } \sum_{j=1}^{n} b_j x_j < 0. \end{cases}$$

Ahora si en el punto inicial x la función F no es diferenciable entonces $\partial F(x)$ es no vacio y cada uno de los subgradientes tiene la forma $B^t\zeta$ con ζ_i igual que en la relación anterior, excepto cuando

subdiferencial de f en x y es denotado por $\partial f(x)$.
(c) Cuando $\partial f(x) = \emptyset$ se dice que f es subdiferenciable en x.

Ejemplo. Sea $f: \mathbb{R} \to \mathbb{R}$ con $f(x) = |x|$. Claramente esta función no es diferenciable en 0, pero si es subdiferenciable en este punto. De la relación (4.27) se tiene

$$\frac{|y|}{y} a \leq 1 \quad \Rightarrow \quad a \in [-1, 1]$$

Este ejemplo puede generalizarse de la manera siguiente; si $f:(a, b) \to \mathbb{R}$ es convexa, entonces es subdiferenciable en todo punto $c \in (a, b)$ y además $\partial f(c) = [f'_-(c), f'_+(c)]$, donde $f'_-(c)$ y $f'_+(c)$ son la derivada izquierda y derecha de f en c respectivamente. En general se tiene que si f es una función convexa de $\mathbb{R}^n \to \mathbb{R}$ entonces esta es subdiferenciable [Vai].

Otra propiedad importante de la generalización de derivada es que si la función f es diferenciable en x entonces $\partial f(x)$ tiene sólo un elemento y coincide con el $\nabla f(x)$.

El método propuesto por Hager y perfeccionado por Higham en [Hi3], trata de maximizar la función $F(x) = \|Bx\|_1$ en S. El método parte de un cierto punto del S denotado por x. Si F es diferenciable en x la desigualdad (4.26) asienta que la manera de encontrar un $y^* \in S$ donde la función sea mayor es maximizando la cantidad

$$\nabla F(x)^t(y - x).$$

Como $|\nabla F(x)^t y| \leq \|\nabla F(x)\|_\infty \|y\|_1 \leq \|\nabla F(x)\|_\infty$, y^* puede elegirse igual a un vértice de S, $(\pm e^i, i = 1, 2, \ldots, n)$ donde $\|\nabla F(x)\|_\infty = |\nabla F(x)_j|$. Por otra parte si $\|\nabla F(x)\|_\infty \leq \nabla F(x)^t x$ entonces x es un máximo local de F sobre S.

Como $F(x) = \|Bx\|_1$, el gradiente de F es igual a $\nabla F(x) = B^t\zeta$, donde

$$\begin{cases} \zeta_i = 1 & \text{Si } \sum_{j=1}^{n} b_{ij} x_j > 0 \\ \zeta_i = -1 & \text{Si } \sum_{j=1}^{n} b_{ij} x_j < 0. \end{cases}$$

Ahora si en el punto inicial x la función F no es diferenciable entonces $\partial F(x)$ es no vacio y cada uno de los subgradientes tiene la forma $B^t\zeta$ con ζ_i igual que en la relación anterior, excepto cuando

$$\sum_{j=1}^{n} b_{ij}x_j = 0;$$

en tal caso ζ_i puede tomar cualquier valor en [-1, 1].

La desigualdad (4.27) sugiere la forma de elegir un subgradiente y de moverse de x al punto $y^* \in S$; ésto se logra maximizando la cantidad $a^t(y-x)$. Como $|a^t y| \leq \|a\|_\infty$ y $F(x) = F(-x)$, y^* puede elegirse igual al vértice e^j donde $|a_j| = \|a\|_\infty$. Si $\|a\|_\infty > a^t x$ entonces $F(y^*) > F(x)$ esta asegurada.

Cuando $\|a\|_\infty \leq a^t x$ se ha llegado a un máximo que en éste caso, no se asegura que sea global a S.

Hager en [Hal] concretó los resultados anteriores en el siguiente algoritmo.

Algoritmo . Dada una matriz $A \in \mathbb{R}^n$ el algoritmo estima $\gamma \leq \|A^{-1}\|_1$.

1) Elegir x con $\|x\|_1 = 1$

Repite

2) Resolver $Ay = x$

3) $\zeta = \mathrm{sgn}(y)$

4) Resolver $A^t z = \zeta$

5) Si $\|z\|_\infty \leq z^t x$ entonces termina con $\gamma = \|y\|_1$

6) $x = e^j$, donde $|z_j| = \|z\|_\infty$

Sgn es la función de $\mathbb{R}^n \to \mathbb{R}^n$ definida por

$$\mathrm{sgn}(x) = (s_i), \qquad s_i = \begin{cases} 1 & \text{si } x_i \geq 0 \\ -1 & \text{si } x_i < 0. \end{cases}$$

El algoritmo parte de un punto elegido en la frontera de S, y cuando la función no es diferenciable se elige un subgradiente particular. En el paso 3) si y_i es cero entonces $\zeta_i = 1$ y con estos valores es calculado el subgradiente en el paso 4). La teoría garantiza que el punto x donde termina el algoritmo es un máximo local si $y = A^{-1}x$ tiene todas las entradas diferentes de cero. Si el vector y tiene entradas cero ésto no puede asegurarse.

Ejemplo 4.3. Sea H_3 la matriz de Hilbert de orden 3. H_3^{-1} esta dada por

$$H_3^{-1} = \begin{bmatrix} 9 & -36 & 30 \\ -36 & 192 & -180 \\ 30 & -180 & 180 \end{bmatrix}.$$

Aplicando el algoritmo anterior a matriz A, con $x = [\frac{1}{3}, \frac{1}{3}, \frac{1}{3}]$ como valor inicial, se obtiene, en la primera pasada:

Paso 2) Resolver $Ay = x \Rightarrow y = [1, -8, 10]^t$

Paso 3) $\zeta = [1, -1, 1]^t$

Paso 4) Resolver $A^t z = \zeta \Rightarrow z = [75, -408, 390]^t$

Paso 5) Como $\|z\|_\infty = 408$ y $z^t x = 19$ ejecutar paso 6)

Paso 6) $x = [0, 1, 0]^t$.

En la segunda pasada se tiene

Paso 2) Resolver $Ay = x \Rightarrow y = [-36, 192, -180]^t$

Paso 3) $\zeta = [-1, 1, -1]^t$

Paso 4) Resolver $A^t x = \zeta \Rightarrow z = [-75, 408, -390]$

Paso 5) Como $\|z\|_\infty = 408$, $z^t x = 408$ y $\|z\|_\infty \leq z^t x$.

Termina con $\gamma = 408$.

Un caso especial de este algoritmo es cuando la matriz A es una matriz M. En la primera iteración del algoritmo $\|z\|_\infty = \|A^{-1}\|_\infty$, y el algoritmo termina a lo más en dos iteraciones.

Higham en [Hi3], presenta una ligera variante del método anterior, el cual lo salva de posibles oscilaciones, debido a errores de redondeo. También evita resolver dos veces el mismo sistema, como sucede con las matrices de comparación M.

4.2. Matrices tridiagonales.

Para matrices con cierta estructura existen algoritmos más adecuados para estimar el número de condición, en particular, para estimar la norma de la matriz inversa. En ciertas aplicaciones aparecen en forma natural las matrices tridiagonales, por ejemplo, en los métodos de diferencias para problemas a la frontera, en el manejo de Splines y en la solución numérica de relaciones de recurrencia lineales de segundo orden. Una matriz tridiagonal tiene la estructura siguiente:

(4.28) $$A = \begin{bmatrix} a_1 & c_1 & & & \\ b_2 & a_2 & c_2 & & \\ & \ddots & \ddots & \ddots & \\ & & & c_{n-1} & \\ & & b_n & a_n & \end{bmatrix}.$$

Higham presenta en [Hi2], algunos algoritmos eficientes para calcular $\|A^{-1}\|$ en la norma 1 o ∞, cuando la matriz es **irreducible**, es decir, $b_2, \ldots b_n$ y $c_1, c_2, \ldots c_{n-1}$ son diferentes de cero.

La matriz A puede representarse por O(n) cantidades: los elementos diferentes de cero . El teorema siguiente demuestra que también la matriz inversa puede representarse en términos de O(n) cantidades, aunque A^{-1} tiene, en general, n^2 elementos diferentes de cero.

Teorema 4.1: Sea A una matriz tridiagonal no singular.

(1) Si A es irreducible y $A^{-1} = (\alpha_{ij})$ entonces existen vectores x, y, p y q tales que:

(4.29) $$\alpha_{ij} = \begin{cases} x_i y_j & i \le j \\ p_i q_j & i \ge j \end{cases}$$

(2) Si A es reducible existen dos casos:

(a) Si $c_1 = 0$, entonces la matriz A tiene la estructura

$$\begin{bmatrix} A_1 & 0 \\ B_1 & A_2 \end{bmatrix},$$

donde $A_1 \in \mathbb{R}^{1\times 1}$ y $A_2 \in \mathbb{R}^{(n-1)\times(n-1)}$ son tridiagonales, y

$$A^{-1} = \begin{bmatrix} A_1^{-1} & 0 \\ X & A_2^{-1} \end{bmatrix},$$

donde $X \in \mathbb{R}^{(n-1)\times 1}$ es una matriz de rango 1, si $b_{i+1} = 0$. El teorema se aplica recursivamente a las submatrices A_1 y A_2.

(b) Si $b_{i+1} = 0$, la parte (a) del teorema puede aplicarse a A^t.

Dem: Se encuentra en [Ik1].

De acuerdo con este resultado, la matriz inversa puede escribirse como:

(4.30) $$A^{-1} = \begin{bmatrix} x_1y_1 & x_1y_2 & \cdots & x_1y_n \\ p_1q_1 & x_2y_2 & \cdots & x_2y_n \\ \vdots & \vdots & & \vdots \\ p_nq_1 & p_nq_2 & \cdots & x_n'y_n \end{bmatrix}.$$

Para calcular la norma infinita de la matriz inversa, es necesario obtener la suma de los valores absolutos de los elementos de cada renglón. Para el renglon i-ésimo se tiene

$$|p_iq_1| + |p_iq_2| + \ldots + |p_iq_{i-1}| + |x_iy_i| + \ldots + |x_iy_n|,$$

o en forma equivalente

(4.31) $$|p_i|(|q_1| + \ldots + |q_{i-1}|) + |x_i|(|y_i| + \ldots + |y_n|)$$

Esta expresión proporciona un método eficiente para obtener la norma infinita de la inversa. Las sumas que aparecen entre paréntesis, pueden emplearse para calcular la suma de los valores absolutos de las entradas del renglón $i + 1$; solo debe agregarse un término más.

Para tener el algoritmo completo es necesario dar un método para obtener los vectores x, y, p y q. El método empleado es igual al propuesto por Ikebe [Iki] para determinar la estructura de la inversa de una matriz de Hessemberg.

Los vectores x e y pueden calcularse a partir de los sistemas

(4.32) $$A\,x = y_n^{-1}\,e_n^t,$$

(4.33) $$A^t y = x_1^{-1} e_1,$$

que corresponden a los sistemas definidos por la última columna de $AA^{-1} = I$ y el primer renglón de $A^{-1}A = I$. Los vectores e_1 y e_n son el primero y el n-ésimo vector unitario. Los vectores p y q pueden obtenerse de manera similar usando A^t en lugar de A. Existe un grado de libertad para resolver los sistemas (4.32) y (4.33). Una posibilidad es elegir $x_1 = 1$.

La programación de este algoritmo se encuentra en el apéndice C.6).

Existe redundancia en la representación (4.30) de A^{-1}, los cuatro vectores x, y, p y q contienen 4n-2 valores "libres" mientras A depende solo de 3n-2 cantidades. Esta diferencia se debe a que el teorema anterior considera las partes superior e inferior de A por separado. El teorema siguiente proporciona una representación más concisa.

Teorema 4.2.Sea A una matriz tridiagonal no singular e irreducible. Entonces existen vectores x y y tal que $A^{-1} = (a_{ij})$ esta dada por

$$a_{ij} = \begin{cases} x_i y_j d_j & i \le j \\ y_i x_j d_j & i \ge j, \end{cases}$$

donde

$$d_j = \prod_{r=1}^{j-1} \left(\frac{c_r}{b_{r+1}} \right), \qquad 1 \le j \le n.$$

Dem: Se encuentra en [Hi2].

De acuerdo a este resultado, la inversa de la matriz A tiene la representación siguiente:

$$A^{-1} = \begin{bmatrix} x_1 y_1 d_1 & x_1 y_2 d_2 & \cdots & x_1 y_n d_n \\ y_2 x_1 d_1 & x_2 y_2 d_2 & \cdots & x_2 y_n d_n \\ \vdots & \vdots & & \vdots \\ y_n x_1 d_1 & y_n x_2 d_2 & \cdots & x_n y_n d_n \end{bmatrix}. \tag{4.34}$$

Los vectores x e y pueden obtenerse, como en el caso anterior, de la solución de los sistemas algebraicos definidos por la última columna de $AA^{-1} = I$ y del primer renglón de $A^{-1}A = I$:

$$Ax = (y_n d_n)^{-1} e_n ,$$

$$A^t y = x_1^{-1} e_1 .$$

Como el valor d_j es común a todos los elementos de una columna de A^{-1}, resulta más eficiente calcular la norma 1 de A^{-1} en lugar de la norma infinito.

La programación del algoritmo se encuentra en el apéndice (B6).

Resumiendo. En este capítulo hemos presentado sólo los métodos de estimación para dos casos especiales de matrices, debido a su gran importancia práctica. Quedaron fuera de este trabajo otros casos, que deberían ser estudiados, para completez del tema.

CAPITULO 5

EJEMPLOS Y CONTRAEJEMPLOS

5.1 Matrices de prueba
5.2 Comparación de los métodos
5.3 El método DIV-MOD
5.4 Contraejemplos.

Este capítulo esta dedicado a comparar algunos de los métodos de estimación del número de condición de matrices, cubiertos en el capítulo anterior, proporcionar ejemplos donde uno o varios métodos fallan y presentar los resultados obtenidos con el nuevo método (DIV-MOD).

Cuando el método aproxima por arriba al número de condición, el mal condicionamiento de una matriz puede sobreestimarse y parecer mal condicionada, o bien, parecer una matriz bien condicionada, cuando el método aproxima por abajo este número.

El criterio empleado para comparar los métodos, es el cociente

$$S(A) = K_1^{\sim}(A) \ / \ K_1(A),$$

donde $K_1^{\sim}(A)$ es el estimador del número de condición de la matriz A, $K_1(A)$. Este cociente es menor o igual que 1 para los métodos que producen estimadores por debajo de $K_1(A)$. Cuando los estimadores sean cotas superiores del valor real, el reciproco del cociente será empleado.

Los métodos de estimación fueron programados para la norma 1 y probados extensivamente con un gran número de matrices de prueba: matrices triangulares superiores obtenidas de la factorización LU o QR de ciertas matrices. Cinco tipos diferentes de matrices fueron empleados, más la factorización LU ó QR, hacen un total de 10 conjuntos de matrices de prueba.

En la primera sección se describen los conjuntos de matrices de prueba.

En las dos secciones posteriores se presentan los resultados de las comparaciones, y la última está dedicada a los contraejemplos.

5.1 Matrices de prueba.

5.1.1 Matriz de Hilbert. La matriz de Hilbert puede aparecer al resolver un problema de aproximación por medio del criterio de mínimos cuadrados. Si la función de peso es $w(x) = 1$ y el intervalo de interes es $[0,1]$, entonces el problema de mínimos cuadrados se reduce al sistema algebraico lineal

$$\sum_{j=0}^{n} \frac{a_j}{i+j-1} = \int_0^1 f(x)x^i dx \qquad i = 0, 1, \ldots, n. \tag{5.1}$$

La matriz de coeficientes de (5.1) es la matriz de Hilbert de orden n+1, H_{n+1}. La solución del sistema (5.1) es extremadamente sensitiva a pequeños cambios de las entradas de la matriz y del lado derecho.

