Vol. **15**, **4**, 461–470 (1999)

Revista Internacional de Métodos Numéricos para Cálculo y Diseño en Ingeniería

Descomposiciones ortogonales para el cálculo del rango numérico matricial

Gregorio Quintana Ortí y Enrique S. Quintana Ortí

Departamento de Informática
Universidad Jaime I
Campus Penyeta Roja
12071 Castellón, España
Tel.: 34-964-72 83 42, Fax: 34-964-72 84 35
e-mail:gquintan@inf.uji.es
http://venus.unl.edu.ar/gtm-eng.html

Resumen

El cálculo del rango numérico matricial surge en numerosas aplicaciones de la ciencia y de la ingeniería. Actualmente existen tres aproximaciones numéricas básicas para efectuar este cálculo: la descomposición SVD, la descomposición URV y las descomposiciones QR reveladoras de rango (QRRR).

En este trabajo se analizan experimentalmente varios algoritmos secuenciales, basados en las tres aproximaciones anteriores para el cálculo del rango numérico matricial. Así, en el estudio comparativo experimental se emplea una implementación propia para el cálculo de la descomposición URV y dos nuevas rutinas para el cálculo de la descomposición QRRR. Además se utilizan las rutinas de la librería LAPACK para el cálculo de la descomposición SVD y la descomposición QR con pivotamiento de columnas.

Los resultados experimentales muestran que la descomposición QRRR es en la práctica tan fiable como las costosas descomposiciones SVD y URV. Además, estas descomposiciones QRRR presentan la ventaja fundamental de su bajo coste computacional.

ORTHOGONAL DESCOMPOSITIONS FOR COMPUTING THE NUMERICAL RANK

Summary

Many applications from science and engineering require the computation of the numerical rank of a matrix. Currently, there exist three basic approaches for such computation: the SVD decomposition, the URV decomposition and the rank-reveling QR (RRQR) decomposition.

In this paper, several serial algorithms for the computation of the numerical rank, based on the above-mentioned approaches, are experimentally analyzed. Thus, the experimental study includes our own implementation of the URV decomposition and two new routines for the RRQR factorization. Furthermore, we employ routines from LAPACK for the SVD decomposition and the QR factorization with column pivoting.

The experimental results show that the RRQR decomposition is, in practice, a reliable as the computationally expensive SVD and URV decompositions. Furthermore, these RRQR decompositions present a much lower computational cost.

 ISSN: 0213–1315 Recibido: Noviembre 1995

INTRODUCCIÓN

Consideremos la matriz $A \in \mathbb{R}^{m\times n}$ donde, sin pérdida de generalidad, $m \geq n$. Esta matriz puede descomponerse en el producto de dos matrices de columnas ortonormales $U = [u_1, u_2, \ldots, u_n] \in \mathbb{R}^{m\times n}$ y $V = [v_1, v_2, \ldots, v_n] \in \mathbb{R}^{n\times n}$ y una matriz diagonal $n \times n$ como sigue

$$U^T AV = \mathrm{diag}(\sigma_1, \sigma_2, \ldots \sigma_n) \tag{1}$$

donde $\sigma_{\max}(A) = \sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_n = \sigma_{\min}(A) \geq 0$. La descomposición (1) se conoce como la descomposición en valores singulares, o SVD, en tanto que los elementos $\sigma_1, \quad i = 1, \ldots, n$ son los valores singulares de la matriz A y $u_i, v_i, \quad i = 1, \ldots, n$ son respectivamente los correspondientes vectores singulares por la izquierda y por la derecha.

El rango de la matriz se define como

$$\text{rango } (A) = \dim(\mathrm{lin}(A)) \tag{2}$$

es decir, el rango es la dimensión del subespacio generado por la envoltura lineal de las columnas de A[11]. A partir de ahora consideraremos que $A \in \mathbb{R}^{m\times n}$ tiene rango r.

Existe una conocida relación entre el rango matricial y los valores singulares de la matriz[11], de modo que

$$\text{rango } (A) = r \quad \text{si y sólo si} \quad \sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_r > \sigma_{r+1} = \sigma_n = 0 \tag{3}$$

En la práctica, el uso de ordenadores donde la precisión es inevitablemente finita provoca la introducción de errores de redondeo en los cálculos aritméticos. Por esta razón se define el rango numérico matricial[10]. En concreto, A tiene rango numérico matricial r con respecto a ϵ si y sólo si

$$r = \text{rango } (A\epsilon) = \min_{||A-B||_2 \leq \epsilon} \text{rango } (B) \tag{4}$$

El cálculo del rango numérico de matrices densas es un problema que está presente en diversas aplicaciones de ingeniería; por ejemplo en la solución de problemas de mínimos cuadrados aparece en el diseño asistido por ordenador[13] la solución de ecuaciones integrales[8], el cálculo de *splines*[12], la formación de rayos[4], la estimación espectral[16], la regularización[14], etc. Recientemente, el cálculo del rango numérico también está cobrando especial importancia en problemas de división espectral[2,22].

Existen básicamente tres tipos de métodos numéricos para el cálculo del rango numérico: la descomposición SVD, la descomposición URV y la descomposición QR reveladora de rango (que abreviamos como QRRR).

El estudio del rango numérico matricial mediante la descomposición SVD es altamente fiable desde el punto de vista numérico. Además, esta descomposición ofrece información completa sobre la distribución de los valores singulares de la matriz. Sin embargo, si únicamente interesa conocer el rango numérico, esta descomposición tiene un coste computacional excesivo. En consecuencia, la descomposición SVD sólo resulta viable para matrices de dimensión reducida y/o problemas donde la precisión de los resultados juega un papel primordial.

Una alternativa diferente, conocida como la descomposición URV[21], permite la separación del subespacio nulo de una matriz y proporciona una base explícita de este subespacio. En esta descomposición se factoriza A en el producto de dos matrices de columnas ortonormales $U \in \mathbb{R}^{m\times n}$ y $V \in \mathbb{R}^{n\times n}$ y una matriz triangular superior $n \times n$ de la forma

$$U^T AV = \begin{pmatrix} R_{11} & R_{12} \\ 0 & R_{22} \end{pmatrix} \tag{5}$$

donde $R_{11} \in \mathbb{R}^{r\times r}$ (siendo r el rango numérico de A) y $\|R_{12}\|$ y $\|R_{22}\|$ son del mismo orden que σ_{r+1}. Esta descomposición resulta tan fiable como la SVD para la separación del subespacio nulo de una matriz. Además, presenta la ventaja adicional, especialmente interesante en problemas de procesamiento de la señal, de una actualización fácil y de reducido coste si se añaden nuevas filas a A.

Finalmente, en la descomposición QR reveladora de rango (QRRR) se descompone A en el producto de una matriz $Q \in \mathbb{R}^{m\times n}$ con columnas ortonormales, una matriz de permutación $P \in \mathbb{R}^{n\times n}$ y una matriz triangular superior $n \times n$ de la forma

$$U^T AP = \begin{pmatrix} R_{11} & R_{12} \\ 0 & R_{22} \end{pmatrix} \tag{6}$$

donde $\|R_{11}\|_2\|R_{11}^{-1}\|_2 = \sigma_1(R_{11})/\sigma_r(R_{11}), R_{11} \in \mathbb{R}^{r\times r}$ y $\|R_{22}\|_2$ es del orden σ_{r+1}. Es fácil observar que la descomposición QRRR ofrece mucha menos información que los dos tipos de descomposiciones anteriores. Por ejemplo, no se dispone directamente de los valores singulares de la matriz como en la SVD, ni de una base del subespacio nulo como en la descomposición URV. En cambio, la descomposición (6) sí permite detectar, con un considerablemente menor coste computacional, el rango numérico de la matriz al descomponer A en una submatriz R_{11} bien condicionada y una submatriz R_{22} de norma pequeña.

En este trabajo se presenta un análisis experimental comparativo de los tipos de descomposicones descritos. El estudio incluye la descomposición SVD, disponible en la librería LAPACK, una implementación propia de la descomposición URV, la descomposición QRRR más difundida: la descomposición QR con pivotamiento de columnas y dos nuevos algoritmos para la descomposición QRRR.

El trabajo está estructurado de modo que en la sección siguiente se describen los métodos numéricos estudiados para el cálculo del rango numérico matricial, a saber, las descomposiciones SVD, URV y QRRR. En el apartado posterior se presentan los resultados experimentales del estudio comparativo. En el apartado final se ofrecen las conclusiones de este trabajo.