El número de condición de la matriz H_n crece exponencialmente respecto a n. Savage en 1954, [Sa1], cálculo el número de condición, los valores y vectores propios a las diez primeras matrices de Hilbert. Ahora ya se conoce más acerca de esta matriz, por ejemplo, se tienen expresiones exactas para el determinante y la inversa [Is1], además de aproximaciones asintóticas para el número de condición $K_1(A)$, [Gr1].

5.1.2. Matriz de Vandermonde. La matrix de Vandermonde aparece de manera natural en el problema de interpolación de una función por medio de un polinomio.

La matriz de Vandermonde de orden n es una matriz de la forma:

$$V = V(x_1, x_2, \ldots, x_n) = \begin{bmatrix} 1 & 1 & \ldots & 1 \\ x_1 & x_2 & \ldots & x_n \\ \ldots & \ldots & \ldots & \ldots \\ x_1^{n-1} & x_2^{n-1} & \ldots & x_n^{n-1} \end{bmatrix},$$

donde los nodos x_k son números reales o complejos.

No se conocen relaciones explícitas para el número de condición de la matriz $V_n(x)$ en diferentes normas. La mayoría de los resultados están

relacionados con la norma ∞ y cuando los nodos satisfacen cierta propiedad.

Gautschi en [Ga1], [Ga2, [Ga7], [Ga8] y [Ga12] ha encontrado cotas para la norma infinita de la matriz y su inversa, además de expresiones explícitas para el número de condición de la matriz V_n cuando los nodos son reales y no negativos o bien están distribuidos simétricamente respecto al origen, es decir, $x_i + x_{n-i+1} = 0$, para i = 1, 2, .., n.

5.1.3 Matrices Aleatorias. Estas matrices fueron generadas de manera que sus entradas fueron elegidas aleatoriamente con distribución uniforme en [-1, 1].

5.1.4 Matrices con cierta distribución de valores singulares. Stewart, en [St2], propone un método barato para construir matrices aleatorias ortogonales, que pueden emplearse para obtener matrices de prueba, con cierta distribución de sus valores singulares. Las dos distribuciones típicas, son:
primera, los primeros n - 1 valores singulares son iguales a 1 y el n-ésimo a $1 / K_2(A)$, donde $K_2(A)$ es el número de condición espectral deseado para la matriz A, es decir:

$$\sigma_1 = \sigma_2 = \ldots = \sigma_{n-1} = 1, \quad \sigma_n = 1 / K_2(A).$$

La segunda distribución es un decrecimiento exponencial de los valores singulares:

$$\sigma_2 / \sigma_1 = \sigma_3 / \sigma_2 = \ldots = \sigma_n / \sigma_{n-1}$$

con

$$\sigma_1 = 1 \text{ y } \sigma_n = 1 / K_2(A).$$

La forma de obtener estas matrices es partir de una matriz diagonal Σ con la distribución deseada de valores singulares y de dos matrices ortogonales aleatorias U y V. Por el teorema de descomposición en valores singulares, se tiene que $A = V^t \Sigma U$ es una matriz con sus valores singulares distribuidos de acuerdo a la distribución establecida por Σ.

El método de generación de matrices ortogonales aleatorias propuesto por Stewart está basado en el resultado siguiente: Si A es una matriz n x n, donde las entradas son elegidas en forma independiente y aleatoria con distribución normal $n(0, \sigma^2)$ y las matrices Q y R son las matrices obtenidas por la factorización QR normalizada, de manera que los elementos de la diagonal de R

sean positivos, entonces las matrices Q estan distribuidas sobre el conjunto de matrices ortogonales de orden n, O_n, de acuerdo a la distribución μ, la medida de Haar, [St2]. El teorema de Haar [Ha3], asegura la existencia de una única medida normalizada e invariante bajo traslaciones izquierdas para el grupo topológico compacto de matrices ortogonales con cualquiera de las normas de matrices como métrica, esto es $\mu(O_n) = 1$ y $\mu(HM) = \mu(M)$ para cualquier conjunto medible $M \subseteq O_n$ y cualquier $H \in O_n$.

El método para generar las matrices ortogonales que surge en forma natural es factorizar una matriz con entradas distribuidas normalmente y quedarse con la matriz ortogonal Q. Este método es costoso en número de operaciones y espacio. Una mejor alternativa está basada en el teorema siguiente [St2].

Teorema: Sean x_1, x_2, ..., x_{n-1} vectores en R^n, R^{n-1}, ..., R^2 respectivamente, con distribución $n(0, \sigma^2)$. Sean H_j, j = 1, 2, .., n-1, las transformaciones de Householder que reducen a x_j en $r_{jj}e_1$. Sea $H_j = (I_{j-1}, H_j)$ y $D = diag(sgn(r_{11}), sgn(r_{22}), ..., sgn(r_{nn}))$. Entonces el producto $Q = D H_1H_2... H_n$ es una matriz ortogonal aleatoria distribuida de acuerdo a la medida de Haar sobre el conjunto de matrices ortogonales de orden n, O_n.

Dem: Se encuentra en [St2].

Este resultado sugiere una forma de construir una matriz ortogonal con requerimientos de memoria reducidos; unicamente se necesitan $n^2/2$ localidades de memoria para almacenar los vectores x_1, x_2, ..., x_n.

En ciertas aplicaciones no es necesario conocer explicitamente la matriz Q. Por ejemplo, para evaluar el producto Qy, sólo se requieren n^2 operaciones para aplicar consecutivamente las trasformaciones de Householder.

Este método fué programado y empleado en la generación de las matrices de prueba. El listado correspondiente aparece en el apéndice B.2. El subprograma RANDMS de la libreria IMSL fué empleado para la generación de los vectores aleatorios con distribución $n(0, \sigma^2)$.

Las matrices de prueba fueron las matrices triangulares superiores de las factorizaciones LU y QR de las matrices arriba mencionadas. La combinación de los diferentes tipos de matrices y las factorizaciones, dan un total de 10 conjuntos de matrices de prueba.

RND+LU Matrices triangulares superiores, obtenidas de la factorización LU de matrices con entradas distribuidas uniformemente en [-1, 1].

RND+QR Igual que en el caso anterior sólo que con la factorización QR.

SLT+LU Matrices triangulares superiores, obtenidas de la factorización LU de matrices con distribución de valores singulares igual a: $\sigma_1 = \sigma_2 = \ldots = \sigma_{n-1} = 1$ y $\sigma_n = 1/K(A)$. $K(A) = 10,\ 10^2,\ 10^3,\ 10^4$ y 10^5.

SLT+QR Igual que en el conjunto anterior sólo que con factorización QR.

DXP+LU Matrices triangulares superiores, obtenidas de la factorización LU de matrices con distribución de valores singulares igual a: $\sigma_1 = 1$, $\sigma_n = 1/K(A)$, $\sigma_2/\sigma_1 = \sigma_3/\sigma_2 = \ldots = \sigma_n/\sigma_{n-1}$. $K(A) = 10,\ 10^2,\ 10^3,\ 10^4$ y 10^5.

DXP+QR Igual que el conjunto anterior sólo que con factorización QR.

HLB+LU Matrices triangulares, obtenidas de la factorización LU de la matriz de Hilbert; $h_n(i, j) = 1 / (i + j - 1)$, $i,j = 1, 2, \ldots, n$. Para $n = 2, 3, \ldots, 20$.

HLB+QR Igual que el conjunto anterior pero con factorización QR.

VAN+LU Matrices triangulares, obtenidas de la factorización LU de la matriz de Vandermonde; $V(x_1, x_2, \ldots, x_n)$ con x_i = i-ésima raiz del polinomio de Chebyshev de orden n, $T_n(x)$. Para $n = 2, 3, \ldots, 20$.

VAN+QR Igual que el conjunto anterior pero con factorización QR.

5.2 Comparación de los métodos.

Los métodos para su comparación se han dividido en tres grupos: primero,

estimadores tipo Linpack; segundo, estimadores vía matrices de comparación y tercero, el método vía optimización convexa. La norma 1 fué empleada en todos los métodos y quedaron fuera de comparación los métodos de estimación de número de condición de matrices tridiagonales y algunas variantes de método LINPACK. La mayoría de los experimentos numéricos se realizaron en doble precisión en una computadora IBM-AT sin coprocesador aritmético y fueron empleados los lenguajes Fortran (Mirosoft 3.3) y C (Turbo C, V2.0).

5.2.1 Estimadores tipo Linpack. Este grupo comprende los estimadores:

DECOMP Aparece en el subprograma DECOMP y SOLVE,

LINPACK Estimador empleado por algunos subprogramas del paquete Linpack (DGECO, DTRCO, etc),

RETRO Estimador retrospectivo y

DIV-VNC Divide y venceras.

Este tipo de métodos es el más popular, en especial, DECOMP y LINPACK. Los resultados que a continuación se presentan corresponden a matrices triangulares. Sin embargo, como se vió en el capítulo anterior, los heurísticas pueden aplicarse a otro tipo de matrices.

En las tablas (5.1) a (5.6), se presentan los resultados obtenidos con estos estimadores, para las matrices de prueba SLT*LU y DXP*LU.

K(A)	n = 5	10	20	30	40
10^1	0.682	0.456	0.321	0.472	0.426
10^2	0.946	0.879	0.790	0.848	0.813
10^3	0.994	0.988	0.974	0.980	0.973
10^4	0.999	0.999	0.999	0.998	0.997
10^5	1.0	1.0	1.0	1.0	1.0

Tabla (5.1). Cocientes de estimación, S(A), promedio obtenidos por el método DECOMP aplicado a las matrices de prueba SLT*LU.

K(A)	n = 5	10	20	30	40
10^1	0.694	0.502	0.349	0.480	0.434
10^2	0.946	0.898	0.799	0.850	0.815
10^3	0.994	0.989	0.975	0.980	0.974
10^4	0.999	0.999	0.998	0.998	0.997
10^5	1.0	1.0	1.0	1.0	1.0

Tabla (5.2). Cocientes de estimación, S(A), promedio para el método LINPACK aplicado a las a las matrices de prueba SLT*LU.

K(A)	n = 5	10	20	30	40
10^1	0.979	0.993	1.0	1.0	1.0
10^2	0.997	0.999	1.0	1.0	1.0
10^3	1.0	1.0	1.0	1.0	1.0
10^4	1.0	1.0	1.0	1.0	1.0
10^5	1.0	1.0	1.0	1.0	1.0

Tabla (5.3). Cocientes de estimación, S(A), promedio para el método RETRO aplicado a las matrices de prueba SLT*LU.

K(A)	n = 5	10	20	30	40
10^1	0.546	0.443	0.345	0.319	0.322
10^2	0.655	0.499	0.376	0.330	0.382
10^3	0.800	0.592	0.456	0.390	0.441
10^4	0.871	0.667	0.503	0.534	0.482
10^5	0.940	0.703	0.536	0.548	0.561

Tabla (5.4). Cocientes de estimación, S(A), promedio par el método DECOMP aplicado a las matrices de prueba DXP*LU.

K(A)	n = 5	10	20	30	40
10^1	0.579	0.479	0.411	0.392	0.387
10^2	0.678	0.538	0.455	0.417	0.442
10^3	0.798	0.600	0.515	0.469	0.503
10^4	0.881	0.677	0.566	0.568	0.536
10^5	0.936	0.711	0.582	0.594	0.591

Tabla (5.5). Cocientes de estimación, S(A), promedio para el método LINPACK aplicado a las matrices de prueba DXP+LU.

K(A)	n = 5	10	20	30	40
10^1	0.950	0.943	0.838	0.814	0.804
10^2	0.929	0.926	0.961	0.885	0.900
10^3	0.962	0.970	0.946	0.921	0.946
10^4	0.982	0.984	0.944	0.955	0.948
10^5	0.993	0.923	0.973	0.883	0.923

Tabla (5.6). Cocientes de estimación, S(A), promedio para el método RETRO aplicado a las matrices de prueba DXP+LU.

La primera observación que se desprende de las tablas y que ha sido comentada por otros autores, es que en la práctica estos métodos no están por debajo del 10% del número de condición. Sin embargo pueden construirse ejemplos donde esta observación no se cumple. En la sección 5.4 se darán varios ejemplos.

Los tres métodos probados son sensitivos al tamaño de la matriz y a su mal condicionamiento. Entre mayor es el orden de la matriz menor es la aproximación. Se presenta una relación inversa con el mal condicionamiento, pues mejora la estimación mientras el número de condición de la matriz del sistema es mayor. Esto último es cierto, aunque no depende exclusivamente del

número de condición, sino también de cierta estructura propia de cada matriz. Por ejemplo, las tablas con los datos tipo SLT y DXP, muestran que aunque las matrices tienen el mismo número de condición, la calidad de la estimación depende de las distribución de sus valores singulares. Una distribución exponencial de éstos hace más difícil la estimación.

5.2.2 Estimadores vía matrices de comparación.

Fueron empleadas las matrices de prueba M(T), W(T) y Z(T), para estimar el número de condición de las matrices de prueba; se encontró que para un gran número de matrices de prueba, el cociente de comparación es inferior a 0.001.

En las tablas (5.7) a (5.11) se presentan los resultados, obtenidos por medio de las matrices de comparación, donde el estimador no estuvo muy por arriba del valor real.

Algunas observaciones que se desprenden de estas tablas son; los estimadores vía matrices de comparación son de menor complejidad y exactitud. Los métodos son poco sensibles al mal condicionamiento de la matriz, en comparación con los del grupo anterior. También, al aumentar el número de condición no se obtiene una mejor estimación.

K(A)	n = 5	10	20	30	40
10^1	0.874	0.718	0.652	0.557	0.579
10^2	0.869	0.697	0.546	0.561	0.573
10^3	0.883	0.683	0.583	0.559	0.573
10^4	0.858	0.686	0.624	0.559	0.573
10^5	0.844	0.628	0.502	0.559	0.573

Tabla (5.7). Cocientes de estimación promedio para el método MT aplicado a las matrices SLT+LU.

K(A)	n = 5	10	20	30	40
10^1	0.605	0.279	0.126	0.054	0.043
10^2	0.606	0.313	0.098	0.065	0.083
10^3	0.620	0.267	0.115	0.064	0.076
10^4	0.606	0.288	0.149	0.063	0.076
10^5	0.619	0.240	0.098	0.063	0.075

Tabla (5.8). Cocientes de estimación promedio para el método WT aplicado a las matrices de prueba SLT+LU.

K(A)	n = 5	10	20	30	40
10^1	0.204	0.004	0.0	0.0	0.0
10^2	0.124	0.003	0.0	0.0	0.0
10^3	0.164	0.002	0.0	0.0	0.0
10^4	0.176	0.008	0.0	0.0	0.0
10^5	0.139	0.004	0.0	0.0	0.0

Tabla (5.9). Cocientes de estimación promedio para el método ZT aplicado a las matrices SLT+LU.

K(A)	n = 5	10	20	30	40
10^1	0.748	0.609	0.428	0.344	0.366
10^2	0.720	0.579	0.382	0.319	0.341
10^3	0.665	0.516	0.384	0.319	0.329
10^4	0.604	0.420	0.396	0.300	0.327
10^5	0.566	0.366	0.356	0.233	0.282

Tabla (5.10). Cocientes de estimación promedio para el método MT aplicado a las matrices SXP+LU.

K(A)	n = 5	10	20	30	40
10^1	0.297	0.080	0.018	0.007	0.001
10^2	0.088	0.007	0.0	0.0	0.0
10^3	0.005	0.0	0.0	0.0	0.0
10^4	0.0	0.0	0.0	0.0	0.0
10^5	0.0	0.0	0.0	0.0	0.0

Tabla (5.11). Cocientes de estimación promedio para el método WT aplicado a las matrices DXP+LU.

5.2.3 Estimador vía optimización convexa.

Los resultados obtenidos por este método aparecen en las tablas (5.12) y (5.13).

Sin duda, el estimador vía optimización convexa, es el mejor estimador general de todos los presentados en este trabajo. También es sensitivo al tamaño de la matriz y al mal condicionamiento de la matriz pero en menor grado que los estimadores tipo Linpack. El número de iteraciones en la mayoría de los casos fué solamente de 2.

Otra propiedad de este estimador, es su independencia respecto a la estructura de datos empleada para representar la matriz, sólo es necesario contar con un subprograma para resolver sistemas del tipo $A\,x = b$ y $A^t x = b$.

K(A)	n = 5	10	20	30	30
10^1	1.0	1.0	1.0	1.0	1.0
10^2	1.0	1.0	1.0	1.0	1.0
10^3	1.0	1.0	1.0	1.0	1.0
10^4	1.0	1.0	1.0	1.0	1.0
10^5	1.0	1.0	1.0	1.0	1.0

Tabla (5.12). Cocientes de estimación promedio para el método OPT-CNV aplicado a las matrices de prueba SLT*LU.

K(A)	n = 5	10	20	30	40
10^1	0.949	0.947	0.945	0.880	0.916
10^2	1.000	0.983	0.958	0.951	0.962
10^3	1.000	0.976	0.961	0.952	0.987
10^4	1.000	1.000	0.970	0.958	0.986
10^5	1.000	1.000	0.981	1.000	0.983

Tabla (5.13). Cocientes de estimación promedio para el método OPT-CNV aplicado a las matrices de prueba DXP*LU.

5.3. El metodo DIV-MOD.

El método propuesto, DIV-MOD, fué probado extensamente, con diferentes matrices de prueba y los resultados fueron comparados con los obtenidos por otros métodos. Las tablas (5.14) y (5.15) presentan los resultados de este método aplicado a matrices con cierta distribución de sus valores singulares.

K(A)	n = 5	10	20	30	40
10^1	1.0	1.0	1.0	1.0	1.0
10^2	1.0	1.0	1 0	1.0	1.0
10^3	1.0	1.0	1.0	1.0	1.0
10^4	1.0	1.0	1.0	1.0	1.0
10^5	1.0	1.0	1.0	1.0	1.0

Tabla (5.14). Cocientes de estimación promedio para el método DIV-MOD aplicado a las matrices de prueba SLT+LU.

K(A)	n = 5	10	20	30	40
10^1	1.000	0.971	0.848	0.780	0.821
10^2	1.000	0.976	0.995	0.849	0.900
10^3	1.000	1.000	0.950	0.934	0.946
10^4	1.000	1.000	0.976	0.985	0.948
10^5	1.000	1.000	0.989	0.934	0.961

Tabla (5.15). Cocientes de estimación promedio para el método DIV-MOD aplicado a las matrices de prueba DXP+LU.

Los resultados obtenidos al aplicar los métodos anteriores a las matrices: HLB+LU, HLB+QR, VAN+LU y VAN+QR, se presentan en las tablas (5.16) a (5.19).

N	DECOMP	LINPACK	RETRO	OPT-CNV	DIV-MOD	M(T)
2	0.950	0.950	1.000	1.000	1.000	1.000
4	0.956	0.956	1.000	1.000	1.000	0.441
6	0.981	0.981	0.992	1.000	1.000	0.073
8	0.989	0.989	0.985	1.000	1.000	0.006
10	0.988	0.988	0.975	1.000	1.000	0.000
12	0.989	0.989	0.969	1.000	1.000	0.000
14	0.894	0.894	0.964	1.000	1.000	0.000
16	0.773	0.708	0.677	1.000	1.000	0.000
18	0.489	0.522	1.000	1.000	1.000	0.000
20	0.460	0.400	0.962	1.000	1.000	0.000

Tabla (5.16). Cociente de estimación para los diferentes métodos aplicados a las matrices HLB+LU.

N	DECOMP	LINPACK	RETRO	OPT-CNV	DIV-VNC	M(T)
2	0.960	0.960	1.000	1.000	1.000	1.000
4	0.979	0.979	1.000	1.000	1.000	0.428
6	0.984	0.984	0.994	1.000	1.000	0.071
8	0.986	0.986	0.981	1.000	1.000	0.006
10	0.988	0.988	0.975	1.000	1.000	0.000
12	0.989	0.989	0.970	1.000	1.000	0.000
14	0.835	0.835	0.922	1.000	1.000	0.000
16	0.977	0.977	0.989	1.000	1.000	0.000
18	0.341	0.545	1.000	1.000	1.000	0.000
20	0.569	0.692	0.397	0.930	1.000	0.000

Fig (5.17) Cociente de estimación para los diferentes métodos aplicados a las matrices HLB+QR.

N	DECOMP	LINPACK	RETRO	OPT-CNV	DIV-MOD	M(T)
2	0.879	0.879	0.828	1.000	1.000	1.000
4	0.759	0.759	0.568	0.837	1.000	0.450
6	0.664	0.576	0.447	1.000	0.523	0.078
8	0.750	0.701	0.843	1.000	1.000	0.005
10	0.642	0.606	0.788	1.000	1.000	0.000
12	0.605	0.580	0.835	1.000	1.000	0.000
14	0.682	0.682	0.793	1.000	1.000	0.000
16	0.668	0.668	0.791	1.000	1.000	0.000
18	0.535	0.478	0.623	1.000	0.506	0.000
20	0.596	0.552	0.743	1.000	1.000	0.000

Tabla (5.18). Cociente de estimación de los diferentes métodos aplicados a las matrices VAN+LU

N	DECOMP	LINPACK	RETRO	OPT-CNV	DIV-MOD	M(T)
2	0.879	0.879	1.000	1.000	1.000	1.000
4	0.850	0.850	0.991	0.531	1.000	0.447
6	0.754	0.754	0.779	1.000	1.000	0.071
8	0.697	0.697	0.597	1.000	1.000	0.004
10	0.674	0.674	0.690	1.000	1.000	0.000
12	0.667	0.667	0.946	1.000	1.000	0.000
14	0.665	0.665	0.970	1.000	1.000	0.000
16	0.665	0.665	0.982	1.000	1.000	0.000
18	0.670	0.670	0.988	1.000	1.000	0.000
20	0.677	0.677	0.756	1.000	1.000	0.000

Tabla (5.19). Cociente de estimación para los diferentes métodos aplicados a las matrices VAN+QR.

El método DIV-MOD, de acuerdo a los anteriores resultados, proporciona estimadores bastante cercanos al valor real y solo comparables con el método de optimización convexa y en menor grado con el retrospectivo.

En las tablas siguientes se presentan algunas comparaciones entre los diferentes métodos de estimación, incluyendo el método DIV-VNC.

Los resultados del método DIV-VNC y DIV-MOD son muy parecidos para los datos de prueba SLT+LU y DXP+LU cuando $K_2(A)$ es igual a 10^3. En los otros casos, los resultados son similares.

Método	n = 5	10	20	30	40
DECOMP	0.994	0.998	0.974	0.980	0.973
LINPACK	0.994	0.989	0.975	0.980	0.974
RETRO	1.0	1.0	1.0	1.0	1.0
DIV-VNC	0.999	0.999	1.0	1.0	1.0
OPT-CNV	1.0	1.0	1.0	1.0	1.0
DIV-MOD	1.0	1.0	1.0	1.0	1.0

Tabla (5.20). Comparación de los diferentes estimadores con las matrices de prueba SLT+LU con $K_2(A) = 10^3$.

Método	n = 5	10	20	30	40
DECOMP	0.800	0.592	0.456	0.390	0.441
LINPACK	0.798	0.600	0.515	0.469	0.503
RETRO	0.962	0.970	0.946	0.921	0.946
DIV-VNC	0.989	0.949	0.918	0.814	0.926
OPT-CNV	1.0	0.976	0.961	0.952	0.987
DIV-MOD	1.0	1.0	0.950	0.934	0.946

Tabla (5.21) Comparación de los diferentes estimadores con las matrices de prueba DXP+LU con $K_2(A) = 10^3$.

También fueron comparados los métodos DIV-VNC y DIV-MOD, con las matrices RND+LU y los resultados se presentan en la tabla (5.22). De acuerdo con las dos primeras entradas de la tabla, el porcentaje de los estimadores obtenidos, por medio del método propuesto, DIV-MOD, que difieren a lo más en un 20% del exacto, fué del 90%, mientras que para el método DIV-VNC fué solamante un 10%.

$K^{-}(A)/K(A)$		DIV-VNC	DIV-MOD
>	≤		
0.9	1.0	6.0%	85.2%
0.8	0.9	4.0%	5.6%
0.7	0.8	4.0%	3.6%
0.6	0.7	10.8%	2.0%
0.5	0.6	9.6%	2.0%
0.4	0.5	9.2%	1.2%
0.3	0.4	14.0%	0.8%
0.0	0.3	43.2%	0.0%

Tabla (5.22). Comparación de los métodos DIV-VNC y DIV-MOD para la matrices de prueba RND+LU.

La tabla (5.22) muestra que el método DIV-VNC puede proporcionar estimadores muy alejados del valor exacto, inclusive menores a los de LINPACK, RETRO o DECOMP, mientras que los obtenidos por el método propuesto, DIV-MOD, son mejores que cualquiera de los métodos heuristicos mencionados anteriormente; Además las tablas (5.20), (5.21) y otros experimentos numéricos no reportados aquí, sugieren que los resultados de nuestro método sólo pueden ser igualados por el método via optimización convexa.

5.4 Contraejemplos.

Los ejemplos que se presentan en esta sección reafirman el hecho de que ninguno de los métodos existentes para estimar el número de condición de una matriz es el mejor. Para cada método es posible construir un ejemplo donde el resultado difiera tanto como se quiera del valor exacto del número de condición. Un contraejemplo para un método no necesariamente lo es para otro y en el mejor de los casos puede servir para varios métodos basados en el mismo resultado.

5.4.1 Métodos tipo Linpack. Estos estimadores son, sin duda, los más estudiados y usados en los códigos de programación para resolver el problema Ax = b. En el trabajo de Cline y R. K. Rew [CI2] se presentan cuatro ejemplos y la estimación lograda por los métodos DECOMP, LINPACK y LINPACKM[1]. El próposito de esta sección es mostrar como se comportan estos ejemplos con el método DIV-VNC. En la tabla (5.23) se resume el análisis de estos ejemplos.

Ejemplo I):

$$A = \begin{bmatrix} 1 & 0 & k & -k \\ 0 & 1 & -k & k \\ 1 & -1 & 2k+1 & -2k \\ -1 & 1 & -2k & 2k+1 \end{bmatrix}$$

$$L = \begin{bmatrix} 1 & & & \\ 0 & 1 & & \\ 1 & -1 & 1 & \\ -1 & 1 & 0 & 1 \end{bmatrix} \qquad U = \begin{bmatrix} 1 & 0 & k & -k \\ & 1 & -k & k \\ & & 1 & 0 \\ & & & 1 \end{bmatrix}$$

El primer paso en estos métodos es resolver el sistema $A^t x = b$; si la matriz ya se encuentra factorizada en la forma A = LU, es equivalente a resolver los sistemas triangulares:

---

[1] Este método no fué considerado en la comparación pero presenta caracteristicas similares a las de LINPACK.

$$U^t z = b$$

y

$$L^t x = z.$$

Al resolver el primer sistema transpuesto, se elige el vector b de manera que $\|z\|_1$ se máxima. De acuerdo al método DIV-VNC el procedimiento a seguir es:

Paso 1) Resolver los sistemas: $ly_1 = 1$, $ly_2 = 1$, $ly_3 = 1$, $ly_4 = 1$.

paso 2) Elegir $\lambda \in (0, 1)$ para cada uno de los sistemas (2x2) triangulares inferiores, que aparecen en la diagonal de U^t, de manera que $\|z\|_1$ sea máxima.

$$U^t = \begin{bmatrix} \begin{bmatrix} 1 & \\ 0 & 1 \end{bmatrix} & \\ & \begin{bmatrix} k & -k \\ -k & k \end{bmatrix} \begin{bmatrix} 1 & \\ 0 & 1 \end{bmatrix} \end{bmatrix} \begin{bmatrix} z_1 \\ z_2 \\ \hline z_1 \\ z_2 \end{bmatrix} = \begin{bmatrix} \lambda \\ (1-\lambda) \\ \hline \lambda \\ (1-\lambda) \end{bmatrix}$$

Como ambos sistemas son iguales, basta con analizar uno de ellos.

$$\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} z_1 \\ z_2 \end{bmatrix} = \begin{bmatrix} \lambda \\ 1-\lambda \end{bmatrix}$$

Si $\lambda = 0$, $\|z\|_1 = \|y_2\| = 1$.

Si $\lambda = 1$, $\|z\|_1 = \|y_1\| + \|0\|_1 = 1$.

Como la norma de z es igual en ambos casos, el método elige $\lambda = 1$. De esto resulta que $z = [1, 0]$. Hacer $y = z$.

Paso 3) Elegir $\lambda \in (0, 1)$ para maximizar $\|z\|_1$ en el sistema:

$$\begin{bmatrix} T_{11} & \\ T_{21} & T_{22} \end{bmatrix} \begin{bmatrix} z_1 \\ z_2 \end{bmatrix} = \begin{bmatrix} \lambda d_1 \\ (1-\lambda) d_2 \end{bmatrix}$$

Con

$$T_{11} = T_{22} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad T_{21} = \begin{bmatrix} k & -k \\ -k & k \end{bmatrix}, \quad d_1 = d_2 = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

Si $\lambda = 0$ $\|z\|_1 = \|y_2\|_1 = 1$.

Si $\lambda = 1$ $\|z\|_1 = \|y_1\|_1 + \|T_{22}^{-1} T_{21} y_1\|_1 = 1 + 2k$.

Si $k > 0$ el vector b resultante es $[1, 0, 0, 0]^t$ y $z = [1, 0, -k, k]^t$.

paso 4) Resolver el sistema el sistema triangular $L^t x = z$; x resulta igual a $[2k+1, -2k, -k, k]$ y finalmente

paso 5) Resolver el sistema $Aw = x$ para obtener el estimador de la norma de la inversa; $\|A^{-1}\|_1 \approx \|w\|_1 / \|x\|_1$. $K_1^-(A) = 20k^2 + 18k + 3$

Ejemplo II):

$$B = \begin{bmatrix} 1 & -1 & -2k & 0 \\ 0 & 1 & k & -k \\ 0 & 1 & k+1 & -(k+1) \\ 0 & 0 & 0 & k \end{bmatrix}$$

$$L = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad U = \begin{bmatrix} 1 & -1 & -2k & 0 \\ 0 & 1 & k & -k \\ 0 & 0 & 1 & -1 \\ 0 & 0 & 0 & k \end{bmatrix}$$

$d^t = [1, 0, 0, 0]$, es el vector obtenido al aplicar el método DIV-VNC a la matriz U^t. Una vez resueltos los sistemas

$$U^t y = d \quad \Rightarrow \quad y^t = [1, 1, k, 2],$$
$$L^t x = y \quad \Rightarrow \quad x^t = [1, 1-k, k, 2],$$
$$B\,w = x \quad \Rightarrow \quad w = [2k^2 - 2k + 6,\ 1 - 2k^2,\ 2k + \tfrac{2}{k} - 1,\ \tfrac{2}{k}].$$

se obtiene la estimación de la norma de la inversa de la matriz B.

$$\|B^{-1}\| \approx \|w\| / \|x\| = \frac{4k^2 + \frac{4}{k} + 6}{2k + 2}.$$

Finalmente el estimador de número de condición resulta:

$$K_1(B) = \frac{2(4k+1)(k^2+k+1)}{k(k+1)}$$

Ejemplo III).

$$C = \begin{bmatrix} 1 & 1-2k^{-2} & -2 \\ 0 & k^{-1} & -k^{-1} \\ 0 & 0 & 1 \end{bmatrix}, \text{ con } k \geq 3.$$

$$C^{-1} = \begin{bmatrix} 1 & -(k^2-2)k^{-1} & (k^2+2)k^{-2} \\ 0 & k & 1 \\ 0 & 0 & 1 \end{bmatrix}.$$

El estimador de $\|C^{-1}\|$ obtenido $2k - \frac{2}{k}$ y por lo tanto de $K_1^{\sim}(C) = \|z\| \|C\|$ es igual a

$$(3 + \frac{1}{k})(2k - \frac{2}{k}),$$

que coincide con el valor exacto del número de condición.

Ejemplo IV).

$$D = \begin{bmatrix} 1 & & & & 0 \\ -1 & 1 & & & \\ -1 & -1 & 1 & & \\ \vdots & \vdots & \ddots & \ddots & \\ -1 & -1 & \cdots & -1 & -1 \end{bmatrix}$$

Siguiendo el mismo procedimiento se obtiene que $K_1^{\sim}(D) = K_1(D)$.

	MATRIZ DE PRUEBA			
	A	B	C	D
K_1	$24k^2+22k+3$	$8k^2+6k+1$	$6k+2+O(k^{-1})$	$n2^{n-1}$
$K_1^{\sim}$	$6k+1$	$\frac{28}{5}k+O(1)$	$6k-4+O(K^{-1})$	n
$K_2^{\sim}$	$20K^2+10k+O(1)$	$\frac{28}{5}k+O(1)$	$\frac{5}{2}+O(k^{-1})$	n
$K_3^{\sim}$	$20K^2+18k+O(1)$	$8k^2-6k+14+O(k^{-1})$	Exacto	Exacto
K_1= Exacto, $K_1^{\sim}$= DECOMP, $K_2^{\sim}$= LINPACK y $K_3^{\sim}$= DIV-VNC				

Tabla (5.23). Comparación de los diferentes métodos de estimación del número de condición de las matrices de prueba.

5.4.2 Matrices de comparación.

Para este tipo de métodos no es necesario construir ejemplos, donde estos métodos fallen, pues un gran número de las matrices de prueba, empleadas en la generación de las tablas de la sección anterior, sirven para este propósito. En la tabla (5.24) se presentan los estimadores obtenidos por medio de las matrices de comparación, aplicados a la parte triangular superior (matriz U de LU), de las matrices de la sección anterior.

	MÉTODO			
MATRIZ	EXACTO	M(T)	W(T)	Z(T)
A	$(2k+1)^2$	Exacto	Exacto	$(3k+1)(k+1)^3$
B	$6k^2+5k+1$	$12k^2+7k+O(1)$	$8k^3+14k^2+7k+1$	$(2k+1)(6k+1)^3$
C	$6k+2+O(k^{-1})$	Exacto	$9k+3$	$27k+9$
D	$n2^{n-1}$	Exacto	Exacto	Exacto

Tabla (5.24). Comparación de los estimadores vía matrices de comparación de las matrices U(A), U(B), U(C) y U(D). U(X) es la matriz triangular superior de la factorización LU de X.

El número de condición de la matriz D en todos los métodos es igual al exacto. Esto se debe a que la matriz D es una matriz M. La matriz U(A) y U(B) pueden servir también de contraejemplo para estos métodos.

5.4.3 Optimización convexa.

Higham en [Hi3] proporciona varios contraejemplos para este método, el primero de ellos son las matrices de Pei, [Gri].

$$A = \alpha I + ee^t, \quad \alpha > 0.$$

La inversa de estas matrices y su norma estan dadas en forma explicita por las relaciones siguientes:

$$A^{-1} = \alpha^{-1}I - \frac{1}{\alpha(\alpha+n)}ee^t, \qquad \|A^{-1}\|_1 = \frac{\alpha + 2(n-1)}{\alpha(\alpha+n)}$$

Con esta matriz, el algoritmo termina en el primer ciclo, alcanzando un máximo local que puede ser arbitrariamente pequeño. En el primer paso del algoritmo se tiene

1) Elegir x con $\|x\|_1 = 1$

Repeat

2) Resolver $Ay = x$

3) $\zeta = \text{sing}(y)$

4) Resolver $A^t z = \zeta$

5) Si $\|z\|_\infty \le z^t x$ entonces termina con $\gamma = \|y\|_1$

6) $x = e_j$, donde $|z_j| = \|z\|_\infty$

Until false

x puede ser elegida de varias maneras, por ejemplo, $x = n^{-1}e$. Del paso 2) se obtiene que $y = A^{-1}(n^{-1}e) = n^{-1}(\alpha + n)^{-1}e$, y así $\zeta = e$. El subgradiente $z = A^{-1}e = (\alpha + n)^{-1}e$, se alcanza en el paso 4). Así el algoritmo termina en el paso 5). El estimador de la norma de la inversa es $\gamma = \|y\|_1 = (\alpha + n)^{-1}$ y el cociente de estimación resulta

$$\frac{\gamma}{\|A^{-1}\|_1} = \frac{1}{\alpha + 2(n-1)} \longrightarrow 0, \text{ cuando } \alpha \longrightarrow 0 \quad y(o) \quad n \to \infty.$$

El segundo ejemplo es la matriz bidiagonal:

$$B_n = \begin{bmatrix} 1 & 1 & & \\ & 1 & 1 & \\ & & \ddots & \ddots \\ & & & 1 \\ & & & 1 \end{bmatrix} \qquad B_n^{-1} = \begin{bmatrix} 1 & -1 & 1 & \cdots & (-1)^{n+1} \\ & 1 & -1 & \cdots & \vdots \\ & & \ddots & \ddots & \vdots \\ & & & \ddots & -1 \\ & & & & 1 \end{bmatrix}$$

Si n es impar, la primera iteración se termina con $y = n^{-1}(1, 0, 1, \ldots)$; e $y = n^{-1}(0, 1, 0, \ldots)$ en otro caso. Así $\zeta = e$ y $z = B_n^{-t}e = [1, 0, 1, \ldots]^t$. Como la prueba de convergencia (Paso 5) no se satisface se elige el índice j, más pequeño, tal que $|z_j| = \|z\|_\infty$. Entonces $x = e_1$ para la siguiente iteración.

En la segunda iteración resulta $y = e_1$, y los vectores ζ y z, son iguales a los del primer paso; así el algoritmo termina con $\gamma = \|y\|_1 = 1$ y el cociente de estimación resulta:

$$\frac{\gamma}{\|B_n^{-1}\|} = \frac{1}{n}$$

La subestimación puede ser significativa para valores grandes de n.

5.4.4. Método DIV-MOD

La siguiente matriz es un contraejemplo para el método DIV-MOD

$$A = \begin{bmatrix} 1/3 & 0 & 0 & -k \\ & 1 & 0 & -k \\ & & 1/2 & 0 \\ & & & 1 \end{bmatrix} \quad \text{con } k \geq 1,$$

tiene como inversa la matriz

$$A^{-1} = \begin{bmatrix} 3 & 0 & 0 & 3k \\ & 1 & 0 & k \\ & & 2 & 0 \\ & & & 1 \end{bmatrix}, \tag{5.2}$$

con $\|A\|_1 = 2k + 1$, $\|A^{-1}\|_1 = 4k + 1$. Por lo tanto $K(A) = 8k^2 + 6k + 1$. Para los dos matrices triangulares 2x2, el método DIV-MOD selecciona la columa 1 y 3 como las de mayor norma. De acuerdo con (5.2) la norma de estas columnas es 3 y 1 respectivamente. Sin embargo, la columna de A^{-1} con mayor norma es la columna 4, la cual queda excluida. En el paso siguiente se combinan las dos submatrices anteriores para formar parte de la matriz A. La columna de A^{-1} con mayor norma es elegida de entre las columnas (1 ó 3) de las submatrices que la integran. La norma estimada de $\|A^{-1}\|$ resulta igual a 3 y el $K(A) = 6k + 3$, el cual es inferior al valor real.

CAPITULO 6

CONCLUSIONES

La mayoría de los métodos de estimación del número de condición de matrices triangulares pueden emplearse para estimar otro tipo de matrices. Para matrices pequeñas y densas puede emplearse el procedimiento siguiente:

a) Calcular $\|A\|$,
b) Aplicar la estrategia al sistema $U^t w = d$,
c) Resolver $L^t y = w$,
d) Resolver $Lv = y$,
e) Resolver $Ux = v$,
f) $K(A) = \|z\| \, \|A\| \, / \, \|y\|$.

En el paso b) puede aplicarse cualquiera de las heurísticas del capítulo 4.

Para el método de optimización convexa sólo se requiere escribir un subprograma que resuelva el sistema $Ax = b$ y el transpuesto.

Una de las aportaciones de este trabajo es el método DIV-MOD, que fué probado exhaustivamente con las matrices de prueba mencionadas en el capítulo 5. El método DIV-MOD, en la mayoría de las matrices de prueba, proporciona mejores resultados que cualquier otro tipo de estimadores heurísticos y sólo comparables con los obtenidos por el método vía optimización convexa.

Para probar los métodos de estimación empleamos las matrices triangulares superiores, obtenidas de la factorización LU o QR, de matrices de prueba reportadas por otros autores; en particular para las matrices: SLT, DXP y RND.

Para la factorización LU y QR se encontró que el cociente

$$K^{-}(A) \; / \; K(A),$$

es sistemáticamente más grande para las matrices triangulares, U y R, que para la matriz inicial.

Existen varios criterios de comparación para los métodos de estimación

del número de condición. El primero de ellos es el número de operaciones que se necesitan. En la tabla 6.1 se presenta el orden de operaciones requeridas por algunos de los métodos del capítulo 4.

Método	Norma	Costo	Tipo de Cota
M(T)	∞	$n^2/2$ flops	Superior
W(T)	∞	$n^2/2$ flops	Superior
DECOMP	1	n^2 flops	Inferior
LINPACK	1	$5n^2/2$ flops	Inferior
RETRO	2	$5n^2/2$ flops	Inferior
OPT-CON	1	$2n^2$ o $3n^2$, en la practica	Inferior
DIV-MOD	1	$3n^2+3n$, para $n=2^k$	Inferior

Tabla 6.1 Costo de algunos estimadores del número de condición.

El costo de los métodos es pequeño comparado con el número de operaciones requerido para factorizar una matriz. Sin embargo para matrices grandes, poco densas, con o sin estructura el costo puede compararse con la factorización.

Cuando la matriz tiene estructura, existen métodos de estimación más eficientes, por ejemplo, para matrices tridiagonales o tipo Hessemberg, puede emplearse algunos de los métodos mencionados en el capítulo 4. Por ejemplo, la norma de la inversa de una matriz de Hessemberg puede obtenerse en 3n flops y en 17n para una matriz tridiagonal. Cuando la matriz es tridiagonal y simétrica puede reducirse a 11n flops.

Otro criterio de comparación es la aproximación que proporcionan al valor exacto del número de condición. Del capítulo 5 se concluye, primero, que no existe el mejor estimador, es decir, siempre es posible construir un ejemplo donde el estimador falle tanto como se quiera y segundo, que para las matrices generadas para este trabajo los estimadores DIV-MOD y OPT-CONV son los mejores. El último de estos estimadores tiene una ventaja adicional sobre los heurísticos; puede emplearse con cualquier tipo de matriz, sólo basta tener una rutina para resolver el sistema $A\ x = b$ y $A^t\ x = b$. Esto hace independiente al método de la representación de la matriz y de la

factorización empleada.

El problema de estimación del número de condición es muy interesante y amplio. El tabajo puede continuarse por alguno de los siguientes caminos.

a) Construcción de métodos de estimación del número de condición para otro tipo de matrices. Por ejemplo, para matrices poco densas, definidas positivas, Toeplitz, etc).

b) Desarrollo de métodos de estimación para el número de condición en la norma 2. Esta norma es importante por su relación con los valores propios de la matriz.

c) Tratamiento de problemas mal condicionados. Una vez identificado el mal condicionamiento del problema, es necesario resolverlo por algún método. Los métodos más conocidos pueden resultar inestables en ciertas situaciones. Un mejor conocimiento de la estabilidad de los métodos permitirá hacer una mejor elección y tener cotas de error más precisas.

d) Es innegable que el número de condición tradicional puede sobreestimar la sensitividad del la solución del problema $Ax = b$ bajo pequeñas perturbaciones de los datos. Para los problemas comentados en los puntos a)-c) pueden emplearse otras definiciones del número de condición.

e) Estimación del número de condición de una matriz, cuando cambia en una columna. Este tipo de estimadores permitirían decidir el momento de realizar el proceso de actualización de una base en programación lineal.

APENDICE A

A.1 Matriz de Vandermonde y confluentes.

A.2 Polinomios ortogonales.

A.1. Matriz de Vandermonde y confluentes.

En este apéndice se presentan algunos resultados de la matriz de Vandermonde y sus confluentes, necesarios en algunas secciones del trabajo. Algunos de los resultados son unicamente válidos cuando los nodos son reales y positivos o bien están distribuidos simétricamente respecto al origen.

A.1) Definición de la matriz de Vandermonde V_n y confluentes, [Gal].

La matriz de Vandermonde de orden n es una matriz de la forma:

$$V = V(x_1, x_2, \ldots, x_n) = \begin{bmatrix} 1 & 1 & \ldots & 1 \\ x_1 & x_2 & \ldots & x_n \\ \cdots & \cdots & \cdots & \cdots \\ x_1^{n-1} & x_2^{n-1} & \ldots & x_n^{n-1} \end{bmatrix},$$

donde $x = \{x_1, x_2, \ldots, x_n\}^t$ y $n > 1$. Los x_k son números reales o complejos.

La operación de confluencia de la columna l en la columna k consiste en:

a) reemplazar en la l-ésima columna, x_l por $x_k + \varepsilon$,
b) restar la l-ésima columna la k-ésima columna,
c) dividir la nueva columna l por ε y finalmente
d) tomar el límite cuando $\varepsilon \to 0$.

En otras palabras, la confluencia de la columna l en la columna k, consiste en reemplazar la l-ésima columna por la derivada de la k-ésima columna. La matriz resultante es denotada por $U_{n,kl}$.

$$U_{n,kl}(x) = \begin{pmatrix} 1 & \dots & 1 & \overset{\text{Columna } l}{\overset{\downarrow}{0}} & 1 & \dots & 1 \\ x_1 & \dots & x_{l-1} & 1 & x_{l+1} & \dots & x_n \\ x_1^2 & \dots & x_{l-1}^2 & 2x_l & x_{l+1}^2 & \dots & x_n^2 \\ \vdots & & \vdots & \vdots & \vdots & & \vdots \\ x_1^{n-1} & \dots & x_{l-1}^{n-1} & (n-1)x_l^{n-2} & x_{l+1}^{n-1} & \dots & x_n^{n-1} \end{pmatrix}$$

Una matriz obtenida a partir de una o más confluencias de columnas, se dice una matriz confluente de Vandermonde. Por ejemplo, la matriz confluente de Vandermonde de orden 2n es obtenida por confluir las columnas n + l en l, para l = 1, 2,..n. Resultando

$$U_{2n}(x) = \begin{pmatrix} 1 & \dots & 1 & 0 & \dots & 0 \\ x_1 & \dots & x_n & 1 & \dots & 1 \\ x_1^2 & \dots & x_n^2 & 2x_1 & \dots & 2x_n \\ \vdots & & \vdots & \vdots & & \vdots \\ x_1^{2n-1} & \dots & x_n^{2n-1} & (2n-1)x_1^{2n-2} & \dots & (2n-1)x_n^{2n-2} \end{pmatrix}$$

A.2) Norma de V_n y de su inversa. [Ga1], [Ga2] y [Ga8].

Norma de la matriz, $\|V_n\|_\infty$.[Ga1].

$$\|V_n(x)\|_\infty = \max(n, \sum_{i=1}^{n} |x_i|^{n-1})$$

Norma de la inversa, $V_n^{-1}(x)$. [Ga2].

$$\|V_n^{-1}\|_\infty \le \max_{1 \le l \le n} \prod_{\substack{j=1 \\ j \ne l}}^{n} \frac{1 + |x_j|}{|x_j - x_l|},$$

Si $x_j = |x_j| e^{i\theta_j}$, (j = 1, 2, ... n), la relación anterior es una igualdad. En otras palabras, la igualdad se cumple cuando los puntos están sobre un mismo rayo que parte del origen.

A.3) Norma de la matriz inversa cuando los nodos tienen cierta distribución.

Cuando los nodos son reales y $x \neq 0$, existe una expresión exacta para la norma de $V_n^{-1}(x)$, dada por

$$\|V_n^{-1}(x)\|_\infty = \max_{1\le i\le n} \prod_{\substack{j=1\\ j\neq i}}^{n} \frac{1+|x_j|}{|x_j - x_i|} = \max_{1\le i\le n} \left\{ \frac{|p_n(-1)|}{(1+x_i)|p_n'(x_i)|} \right\},$$

donde $p_n(x)$ es el polinomio mónico con ceros en los nodos $x_1, x_2, \ldots, x_n$.

Cuando los nodos son reales y se distibuyen simétricamente alrededor del cero, es decir, $x_i + x_{n-i+1} = 0$, para $i = 1, 2, \ldots, n$, entonces la norma de la inversa está dada explicitamente por

$$\|V_n^{-1}(x)\|_\infty = \max_{1\le i\le \left[\frac{n+1}{2}\right]} \left\{ \frac{|p_n(i)|}{\frac{1+x_i^2}{1+x_i}\,|p_n'(x_i)|} \right\}, \qquad i=\sqrt{-1}$$

A.4) Propiedades de la norma de la inversa.

Simetría de la norma de la inversa. La función de $f:\mathbb{R}^n \to \mathbb{R}$, definida por $f(x_1, x_2, \ldots, x_n) = \|V_n^{-1}(x_1, x_2, \ldots x_n)\|_\infty$ es una función simétrica en las variables $x_1, x_2, \ldots, x_n$.

Cambios de escala. Sea $\omega \neq 0$ un número complejo arbitrario y $V(\omega) = V(wx_1, wx_2, \ldots, wx_n)$ Entonces $\|V_n^{-1}(w)\|_\infty$ depende solamente de $|w|$ y es una función estrictamente decreciente de w.

En el caso real, un cota del decrecimiento relativo de $\|V_n^{-1}(w)\|_\infty$ respecto a los nodos iniciales queda establecido en las dos relaciones siguientes:

Si $x_k \neq 0$ para $k = 1, 2, \ldots, n$, entonces:

$$\frac{w}{w+1}\left|\frac{P_n(-1/w)}{P_n(-1)}\right| < \frac{\|V_n^{-1}(w)\|_\infty}{\|V_n^{-1}(1)\|_\infty} < (w+1)\left|\frac{P_n(-1/w)}{P_n(-1)}\right|,$$

donde $P_n(x) = (x - x_1)(x - x_2)\ldots(x - x_n)$.

Si n es par y $x_k + x_{n+1-k} = 0$, para $k = 1, 2, \ldots, n$, entonces

$$\frac{2(\sqrt{2}-1)w}{w+1}\left|\frac{P_n(1/w)}{P_n(1)}\right| < \frac{\|V_n^{-1}(w)\|_\infty}{\|V_n^{-1}(1)\|_\infty} < \frac{w+1}{2(\sqrt{2}-1)}\left|\frac{P_n(1/w)}{P_n(1)}\right|,$$

donde $P_n(x) = (x - x_1)(x - x_2)\ldots(x - x_n)$.

Es claro que una permutación de las variables $x_1, x_2, \ldots, x_n$ no altera el valor de $\|V_n(x)\|_\infty$, y de $\|V_n^{-1}(x)\|$ cuando la inversa de $V_n(x)$ existe. Esto permite ordenar la variables en algún orden. Si los nodos son reales se supone un orden decreciente:

$$x_1 > x_2 > \ldots > x_n$$

A.5) Número de condición de la matriz de Vandermonde $K_\infty(V_n)$.

A partir de las relaciones de A.2), pueden obtenerse cotas del número de condición de la matriz de Vandermonde. Cuando los nodos son reales y positivos o bien simétricos puede emplearse las relaciones A.3), para obtener una mejor cota.

Otro problema de interés relacionado con la matriz de Vandermonde es: determinar el conjunto de nodos que da un número de condición mínimo. El valor mínimo del número de condición, denotado por $K_{n,\infty} = \inf K_\infty(V_n(x))$, para los nodos no negativos está acotado inferiormente por 2^{n-1} y en el caso simétrico por $2^{n/2}$, en ambos casos, para $n \geq 2$, Gautschi en [Ga7] y [Gal2] proporciona una mejor cota para estos casos. Sea $K_{n,p} = \inf K_p(V_n(x))$, $(p=\infty)$ y x real y no negativo, entonces para $n \geq 2$

$$K_{n,\infty} \geq (n-1)\left\{1 + \left(1 - \frac{1}{n}\right)^{-1/(n-1)}\right\}^{n-1},$$

en particular,

$$K_{n,\infty} > (n-1)2^{n-1}, \text{ para } n \geq 2.$$

Sea $K_{n,\infty}$ como en el resultado anterior, y si además, x es real y simétrico, entonces para $n \geq 4$ se tiene:

$$K_{n,\infty} > \begin{cases} (n-2)\left\{1 + \left(1 - \frac{2}{n}\right)^{-2(n-1)}\right\}^{(n-2)/2}, & n \text{ par} \\ (n-3)\left\{1 + \left(1 - \frac{3}{n}\right)^{-2(n-1)}\right\}^{(n-3)/2}, & n \text{ impar} \end{cases}$$

En particular para $n \geq 4$.

$$K_{n,\infty} = \begin{cases} (n-2)2^{(n-2)/2}, & n \text{ par} \\ (n-3)2^{(n-3)/2}, & n \text{ impar.} \end{cases}$$

No existe una cota superior para el número de condición. Gautschi en [Ga1], demuestra que dado un conjunto de nodos $x_1 > x_2 \ldots x_n > 0$ entonces $\lim_{\lambda \to \infty} K_\infty(V_n(\lambda x)) \to \infty$.

A.6) Norma de la inversa de matrices confluentes de Vandermonde, $U_{n,kl}$ [Ga1], [Ga2] y [Ga3].

Norma de la inversa de una confluente. Sean $x_\nu \neq x_\mu$ para $\nu \neq \mu$ ($\nu, \mu = 1, 2, \ldots, l-1, l+1, \ldots, n$), entonces

$$\|U^{-1}_{n,kl}\|_\infty \leq \max_{\substack{1 \leq \lambda \leq n \\ \lambda \neq l}} a_\lambda \prod_{\substack{\nu=1 \\ \nu \neq \lambda, l}}^{n} \frac{1 + |x_\nu|}{|x_\nu - x_\lambda|},$$

donde:

$$a_\lambda = \begin{cases} \dfrac{1 + |x_\lambda|}{|x_k - x_\lambda|} & (\lambda \neq k, l) \\ \max\left[1 + |x_k|,\ 1 + (1 + |x_k|) \displaystyle\sum_{\substack{\nu=1 \\ \nu \neq k, l}}^{n} \frac{1}{|x_\nu - x_k|}\right] & (\lambda = k) \end{cases}$$

Norma de la inversa de la matrix confluente de orden 2n (U_{2n}).

Sean $x_\nu \neq x_\mu$ para $\nu \neq \mu$ ($\nu, \mu = 1, 2, \ldots, n$), entonces:

$$\|U^{-1}_{2n}\|_\infty \leq \max_{1 \leq \lambda \leq n} b_\lambda \left[\prod_{\substack{\nu=1 \\ \nu \neq \lambda}}^{n} \frac{1 + |x_\nu|}{|x_\nu - x_\lambda|}\right]^2,$$

donde:

$$b_\lambda = \max \left[1 + |x_\lambda|,\ 1 + 2(1 + |x_\lambda|) \sum_{\substack{\nu=1 \\ \nu \neq \lambda}}^{n} \frac{1}{|x_\nu - x_\lambda|} \right]$$

A.2) Polinomios ortogonales.

Los polinomios ortogonales tienen propiedades interesantes y juegan un papel fundamental en la teoría de aproximación. Algunas definiciones y propiedades de los polinomios ortogonales, necesarias en el capítulo 3, son dadas a continuación. Un resumen excelente de este tipo de polinomios aparecen en [Abl].

A.7) Definiciones y notación

Sea $\{f_0, f_1, \ldots f_n, \ldots\}$, una familia de polinomios ortogonales sobre el intervalo [a,b], con función de peso $w(x) \geq 0$,

$$h_n = \int_a^b w(x)\, f_n^2(x)\, dx, \quad n = 0, 1, 2, \ldots$$

$$\mu_0 = \int_a^b w(x)\, dx.$$

Se denotará por k_n y k'_n los coeficientes, de la potencia n y n-1 respectivamente, del polinomio f_n, es decir,

$$f_n(x) = k_n x^n + k'_n x^{n-1} + \ldots \quad , \quad n = 0, 1, \ldots$$

A.8) Relación de recurrencia de tres términos.

Dada una familia de polinomios ortogonales $\{f_0, f_1, \ldots\}$ satisface la siguiente relación de recurrencia

$$f_{n+1} = (a_n + x\ b_n) f_n - c_n\ f_{n-1}, \quad n \geq 1,$$

donde: $b_n = k_{n+1} / k_n$,

$$a_n = b_n \left[\frac{k'_{n+1}}{k_{n+1}} - \frac{k'_n}{k_n} \right],$$

$$c_n = \frac{k_{n+1}\, k_{n-1}\, h_n}{k_n^2\, h_{n-1}} .$$

Algunas de las familias clásicas de polinomios ortogonales son:

A.9) Polinomios de Chebyshev.

$$w(x) = (1 - x^2)^{-1/2} \quad \text{en} \quad [-1, 1].$$

$$T_n(x) = \cos(n \cos^{-1}(x)), \qquad n = 0, 1, \ldots,$$

$$h_n = \begin{cases} \pi & \text{si } n = 0 \\ \pi/2 & \text{si } n \neq 0 , \end{cases}$$

$$\mu_0 = \pi,$$

$$T_0(x) = 1, \qquad T_1(x) = x,$$

$$T_{n+1}(x) = 2 x T_n(x) - T_{n-1}(x).$$

Estos polinomios han sido extensivamente estudiados, debido a su importancia en la teoría de aproximación. El libro de Rivlin [Ri4] está dedicado a estudiar las propiedades de estos polinomios y [Fo4] a sus aplicaciones al análisis numérico.

A continuación se enuncian algunas propiedades adicionales de estos polinomios.

(P1) Norma acotada.

$$|T_n(x)| \leq 1 , \quad \text{para} \quad -1 \leq x \leq 1.$$

(P2) Optimalidad.

Una de las propiedades más importantes de los polinomios de Chebyshev es la propiedad de extremalidad del polinomio de Chebyshev normalizado, $\tilde{T}_n$ (su coeficiente principal es 1), el cual tiene *norma*[1] mínima en [-1, 1] entre todos los polinomios *mónicos* de grado menor o igual que n, es decir, si

---

1 La norma empleada es la norma infinito que para una función g, continua en el intervalo I = [a, b], está definida como $\|g\| = \max_{a \leq x \leq b} |g(x)|$.

$$p(x) = x^n + a_{n-1}x^{n-1} + \ldots + a_0,$$

entonces

$$\|p\| \geq \|T_n^{\sim}\| = \begin{cases} 2^{1-n}, & n > 0 \\ 1, & n = 0 \end{cases}$$

y se obtiene la igualdad cuando $p = T_n^{\sim}$. La demostración de este resultado aparece en [Ri4, pag 56].

(P3) Estimación asintótica de sus coeficientes [Gal0]:

Es conocido en la literatura que:

$$T_n(x/w) = \sum_{k=0}^{(n/2)} c_{nk} x^{n-2k}$$

Donde:

$$c_{nk} = (-1)^k \frac{n(n-k-1)!}{2^k k!(n-2k)!} \left(\frac{2}{w}\right)^{n-2k}, \quad 0 \leq k \leq (n/2)$$

Para t fija, con $0 < t < 1/2$, $k = tn$ y $n \longrightarrow \infty$, se obtiene:

$$c_{nt} \approx \frac{n^{-1/2}}{2\sqrt{2\pi}} \frac{1}{\sqrt{t(1-t)(1-2t)}} \left(\frac{2}{w}\right)^n e^{ng(t)}, \quad n \longrightarrow \infty,$$

donde:

$g(t) = (1-t)\ln(1-t) - t\ln(t) - (1-2t)\ln(1-2t) - 2t\ln(2/w)$

$0 < t < 1/2$

Esta función tiene un máximo en:

$$t_0 = \frac{1}{2}\left(1 - \frac{1}{\sqrt{1+w^2}}\right),$$

Además $g(t_0) = \ln \frac{1-t_0}{1-2t_0} = 1/2\ w(1+w^2)^{-3/4}$. Así una aproximación asintótica para el coeficiente más grande de $T_n(x/w)$ es:

$$\frac{1}{\sqrt{2\pi}} \frac{(1+w^2)^{3/4}}{w} n^{-1/2} \left(\frac{1+\sqrt{1+w^2}}{w}\right)^n \quad ,n \longrightarrow \infty.$$

(P4) Estimación Asintótica de los polinomios de Chebyshev desplazados: $T_n^*(x) = T_n(2x-1)$.

Reemplazando en la relación anterior n por 2n y w por $\sqrt{w}$ y del hecho que $T_n^*(x^2) = T_{2n}(x)$, se obtiene

$$\frac{1}{2\sqrt{\pi}} \; \frac{(1+w)^{3/4}}{\sqrt{w}} n^{-1/2} \left(\frac{2 + w + 2\sqrt{1+w}}{w} \right)^n, \quad n \longrightarrow \infty,$$

una estimación del coeficiente máximo del $T^*(x)$.

A.10) Polinomios de Legendre (Esféricos).

$$w(x) = 1 \text{ en } [-1, 1]$$

$$P_n(x) = \frac{(-1)^n}{2^n n!} \; \frac{d^n}{dx^n} [(1 - x^2)^n], \quad n \geq 1,$$

$$h_n = \frac{2}{2n - 1},$$

$$\mu_0 = 2,$$

$$P_{n+1}(x) = \frac{2n + 1}{n + 1} \, x \, P(x) - \frac{n}{n - 1} P_{n-1}(x).$$

Al igual que los de Chebyshev, están acotados en el intervalo de ortogonalidad.

$$|P_n(x)| \leq 1 \qquad \text{para} \quad -1 \leq x \leq 1$$

APENDICE B

LISTADOS DE PROGRAMAS RELACIONADOS CON POLINOMIOS.

Algunos de los métodos presentados en los capitulos 3 y 4 fueron programados y se presentan en este apéndice y el siguiente. Este apéndice contiene los listados de los programas relacionados con polinomios. Se incluye, además, un subprograma para determinar las propiedades de la aritmética de punto flotante empleada.

Los lenguajes de programación empleados en la programación de los procedimientos fueron Pascal (Turbo Pascal V4.0), C (Turbo C V2.0) y Fortan (Microsoft V3.3). Otros fueron elaborados en los paquetes matemáticos Reduce [Ra1], Derive y Matlab [Ma1].

Los lenguajes tiene caracteristicas numéricas similares cuando emplean el coprocesador Intel-8087 [In1]. Cuando no se cuenta con el procesador, los lenguajes pueden realizar la aritmética con un paquete emulador del 8087, que proporciona una calidad numérica similar, sólo que en un tiempo mayor (De acuerdo a [In1] dos ordenes de magnitud, Ver [In1]). Turbo Pascal proporciona, adicionalmente, una aritmética no tan lenta como la del emulador, pués emplea menos bytes para representar el número, pero con menor calidad numérica.

B.1) **Subprograma MACHAR.** Fue propuesto originalmente por Cody en [Co1], para determinar dinámicamente 13 parámetros relacionados con la aritmética de punto flotante. Los tres primeros parámetros, la base de la aritmética, el número de digitos de la mantisa y si la aritmética trunca o redondea, son determinados con el algoritmo original de M. Malcolm [Ma2]. El subprograma MACHAR fué modificado ligeramente para obtener los resultados adecuados en la microcomputadoras XT y AT. Esta modificación se debe a que la aritmética se realiza en más bits (63) que los empleados para su representación en memoria (48).

```
      SUBROUTINE MACHAR(IBETA, IT, IRND, NGRD, MACHEP, NEGEP, IEXP,
     -     MINEXP, MAXEXP, EPS, EPSNEG, XMIN, XMAX)
```

```
C
      INTEGER I, IBETA, IEXP, IRND, IT, IZ, J, K, MACHEP, MAXEXP
      INTEGER MINEXP, MX, NEGEP, NGRD
      REAL    A, B, BETA, BETAIN, BETAM1, EPS, EPSNEG, ONE, XMAX
      REAL XMIN, Y, Z, ZERO
C
C Ultima revisión - Octubre 22, 1979.
C Autor: W. J. Cody
C Argonne National Laboratory
C
      ONE  = FLOAT(1)
      ZERO = 0.0E0
C-----------------------------------------------------------------------
C DETERMINA IBETA, BETA, A LA MALCOM.
C-----------------------------------------------------------------------
      A = ONE
10    A = A + A
      IF (((A + ONE) - A) - ONE .EQ. ZERO) GO TO 10
      B = ONE
20    B = B + B
      IF ((A + B) - A .EQ. ZERO) GO TO 20
      IBETA = INT((A + B) - A)
      BETA = FLOAT(IBETA)
C-----------------------------------------------------------------------
C DETERMINA IT Y IRND
C-----------------------------------------------------------------------
      IT = 0
      B = ONE
100   IT = IT + 1
      B = B * BETA
      IF (((B + ONE) - B) - ONE .EQ. ZERO) GO TO 100
      IRND = 0
      BETAM1 = BETA - ONE
      IF (( A + BETAM1) - A .NE. ZERO) IRND = 1
C-----------------------------------------------------------------------
C DETERMINA NEGEP Y EPSNEG
C-----------------------------------------------------------------------
      NEGEP = IT + 3
      BETAIN = ONE / BETA
      A = ONE
      DO 200 I = 1, NEGEP
        A = A * BETAIN
200   CONTINUE
      B = A
210   IF ((ONE - A) - ONE .NE. ZERO) GO TO 220
        A = A * BETA
        NEGEP = NEGEP - 1
      GO TO 210
220   NEGEP = -NEGEP
      EPSNEG= A
      IF ((IBETA .EQ. 2) .OR. (IRND .EQ. 0)) GO TO 300
      A = (A * (ONE + A)) / (ONE + ONE)
      IF ((ONE - A) - ONE .NE. ZERO) EPSNEG = A
```

```
C-----------------------------------------------------------------------
C DETERMINA MACHEP Y EPS.
C-----------------------------------------------------------------------
300  MACHEP = -IT - 3
     A = B
310  IF ((ONE + A) - ONE .NE. ZERO) GO TO 320
     A = A * BETA
     MACHEP = MACHEP + 1
     GO TO 310
320  EPS = A
     IF ((IBETA .EQ. 2) .OR. (IRND .EQ. 0)) GO TO 350
     A = (A * (ONE + A)) / (ONE + ONE)
     IF ((ONE + A) - ONE .NE. ZERO) EPS = A
C-----------------------------------------------------------------------
C DETERMINA NGRD
C-----------------------------------------------------------------------
350  NGRD = 0
     IF ((IRND.EQ.0) .AND. ((ONE + EPS) * ONE - ONE).NE.ZERO)NGRD = 1
C-----------------------------------------------------------------------
C DETERMINA IEXP, MINEXP, XMIN
C-----------------------------------------------------------------------
     I = 0
     K = 1
     Z = BETAIN
400  Y = Z
     Z = Y * Y
C-----------------------------------------------------------------------
C VERIFICA POR UNDERFLOW.
C-----------------------------------------------------------------------
     A = Z * ONE
     IF ((A + A .EQ. ZERO) .OR. (ABS(Z) .GT. Y)) GO TO 410
     I = I + 1
     K = K + K
     GO TO 400
410  IF (IBETA .EQ. 10) GO TO 420
     IEXP = I + 1
     MX   = K + K
     GO TO 450
C-----------------------------------------------------------------------
C PARA COMPUTADORAS DECIMALES SOLAMENTE.
C-----------------------------------------------------------------------
420  IEXP = 2
     IZ   = IBETA
430  IF (K .LT. IZ) GO TO 440
     IZ = IZ * IBETA
     IEXP = IEXP + 1
     GO TO 430
440  MX = IZ + IZ - 1
C-----------------------------------------------------------------------
C ENTRA EN UN BUCLE PARA DETERMINAR MINEXP Y XMIN
C TERMINACION DEL BUCLE ES INDICACION DE UN UNDERFLOW.
C-----------------------------------------------------------------------
450  XMIN = Y
```

```
      Y = Y * BETAIN
C-----------------------------------------------------------------
C VERIFICA UNDERFLOW.
C-----------------------------------------------------------------
      A = Y * ONE
      IF (((A + A) .EQ. ZERO) .OR. (ABS(Y) .GE. XMIN)) GO TO 460
      K = K + 1
      GO TO 450
460   MINEXP = -K
C-----------------------------------------------------------------
C DETERMINA MAXEXP Y XMAX
C-----------------------------------------------------------------
      IF ((MX .GT. K + K - 3) .OR. (IBETA .EQ. 10)) GO TO 500
      MX = MX + MX
      IEXP = IEXP + 1
500   MAXEXP = MX + MINEXP
C-----------------------------------------------------------------
C AJUSTE PARA COMPUTADORAS CON BIT IMPLICITO EN LA PARTE
C MAS SIGNIFICATIVA DE LA MANTISA Y PARA COMPUTADORAS CON
C PUNTO DE LA BASE AL EXTREMO DERECHO DE LA MANTISA.
C-----------------------------------------------------------------
      I = MAXEXP + MINEXP
      IF ((IBETA .EQ. 2) .AND. (I .EQ. 0)) MAXEXP = MAXEXP - 1
      IF (I .GT. 20) MAXEXP = MAXEXP - 1
      IF (A .NE. Y ) MAXEXP = MAXEXP - 2
      XMAX = ONE - EPSNEG
      IF (XMAX * ONE .NE. XMAX) XMAX = ONE - BETA * EPSNEG
      A = XMIN
505   IF (A .GE. EPS) GO TO 507
        XMAX =  XMAX / EPS
        A = A * EPS
      GO TO 505
507   XMAX = XMAX  / (BETA * BETA * BETA * (XMIN / A))
      I = MAXEXP + MINEXP + 3
      IF (I .LE. 0) GO TO 520
      DO 510 J = 1, I
        IF (IBETA .EQ. 2) XMAX = XMAX + XMAX
        IF (IBETA. NE. 2) XMAX = XMAX * BETA
510   CONTINUE
520   RETURN
C---------------------- ULTIMA TARJETA DE MACHAR ----------------------
      END
```

B.2) **Evaluación de polinomios por diferentes métodos.** Las representaciones consideradas son la de potencias, potencias desplazada y la de Newton. Para la evaluación de polinomios con alguna de estas representaciones se emplea el método de multiplicaciones anidadas.

```
/*-----------------------------------------------------------------*/
```

```
/*-------------------------------------------------------------------
/* Evaluación de un polinomio con representación de potencias
/*                        P(x) = Σ_{i=0}^{n-1} a_i x^i
/* por medio del método de multiplicaciones anidadas.
/* a[n]     Arreglo de coeficientes del polinomio de grado n - 1
/* n        Número de coeficientes del polinomio.
/* x        Punto a evaluar.
/*-------------------------------------------------------------------*/
double evapolma(a, x, n)
double a[] ;
double x   ;
int    n   ;
{ double s ;

  s = 0.0 ;
  for (s = 1.0 ; n > 0  ;) s = s * x + a[--n] ;
  return(s) ;
}

/*-------------------------------------------------------------------
/*  Evaluación  de  un polinomio con representación de potencias
/*  desplazada por medio del método de multiplicaciones anidadas
/*                        Σ_{i=0}^{n-1} a_i (x - d)^i
/*  a[n]   Arreglo con los coeficientes del polinomio de  grado
/*         n - 1.
/*  d      Desplazamiento.
/*  n      Número de coeficientes del polinomio.
/*  x      Punto a evaluar.
/*-------------------------------------------------------------------*/
double evapolpd(a, d, n, x)
double a[] ;
double d   ;
double x   ;
int    n ;
{ double s ;

  x -= d ;
  for ( s = 1.0 ; n > 0 ; ) s = s * x  + a[--n] ;
 return(s)  ;
}

/*-------------------------------------------------------------------
/* Evaluación de un polinomio con representación de Newton por
/* medio del método de multiplicaciones anidadas.
/*                        Σ_{i=1}^{n-1} a_i Π_{j=1}^{i} (x - d_j)
/* a[n]     Arreglo de coeficientes del polinomio de grado n-1
/* d[n]     Arreglo de desplazamientos.
/* n        Número de coeficientes del polinomio.
/* x        Punto a evaluar.
/*-------------------------------------------------------------------*/
```

```
double evapolnw(a, d, n, x)
double a[] ;
double d[] ;
int    n   ;
double x   ;
{ double s ;

  for (s = 1; n > 0; ) s = s * (x - d[--n]) + a[n] ;
  return(s) ;
}
```

B.3) Generación de polinomios ortogonales. Los métodos programados para obtener un conjunto de polinomios ortogonales, a partir de un conjunto de momentos de potencias, fueron el método de Chebyshev [Whi] y el método de momentos modificados propuesto por Wheeler en [Whi].

```
/*-------------------------------------------------------------
/* Generación de polinomios ortogonales a partir de los
/* 2n primeros momentos. El método fué propuesto originalmente
/* por Chebyshev y permite obtener los coeficientes que apare-
/* cen en la relación de recurrencia de los polinomios ortogo-
/* nales. Este procedimiento puede resultar mal condicionado
/* cuando n es grande.
/* Referencia: John C. Wheeler
/*                Modified moments and Gaussian quadratures
/*               Rocky Mountains,
/*              Journal in Mathematics Vol.4. Num. 2, (1974)
/* powm[2n]  Arreglo de momentos de potencia.
/* ak[n]     Arreglo con los coeficientes a_i que aparecen en la
/*           relación de recurrencia.
/* bk[n]     Arreglo con los coeficientes b_i que aparecen en la
/*           relación de recurrencia.
/* n          2n es el número de momentos inicial.
/*-------------------------------------------------------------*/

cheby(powm, ak, bk, n)
double powm[] ;
double  ak[]  ;
double  bk[]  ;
int     n     ;
{ double za[50], zb[50] ;
  double *pza, *pzb, *ptz;
  int k, l ;

  for (k = 0; k < 2 * n ; k++)
  { za[k] = 0 ;
    zb[k] = powm[k] ;
  }
```

```
    pza  = za ;
    pzb  = zb ;
    ak[0]= 0.0;
    bk[0]= powm[0] ;
    for (k = 1; k < n ; k++)
    { for (l = k ; l <= 2 * n - k ; l++)
        pza[l] = pzb[l + 1] - pza[l] * bk[k - 1] - pzb[l] * ak[k - 1] ;
        bk[k] = pza[k] / pzb[k-1] ;
        ak[k] = pza[k + 1] / pza[k] - pzb[k]/pzb[k - 1] ;
        ptz = pza ;
        pza = pzb ;
        pzb = ptz ;
    }
}
```

APENDICE C

ESTIMADORES DEL NUMERO DE CONDICION

Este apéndice contiene los listados de los diferentes estimadores del número de condición para matrices triangulares y tridiagonales. En el último apartado se presenta el generador de matrices con cierta distribución de sus valores singulares y empleadas como matrices de prueba en el capitulo 5.

C.1 Subprograma DECOMP. Como fue comentado anteriormente el subprograma DECOMP que aparece en [Fo2] presenta un error tipográfico. El texto correcto aparece a continuación.

```
C
C     COND = (1-NORM OF A)*(AN ESTIMATE OF 1-NORM OF A-INVERSE)
C     ESTIMATE OBTAINED BY ONE STEP OF INVERSE ITERATION FOR THE
C     SMALL SINGULAR VECTOR.  THIS INVOLVES SOLVING TWO SYSTEMS
C     OF EQUATIONS, (A-TRANSPOSE)*Y = E  AND  A*Z = Y  WHERE  E
C     IS A VECTOR OF +1 OR -1 CHOSEN TO CAUSE GROWTH IN Y.
C     ESTIMATE = (1-NORM OF Z)/(1-NORM OF Y)
C
C     SOLVE (A-TRANSPOSE)*Y = E
C
      DO 50 K = 1, N
         T = 0.0D0
         IF (K .EQ. 1) GO TO 45
         KM1 = K-1
         DO 40 I = 1, KM1
            T = T + A(I,K)*WORK(I)
   40    CONTINUE
   45    EK = 1.0D0
         IF (T .LT. 0.0D0) EK = -1.0D0
         IF (A(K,K) .EQ. 0.0D0) GO TO 90
         WORK(K) = -(EK + T)/A(K,K)
   50 CONTINUE
      DO 60 KB = 1, NM1
         K = N - KB
         T = WORK(k)
         KP1 = K+1
         DO 55 I = KP1, N
            T = T - A(I,K)*WORK(I)
   55    CONTINUE
```

```
         WORK(K) = T
         M = IPVT(K)
         IF (M .EQ. K) GO TO 60
         T = WORK(M)
         WORK(M) = WORK(K)
         WORK(K) = T
   60 CONTINUE
C
      YNORM = 0.0D0
      DO 65 I = 1, N
         YNORM = YNORM + DABS(WORK(I))
   65 CONTINUE
```

C.2 Estimador retrospectivo. Modificación hecha al suprograma DRTCO de LINPACK para realizar el método retrospectivo. Una modificación importante fué eliminar el escalamiento. Esto no presentó problemas para los ejemplos del capítulo 5, sin embargo requiere de más análisis antes de eliminarlo definitivamente.

```
c
c      rcond = 1/(norm(t)*(estimate of norm(inverse(t)))) .
c     estimate = norm(z)/norm(y) where  t*z = y  and  trans(t)*y = e .
c      trans(t)  is the transpose of t .
c     the components of  e  are chosen to cause maximum local
c     growth in the elements of y .
c      the vectors are frequently rescaled to avoid overflow.
c
c     solve trans(t)*y = e
c
         do 20 j = 1, n
           z(j) = 0.0d0
   20    continue
         do 100 kk = 1, n
           k = kk
           if (lower) k = n + 1 - kk
           if (t(k,k) .eq. 0.0d0) go to 40
           wk =  -z(k)/t(k,k)
           wkm =  1.0d0/t(k,k)
           s = dabs(wk)
           sm= dabs(wkm)
           go to 50
   40      continue
           wk = 1.0d0
           wkm = 1.0d0
   50      continue
           i1 = 1
           if (lower) i1 = n - k + 1
           i2 = k - 1
           if (lower) i2 = n
```

```
            do 55 j = j1, j2
              s = s  + dabs(z(j))
              sm= sm + dabs(z(j))
 55         continue
            if (kk .eq. n) go to 65
            j1 = k + 1
            if (lower) j1 = 1
            j2 = n
            if (lower) j2 = k - 1
            do 60 j = j1, j2
              sm = sm + dabs(z(j)+wkm*t(k,j))
              z(j) = z(j) + wk*t(k,j)
              s = s + dabs(z(j))
 60         continue
 65         if (s .ge. sm) go to 80
            w = wkm - wk
            wk = wkm
            if (kk .eq. n) go to 72
            do 70 j = j1, j2
              z(j) = z(j) + w*t(k,j)
 70         continue
 72         do 75 j = j1, j2
              z(j) = 0.0d0
 75         continue
 80         continue
 90         continue
            z(k) = wk
100       continue
```

C.3 Método DIV-MOD.

C.3.1. Método DIV-MOD en Fortran.

```
      subroutine slvblk(t, ndim, n, z, it, n1, n2, y, znorm)
C
C Selecciona el valor de λ ∈ (0,1) para que la solución de sistema
C triangular
C
C                  | T11  T12 | | z1 |   | λd1     |
C              T z = |          | |    | = |         |
C                  |  0   T22 | | z2 |   | (1-λ)d2 |
C
C
C tenga norma grande. La solución se obtiene a partir de la
C solución de los los subsistemas.
C
C                          T11 y1 = d1
C
C                          T22 y2 = d2
C
C    La solución esta dada por:
C
```

```
C                z2 = (1-λ)y2
C
C                z1 = λy1 - w,    T11w = T12z2
C
C
C t(ndim, n)          Arreglo que contiene la matriz triangular.
C a(it,it)            Posición de la matriz t donde empieza el bloque
C                     T11.
C n1                  Tamaño de la matriz T11.
C n2                  Tamaño de la matriz T22. Esta matriz empieza en
C                     la posición t(it + n1, it + n1).
C
      integer ndim, n, it, n1, n2
      double precision t(ndim, n), z(n), y(n), znorm
      double precision s, znorm1, znorm2, ynorm

      znorm1 = 0.0d0
      ynorm  = 0.0d0
      jx = it  + n1
      jy = jx  + n2 - 1
      iy = it  + n1 - 1
C
C Calcula la norma de Z2
C
      znorm2 = 0.0d0
      do 10 j = jx, jy
        znorm2 = znorm2 + dabs(z(j))
10    continue
      do 50 i = 1, n1
        ix  = iy - i + 1
        s   = 0.0
        do 20 j = jx, jy
          s = s + t(ix, j) * z(j)
20      continue
        if (ix .eq. iy) go to 40
        do 30 j = ix + 1, iy
          s = s - t(ix, j) * y(j)
30      continue
40      y(ix) = s / t(ix, ix)
        ynorm  = ynorm  + dabs(y(ix))
        znorm1 = znorm1 + dabs(z(ix))
50    continue
      if ((znorm2 + ynorm) .ge. znorm1) go to 70
C
C λ = 0
C
      znorm = znorm1
      do 60 j = jx, jy
        z(j) = 0.0d0
60    continue
      go to 90
C
C λ = 1
```

```
C
70    znorm = znorm2 + ynorm
      do 80 j = it, iy
        z(j) = -y(j)
80    continue
90    return
      end

 subroutine dvenc(t, ndim, n, cond, z, y)
C
C Estimador del número de condición de una matriz triangular
C superior por medio del método de divide y venceras.
C
C t(ndim, n)          Matriz   triangular.
C ndim                Número de renglones del arreglo que contiene
C                     la matriz t.
C n                   Dimensión de la matriz t.
C cond                Estimador del número de condición.
C z(n), y(n)          Arreglos  auxiliares.
C
      integer ndim, n
      double precision t(ndim, n), z(n), y(n)
      double precision anorm, znorm, st
      integer ip, iq, k

      anorm = 0.0d0
      do 20 j = 1, n
        st = 0.0d0
        do 10 i = 1, j
          st = st + dabs(t(i,j))
10      continue
        if (t .gt. anorm) anorm = st
20    continue
      do 30 i = 1, n
        z(i) = 1.0d0 / t(i,i)
30    continue
      znorm = dabs(z(1))
      if (n .eq. 1)  go to 90
      ip = 1
40    ipd= 2 * ip
      ns = n  / ip
      na = n  / ipd
      np = mod(ns , 2)
      if (na. eq. 0) go to 80
      ix = 1
      do 50 i = 1, na
        n2 = ip
        if ((np .eq. 0) .and. (i .eq. na)) n2 = n - ip * (ns - 1)
          call slvblk(t, ndim, n, z, ix, ip, n2, y, znorm)
        ix = ix + ipd
50    continue
      if (np .eq. 0) go to 60
      ix = ipd * (na - 1) + 1
```

```
      call slvblk(t, ndim, n, z, ix, ipd, n - ipd * ns, y, znorm)
60    ip = ipd
      go to 40
80    cond = znorm * anorm
      return
      end
```

C.3.2. Método DIV-MOD en Matlab.

```
% Elige el valor de λ ∈ (0,1) y resuelve un sistema triangular por bloques
%
%                T z = [ T11  T12 ] [ z1 ]  =  [   λd1   ]
%                      [  0   T22 ] [ z2 ]     [ (1-λ)d2 ]
%
% de manera que la solución tenga norma grande.


function [y1, y2, ynrm] = slv(t11, t12, t22, y1, y2)
  z1    = solve(t11, t12*y2);
  y1nr  = norm(y1,1);
  y2nr  = norm(y2,1);
  z1nr  = norm(z1,1)
  if (z1nr + y2nr) < y1nr,
    [n,m] = size(y2);
    y2 = zeros(n, 1);
    ynrm = y1nr;
  else
    y1 = -z1;
    ynrm = z1nr + y2nr;
  end
return;

% Obtiene el estimador del número de condición de una matriz triangular
% por medio del método DIV-MOD.
function [y, cnd] = div(t)
  [n,m] = size(t);
  for ix=1:n, y(ix) = 1.0 / t(ix, ix); end
  y = y';
  kb = 1;
  cont=1;
  while cont,
    kd = 2 * kb;
    nm = fix(n / kb);
    ns = fix(n / kd);
    if(ns > 0),
      np = mod(nm, 2);
      i1 = 1;
      j1 = kb;
      for ix = 1:ns,
        if ((np == 0) .* (ix == ns)),
          i2 = j1 + 1;
          j2 = n;
        else
```

```
          i2 = i1 + kb;
          j2 = j1 + kb;
        end
        [y(i1:j1), y(i2:j2),ynr] = slv(t(i1:j1,i1:j1), t(i1:j1,i2:j2),
                                       t(i2:j2,i2:j2));
        i1 = i1 + kd;
        j1 = j1 + kd;
      end
      if (np ~= 0),
        i1 = i2;
        j1 = j2;
        i2 = j1 + 1;
        j2 = n;
        [y(i1:j1), y(i2:j2), ynr] = slv(t(i1:j1,i1:j1), t(i1:j1,i2:j2),
                                        t(i2:j2,i2:j2));
      end
      kb = kd;
    else
      cnd = ynr * norm(t,1);
      cont= 0;
    end;
  end;
return
end;
```

C.4 Estimadores de la norma de las matrices de comparación. Cálculo de la norma de las matrices $M(T)^{-1}$, $W(T)^{-1}$ y $Z(T)^{-1}$, donde T es uma matriz triangular superior. Los subprogramas siguientes corresponden a la norma infinito.

```
      DOUBLE PRECISION FUNCTION NRINFV(V, N)
C
C     CALCULA LA NORMA INFINITA DE UN VECTOR.
C     V(N)    VECTOR DE ENTRADA.
C     N       TAMAÑO DEL VECTOR.
C
      DOUBLE PRECISION V(N)
      DOUBLE PRECISION VMAX

      VMAX = 0.0D0
       DO 10 I = 1, N
         IF (DABS(V(I)) .GT. VMAX) VMAX = DABS(V(I))
10    CONTINUE
       NRINFV = VMAX
       RETURN
       END


      DOUBLE PRECISION FUNCTION NRIMTI(NDIM, N, T, Z)
```

```
C
C     CALCULA LA NORMA INFINITA DE LA INVERSA DE LA MATRIZ M(T).
C     T(NDIM,N)       MATRIZ TRIANGULAR SUPERIOR DE ORDEN N.
C     NDIM            NUMERO DE RENGLONES DEL ARREGLO QUE CONTIENE
C                        LA MATRIZ T.
C     N               ORDEN DE LA MATRIZ T.
C     Z(N)            ARREGLO AUXILIAR.
C
      DOUBLE PRECISION T(NDIM, N), Z(N)
      DOUBLE PRECISION S, NRINFV

      Z(N) = 1.0D0 / T(N, N)
      DO 50 IX = 1, N - 1
        I  = N - IX
        S  = 1.0D0
      DO 20 J = I + 1, N
         S = S + DABS(T(I,J)) * Z(J)
20    CONTINUE
        Z(I) = S / DABS(T(I,I))
50    CONTINUE
      NRIMTI = NRINFV(Z, N)
      RETURN
      END

      DOUBLE PRECISION FUNCTION NRIWTI(NDIM, N, T, Z)
C
C     OBTIENE LA NORMA INFINITA DE L A INVERSA DE LA MATRIZ W(T).
C     T(NDIM, N)      MATRIZ TRAINGULAR SUPERIOR.
C     NDIM            NUMERO DE RENGLONES DEL ARREGLO QUE CONTIENE
C                        A LA MATRIZ T.
C     N               ORDEN DE LA MATRIZ.
C     Z(N)            ARREGLO AUXILIAR
C
      DOUBLE PRECISION T(NDIM, N), Z(N)
      DOUBLE PRECISION S, ALFI, NRINFV

      Z(N) = 1.0D0 / DABS(T(N,N))
      S    = 0.0D0
      DO 40 IX = 1, N - 1
        I  = N - IX
        S = S + Z(I + 1)
      ALFI = 0.0D0
        DO 20 J = I + 1, N
          IF (DABS(T(I,J)) .GT. ALFI) ALFI = DABS(T(I,J))
20      CONTINUE
        Z(I) = (1.0D0 + ALFI * S) / DABS(T(I,I))
40    CONTINUE
      NRIWTI = NRINFV(Z,N)
      RETURN
      END

      DOUBLE PRECISION FUNCTION NRIZTI(NDIM, N, T)
```

```
C
C     CALCULA LA NORMA INFINITA DE LA MATRIZ INVERSA Z(T).
C      T(NDIM, N)      MATRIZ TRIANGULAR SUPERIOR.
C     NDIM            NUMERO DE RENGLONES DEL ARREGLO QUE CONTIENE
C                        A LA MATRIZ T.
C       N               ORDEN DE LA MATRIZ T.
C
        DOUBLE PRECISION T(NDIM, N)
        DOUBLE PRECISION ALFA, BETA, RMAX, TII

        ALFA = 0.0D0
        BETA = DABS(T(N,N))
        DO 20 I = 1, N - 1
          RMAX = 0.0D0
          DO 10 J = I + 1, N
            IF (DABS(T(I, J)) .GT. RMAX) RMAX = DABS(T(I,J))
10        CONTINUE
          TII = DABS(T(I, I))
          IF ( RMAX / TII .GT. ALFA ) ALFA = RMAX / TII
          IF ( TII .LT. BETA) BETA = TII
20      CONTINUE
        NRIZTI = (1.0D0 + ALFA)**(N-1)/BETA
        RETURN
        END
```

C.5. Estimador vía optimización convexa.

```
        subroutine   convx(a, ndim, n, cond, z, y, iter)
C
C Estimador del número de condición de una matriz  triangular
C  superior vía optimización convexa. Hager en [Ha1] presento
C inicialmente el método y posteriormente Higham en [Hi4] propuso
C algunas modificaciones.
C t(ndim, n)          Arreglo que contiene la matriz triangular.
C ndim                Número de renglones del arreglo.
C n                   Orden de la matriz.
C iter                Número de iteraciones (<= 5)
C z(n), y(n)          Arreglos  auxiliares.
C
      integer ndim, n, iter
      double precision a(ndim, n), y(n), z(n), cond
      double precision dasum, anorm, znor, zx, xi, t

      xi = 1.0d0 / dble(n)
      anorm = 0.0d0
      do 20 j = 1, n
        t = 0.0d0
        do 10 i = 1, j
          t = t + a(i, j)
10      continue
```

```
        if (t .gt. anorm) anorm = t
        y(j) = xi
20    continue
      je   = 0
      iter = 1
30     call dtrsl(a, ndim, n, y, 1, info)
      if (info .ne. 0) go to 95
      do 40 i = 1, n
        z(i) = 1.0d0
        if(y(i) .lt. 0.0d0) z(i) = -1.0d0
40    continue
      call dtrsl(a, ndim, n, z, 11, info)
      znorm = 0.0d0
      zx    = 0.0d0
      do 70 i = 1, n
        if (dabs(z(i)) .le. znorm) go to 50
        znorm = dabs(z(i))
        jpos  = i
50      if (je .eq. 0)  go to 60
        if (i  .eq. je) zx = z(i)
        go to 70
60      zx = zx + z(i) * xi
70    continue
      je = jpos
      if ((iter .ge. 5) .or. (znorm .le. zx)) go to 90
      do 80 i = 1, n
        y(i) = 0.0
80     continue
      y(je) = 1.0d0
      iter = iter + 1
      go to 30
90    cond = dasum(n, y, 1) * anorm
      return
95    cond = 1.0e+32
      stop
      end
```

C.6. Estimadores de matrices tridiagonales.

Dos métodos para estimar la norma de la inversa de una matriz tridiagonal.

```
#define ABS(x)          ((x) < 0 ? -(x) : (x))
#define MAX(x, y)       ((x) > (y) ? (x) : (y))

/*-------------------------------------------------------------*/
/* Procedimeinto para obetener la parte superior de la inversa */
/* de una matriz inferior de Hessemberg. De acuerdo al método  */
/* propuesto por Yasuhiko en [Ik1].                            */
```

```
/* El método consiste en  resolver los sistemas Ax = y_n^{-1} e_n  y   */

/* A^t y = x_1^{-1} e_1. La elección de x1 se hace en forma arbitraria  */

/* Este procedimiento puede emplearse para resolver sistemas ---*/
/* tridiagonales o bien para calcular la norma infinito de A^-1.  */
/* a[n]  Elementos de la diagonal de la matriz A.                 */
/* b[n]  Subdiagonal de la matriz A.                              */
/* c[n]  Superdiagonal de la matriz A.                            */
/* x[n]  Vector x.                                                */
/* y[n]  Vector y.                                                */
/* n  Orden de la matriz.                                         */
/*----------------------------------------------------------------*/

invhess(a, b, c, x, y, n)
float  a[];
float  b[];
float  c[];
float  x[];
float  y[];
int    n ;
{ int  i ;

  x[0] = 1.0 ;
  x[1] = -a[0] / c[0] ;
  for (i = 2; i < n; i++)
    x[i] = -(a[i - 1] * x[i - 1] + b[i - 1] * x[i - 2]) / c[i - 1] ;
  y[n - 1] = 1.0 / (b[n - 1] * x[n - 2] + a[n - 1] * x[n - 1]) ;
  y[n - 2] = - a[n - 1] * y[n - 1] / c[n - 2] ;
  for (i = n - 3; i >= 0; i--)
    y[i] = -(a[i + 1] * y[i + 1] + b[i + 2] * y[i + 2]) / c[i] ;
  return ;
}

/*----------------------------------------------------------------*/
/* Norma de la inversa de una matriz tridiagonal                  */
/* a[n]  Diagonal de la matriz.                                   */
/* b[n]  Subdiangonal de la matriz.                               */
/* c[n]  Superdiagonal de la matriz.                              */
/* n  Orden de la matriz.                                         */
/* Costo del algoritmo: 17n Flops.                                */
/*----------------------------------------------------------------*/

float  nrinftri(a, b, c, n)
float  a[] ;
float  b[] ;
float  c[] ;
int    n   ;
{ float x[50] ;
  float y[50] ;
  float p[50] ;
  float q[50] ;
  float s[50] ;
```

```
  float  t[50]  ;
  float  gama   ;
  int    i ;

  invhess(a, b, c, x, y, n) ;
  invhess(a, --c , ++b , q, p, n) ;
  s[n - 1] = ABS(y[n - 1]) ;
  for (i = n - 2; i >= 0; i--) s[i] = s[i + 1] + ABS(y[i]) ;
  t[0] = 1.0 ;
  for (i = 1; i < n; i++) t[i] = t[i - 1] + ABS(q[i]) ;
  gama = MAX(s[0], ABS(p[n - 1]) * t[n - 1]) ;
  for (i = 1; i < n - 1; i++)
    gama = MAX(gama, ABS(p[i]) * t[i - 1] + ABS(x[i]) * s[i]) ;
  return(gama) ;
}

/*-------------------------------------------------------------------*/
/* Segundo algoritmo para determinar la norma infinita de la         */
/* inversa de una matriz tridiagonal. Este algoritmo esta basado     */
/* en el Teorema 2 que aparece en [Hi3, pag 153].                    */
/* Costo del algoritmo: 14n flops.                                   */
/* a[n]  Diagonal de la matriz.                                      */
/* b[n]  Subdiagonal de la matriz.                                   */
/* c[n]  Superdiagonal de la matriz.                                 */
/* n  Orden de la matriz.                                            */
/*-------------------------------------------------------------------*/

float  nrinfitri(a,  b,  c,  n)
float  a[]  ;
float  b[]  ;
float  c[]  ;
int    n ;
{ float  x[50]  ;
  float  z[50]  ;
  float  s[50]  ;
  float  t[50]  ;
  float  d[50]  ;
  float  teta   ;
  float  gama   ;
  int    i ;

  x[0] = 1 ; x[1] = -a[0] / c[0] ;
  for (i = 2; i < n; i++)
    x[i] = -(a[i - 1] * x[i - 1] + b[i - 1] * x[i - 2]) / c[i - 1] ;
  z[n - 1] = 1; z[n - 2] = -a[n - 1] / b[n - 1] ;
  for (i = n - 3; i >= 0; i--)
    z[i] = -(a[i + 1] * z[i + 1] + c[i + 1] * z[i + 2]) / b[i + 1] ;
  teta = a[0] * z[0] + c[0] * z[1] ;
  s[n - 1] = ABS(z[n - 1]) ;
  for (i = n - 2; i >= 0; i--)
    s[i] = s[i + 1] + ABS(z[i]) ;
  t[0] = 1 ;
  for (i = 1; i < n - 1; i++)
```

```
    t[i] = t[i - 1] + ABS(x[i]) ;
  d[0] = 1.0; gama = s[0] ;
  for (i = 1; i < n; i++)
  { d[i] = d[i - 1] * c[i - 1] / b[i] ;
    gama = MAX(gama, (ABS(z[i]) * t[i - 1] + ABS(x[i]) * s[i])*ABS(d[i]));
  }
  return(gama / ABS(teta)) ;
}
```

C.7. Generación de matrices de prueba con cierta distribución de valores singulares.

```
      subroutine dgeaph(v, x, y, ix, iy)
c
c     Aplica una transformación de Householder a un vector.
c     v(n)     Vector que representa la transformacion.
c     x(n)     Vector a transformar.
c
      integer ix, iy
      double precision v(1), x(1), y(1), s, vn
      double precision dnrm2

      n  = iy - ix + 1
      vn = 2.0d0 / dnrm2(n, v(ix), 1)
      do 40  i = ix, iy
        s = 0.0d0
        do 20 j = ix, iy
          s = s + v(i) * v(j) * x(j)
20      continue
        y(i) = x(i) - vn * s
40    continue
      return
      end



      subroutine dgegnr(a, ndim, n, dseed)
c
c     Genera los vectores x(1), x(2), ..., x(n-1), independientes
c     y distribuidos normalmente n(0,1).
c     a(ndim, n)      Matriz donde se almacenan las variables
c                     generadas.
c     ndim            Número de renglones del arreglo que contiene
c                     la matriz a.
c     n               Orden de la matriz a.

 integer ndim, n
 double precision a(ndim, n), dseed
 real   rand(50)

      do 20 j = 1, n - 1
       nr = n - j + 1
```

```
          call ggnpm(dseed, nr, rand)
          do 10 i = j, n
            a(i, j) = rand(i - j + 1)
10        continue
20      continue
        return
        end


 subroutine dgogen(h, q, ndim, n, work, isgn)
c
c     Genera una matriz ortogonal de orden. Emplea el método
c     propuesto por Stewart en [St2].
c
        integer ndim, n, isgn(n)
        double precision h(ndim, n), q(ndim, n), work(n)
        double precision dnrm2, nrh, s

        isgn(n) = 1
        do 60 k = 1, n - 1
          i  = n - k
          kp = k + 1
          s  = h(i, i)
          isgn(i) = 1
          if (s. lt. 0.0d0) isgn(i) = -1
          do 20 l = i, n
            work(l) = h(l,i)
20        continue
          nrh= dnrm2(kp, work(i), 1)
          work(i) = s + dsign(nrh, s)
          do 40 j = i, n
            call dgeaph(work, q(i, j), h(i, j), i, n)
40        continue
60      continue
        do 80 i = 1, n
          do 70 j = 1, n
            q(i, j) = h(i, j) * dble(isgn(i))
70        continue
80      continue
        return
        end


        subroutine dgesvd(a, u, s, v, ndim, n)
c
c     Genera una matrix con valores singulares con cierta
c     distribución. El método consiste en calcular dos matrices
c     ortogonales y aleatorias, u y v,  y aplicar la
c     descomposición en valores singulares. a = u * s * v.
c

        integer ndim, n
```

```
      double precision a(ndim, n), u(ndim, n), v(ndim, n), s(n)

      do 20 j = 1, n
      do 20 i = 1, n
        u(i, j) = u(i, j) * s(j)
20     continue
      call dgepro(u, v, a, ndim, n)
      return
      end
```

BIBLIOGRAFIA

[Ab1]Handbook of mathematical functions
Edited by Abramowitz and Irene A. Stegun
Dover, New York, 1972

[An1]Ned Anderson and Ilkka Karasalo
On computing bounds for the least singular value of triangular matrix.
Bit 15, 1-4(1975).

[Bj1]Åke Björck and Victor Pereyra
Solution of Vandermonde systems of equations
Math. Comp., Vol. 24, No. 112, 893-903(1970)

[Bl1]S. L. Blank, Nishan Krikorian and David S.
A geometrically inspired proof of singular value descomposition
The American Math. Monthly, Vol 96, No. 3, 238-239(1989).

[Ca1]B. C. Carlson and John Todd
Zolotarev's first problem -the best approximation by polynomials of degree ≤ n-2 to x^n-nrx^{n-1} in [-1,1]-
Aequationes Mathematicae 26, 1-33(1983)

[Ch1]Tony F. Chan and D. E. Foulser
Efectively well-conditioned linear systems
SIAM J. SCI. STAT. COMPUT., Vol.9, No.6, 963-969(1988)

[Cl1]A. K. Cline, C. B. Moler, G. W. Stewart and J. H. Wilkinson
An estimate for the condition number of a matrix
SIAM J. Numer. Anal. Vol 16. No. 2, 368-375(1979)

[Cl2]A. K. Cline and K. Rew
A Set of counter-examples to three condition number estimators
SIAM J. Sci. Stat. Comput., Vol 4, No. 4. 602-611(1983)

[Cl3]Alan K. Cline, Andrew R. Conn, Charles F. Van Loan
Generalizing the Linpack condition estimator
Lectures notes in Mathematics edited by D. Dold and B. Eckmann. In Numerical Analysis Proceedings, Cocoyoc, México Springer-Verlag, 73-83(1981).

[Co1]William J. Cody Jr.

Software manual for elementary functions.
Prentice Hall series in Comp. and Math. 1980.

[de1]Carl deBoor
A practical guide to splines
Springer -Verlag, New York, 1978

[Do1]J. J. Dongarra, C. B. Moler, J. R. Bunch and G. W. Stewart
Linpack, user's guide
SIAM, Philadelphia, 1979.

[Fo1]George Forsythe, Cleve B. Moler
Computer solution of linear algebraic system
Prentice-Hall, Inc. Englewood Cliffs, N.J., 1976

[Fo2]George Forsythe, M. A. Malcolm and C. B. Moler
Computer methods for mathematical computations
Prentice-Hall, Inc. Englewood Cliffs, NJ., 1977

[Fo3]George Forsythe
Today's computational methods of linear algebra
SIAM Review 9, 489-515

[Fo4]L. Fox and I. B. Parker
Chebyshev polinomials in numerical analysis
Oxford University Press, 1968

[Ga1]Walter Gautschi
On inverses of Vandermonde and confluent Vandermonde matrices.
Numer. Math. 4, 117-123(1962)

[Ga2]Walter Gautschi
On inverses of Vandermonde and Confluent Vandermonde matrices, II.
Numer. Math. 5, 425-430(1963)

[Ga3]Walter Gautschi
Construction of Gauss-Christofel quadrature formulas
Math. Comp. 22, 251-270(1968)

[Ga4]Walter Gautschi
On the construction of Gaussian quadrature rules from modified moments.
Mat. Comp. Vol. 24, No. 10, 245-260(1970)

[Ga5]Walter Gautschi
The condition of orthogonal polynomials.
Mat. Comp. Vol 26, No. 120, 923-924(1972)

[Ga6]Walter Gautschi

Norm estimates for inverses of Vandermonde matrices
Numer. Math. 23, 337-347(1975)

[Ga7]Walter Gautschi
Optimally conditioned Vandermonde matrices
Numer. Math. 24, 1-12(1975)

[Ga8]Walter Gautschi
On inverses of Vandermonde and Confluent Vandermonde matrices, III.
Numer. Math. 29, 445-450(1978)

[Ga9]Walter Gautschi
Questions of numerical condition related to polynomials
In Recent advances in Numerical Analysis edited by C. de Boor and G. H. Golub. Academic Press, 45-72(1978)

[Ga10]Walter Gautschi
The condition of polynomials in power form.
Math. Comp. Vol 33. No. 145, 343-352(1979)

[Ga11]Walter Gautschi
On generating orthogonal polynomials.
SIAM J. Sci. Stat. Comput, Vol 3. No.3, 289-317(1982)

[Ga12]Walter Gautschi and Gabriele Inglese
Lower Bounds for the condition number of Vandermode matrices
Numer. Math. 52, 241-250(1988)

[Go1]Gene H. Golub, Charles F. Van Loan
Matrix computations
Baltimore: Johns Hopkins Press, 1983

[Go2]Gene Golub and John H. Welsh
Calculation of Gauss quadratures rules.
Math. Comput. 23, 221-230(1969)

[Gr1]R. T. Gregory and D. L. Karney
A Collection of matrices for testing computational algorithms
Robert E. Krieger Publishing Co. Huntington, NY., 1978

[Ha1]William W. Hager
Condition estimates
SIAM J. SCI. STAT. COMPUT. Vol.5, No.2, 311-316(1984)

[Ha2]W. Haussmann and K. Zeller.
Approximate Zolotarev polynomials.
Comp. & Math. with Appls. Vol. 12B, No. 5/6, 1133-1140 (1986)

[Ha3]P. R. Halmos
Measure theory

Van Nostrand Reinhold company, 1950.

[He1]Henrici, P.
Applied and computational complex analysis, Vol I.
Wiley-Interscience, New York, 1974

[He2]Henrici, P.
Essentials of numerical analysis with pocket calculator demostrations
John Wiley & Sons, 1982

[Hi1]Nicholas J. Higham
A survey of condition number estimation for triangular matrices
SIAM Review, Vol.29, No.4, 575-596(1987)

[Hi2]Nicholas J. Higham
Efficient algoritms for computing the condition number of a tridiagonal matrix.
SIAM. Sci. Stat. Comput. Vol. 7. No. 1, 150-165(1986)

[Hi3]Nicholas J. Higham
FORTRAN codes for estimating the one-norm of a real o complex matrix, with applications to condition estimation.
ACM Trans. Math. Softw. Vo 14, No. 4, 381-396(1988)

[Hi4]Nicholas J. Higham
Error analysis of Björck-Pereyra algorithms for solving Vandermonde systems
Numer. Math. 50, 613-632(1987)

[Hi5]Nicholas J. Higam
The accurancy of solutions to triangular systems
SIAM J. NUMER. ANAL., Vol. 26, 1252-1265(1989)

[Ik1]Yasuhiko Ikebe
On inverses of Hessemberg matrices
Liner Algebra and its Appl. 24:93-97(1979)

[In1]The 8086 family user's manual. Numeric supplement
Intel, 1980.

[Is1]Eugene Isaacson and H. Bishop Keller
Analysis of numerical methods
John Wiley & Sons, 1966

[Ja1]P. Jansen and P. Weidner
High -accuracy arithmetic software- some tests of ACRITH problem-solving routines
ACM Trans. On Math. Softw., Vol. 12, No. 1, 62-70(1986).

[Ka1]Ilkka Karasalo

A criterion for truncation of The QR-Descomposition algorithm for the singular linear least squares problem.
Bit 14, 156-166(1974)

[Kn1]Donald E. Knuth
Evaluation of polynomials by computer
Comm. ACM, Vol 5, 595-599(1962)

[Lei]Frans Lemeire
Bounds for condition numbers of triangular and trapezoid matrices.
Bit 15, 55-64(1975)

[Ma1]MATLAB matrix software
Numerical Algorithms Group, Inc, 1101 31st St., Sherborn, Ma 01770

[Mil]Michael A. Malcolm
Algoritms to reveal properties of floating-point arithmetic.
Comm. Of ACM, Vol- 15, No. 11, 949-951(1972)

[No1]Ben Noble
Applied linear Algebra
Prentice Hall, 1969

[Os1] A. Ostrowski
Über die determinanten mit überwiegender Hauptdiagonale
Comment Math. Hel., 10, 69-96(1937)

[O'l]Dianne Prost O'leary
Estimating matrix condition numbers
SIAM J. SCI. STAT. COMPUT. Vol.1, No.2, 205-209(1980)

[Po1]George Poole and Thomas Boullion
A Survey on M-matrices
SIAM Review, Vol.16, No.4, 419-427(1974)

[Ra1]Gerhard Rayna
REDUCE software for algebraic computation
Springer-Verlag New York Inc. 1987

[Ri1]John R. Rice
A theory of condition
J. SIAM. Numer. Anal. Vol. 3, No. 2, 287-310(1966)

[Ri2]John R. Rice
Numerical methods, software, and analysis
McGraw-Hill, Inc. 1983

[Ri3]John R. Rice
Matrix computations & mathematical software
McGraw-Hill, Inc. 1981

[Ri4]Theodore J. Rivlin.
The Chebyshev polynomials
John Wiley and Sons NY, 1974

[Sa1]I. R. Savage and Eugene Lukacs
Tables of inverses, of finite segments of Hilbert matrix.
Contributions to the determination of eigenvalues.
National Bureau of Standards applied Math. Series, no. 39,
105-108 (1954).

[Sa2]R. A. Sack and A. F. Donovan
An algorithm for Gaussian quadrature given modified moments
Numer. Math. 18, 465-478 (1972)

[St1]G. W. Stewart
Introduction to matrix computations
Ne York: Academic Press, Inc. 1973

[St2]G. W. Stewart
The efficient generation of random orthogonal matrices
with an application to condition estimators
SIAM J. Numer. Anal., Vol 17, No. 3, 403-409(1980)

[Sk1]Robert D. Skeel
Scaling for numerical stability in Gaussian elimination
J. Of ACM, Vol. 26, No. 3, 494-526(1979)

[To1]John Todd
Basic numerical mathematics Vol2: Numerical Algebra
Birkhäuser Verlag Basel, 1977

[Va1]Jan Van Tiel
Convex analysis, an introductory text
New York: John Wiley and Sons, 1984

[Vo1]Von Neumann, J, and H. H. Goldstine
Numerical inverting of matrices of high order
Bull. Amer. Math. Soc., 53, 1021-1099(1947)

[We1]Joan R. Westlake
A handbook of numerical matrix inversion and solution of
linear equations.
Control Data Corporation, 1968

[Wh1]John C. Wheeler
Modified moments and Gaussian quadratures.
Rocky mountain, Journal of Mathematics, Vol.4, No. 2, 287-296(1974)

[Wi1]J. H. Wilkinson
The evaluation of the zeros of ill-conditioned polynomials, Part I
Numer. Math. 1, 150-166 (1959).

[Wi2]J. H. Wilkinson
The evaluation of the zeros of ill-conditioned polynomials, Part II.
Numer. Math, 1, 167-180 (1959)

[Wi3]J. H. Wilkinson
Rounding errors in algebraic process
Englewood Cliffs, N. J.:Prentice Hall, 1963

[Wi4]J. H. Wilkinson
The algebraic eigenvalue problem
Oxford: Clarendon Press, 1965

[Wi5]J. H. Wilkinson
The perfidious polynomial
Questions of numerical condition related to polynomials
in recent advances in numerical analysis, C. de Boor and G. Golub, eds. Academic Press, NY, 1-28(1978)

[Zi2]G. Zielke
Some remarks on matrix norm, condition numbers and error estimates for linear equations.
Linear Algebra and its Appl. 110:29-41(1988)