MÉTODOS PARA EL CÁLCULO DEL RANGO NUMÉRICO

La descomposición SVD

El algoritmo de Golub-Reinsch[11] permite calcular la descomposición SVD, definida en (1). En este algoritmo, en primer lugar se bidiagonaliza la matriz A mediante reflectores de Householder[11], como se muestra seguidamente.

Algoritmo de bidiagonalización

Para $j = 1{:}n$
    Calcular un reflector de Householder $U_j \in \mathbb{R}^{m-j+1\times m-j+1}$, tal que
        $U_jA(j{:}m, j) = (\|A(j{:}m, j)\|_2, 0, \ldots, 0)^T$
    Aplicar el reflector de Householder U_j a la matriz
        $A(j{:}m, j{:}n) \leftarrow U_jA(j{:}m, j{:}n)$
    Si $(j \leq n-2)$
        Calcular un reflector de Householder $V_j \in \mathbb{R}^{n-j\times n-j}$, tal que
            $V_jA(j, j+j{:}n)^T = (\|A(j, j+1{:}n)^T\|_2, 0, \ldots, 0)^T$
        Aplicar el reflector de Householder V_j a la matriz
            $A(j{:}m, j+1{:}n) \leftarrow A(j{:}m, j+1{:}n)V_j$
    Fin Si
Fin Para

A continuación se aplica un procedimiento iterativo[9] a la matirz bidiagonal obtenida que anula los elementos no diagonales. Una variación de este procedimiento iterativo[7], a menudo más eficiente, está disponible en la librería de núcleos computacionales LAPACK[1].

El algoritmo de Golub-Reinsch para el cálculo de la descomposición SVD requiere $o(4mn^2 - 4n^3/3)$ operaciones en aritmética de coma flotante (*flops*) para obtener la matriz diagonal de la descomposición y determinar de este modo el rango numério de la matriz.

La descomposición URV

En la referencia[21] se presenta un algoritmo para el cálculo de la descomposición URV, definida en (5). Asimismo, se describe una versión refinada de este algoritmo que permite obtener una matriz triangular superior con elementos diagonales dominantes.

El algoritmo para el cálculo de la descomposición URV requiere una triangularización inicial de la matriz, que puede calcularse con algún tipo de descomposición ortogonal como, por ejemplo, la descomposición QR obtenida mediante reflectores de Householder, rotaciones de Givens o el algoritmo de Gram-Schmidt[11].

Una vez se dispone de esta descomposición inicial, digamos $U^T A = R$, en primer lugar debe determinarse si la matriz triangular R tiene rango completo mediante algún tipo de estimador de condición[15]. Si rango $(R) = n$, la descomposición QR obtenida es asimismo una descomposición URV (siendo V la matriz identidad); en caso contrario es necesario aplicar un procedimiento directo que reduce paulatinamente la matriz triangular R a la forma deseada. El algoritmo que calcula esta descomposición, mediante rotaciones de Givens[11], se describe a continuación.

Algoritmo URV

Calcular una descomposición QR de la matriz A
    $A = UR$
$i \leftarrow n$
Mientras $(\sigma_{\min}(R(1{:}i, 1{:}i) < 0)$ y $(i > 0)$
    Calcular el vector singular por la derecha $v_i \in \mathbb{R}^i$ asociado a $\sigma_{\min}(R(1{:}i, 1{:}i))$
    Para $\hat{k} = 1{:}i-1$
        $k \leftarrow i - \hat{k}$
        Calcular una rotación de Givens $G^r_{i,k} \in \mathbb{R}^{2\times 2}$ tal que
            $G^r_{i,k} v_i(k{:}k+1) = (\|v_i(k{:}k+1)\|_2, 0)$
        Aplicar la rotación de Givens $G^r_{i,k}$ a la matriz
            $R(1{:}k+1, k{:}k+1) \leftarrow R(1{:}k+1, k{:}k+1) G^r_{i,k}$
        Calcular una rotación de Givens $G^l_{i,k} \in \mathbb{R}^{2\times 2}$, tal que
            $G^l_{i,k} R(k{:}k+1, k) = (\|R(k{:}k+1, k)\|_2, 0)$
        Aplicar la rotación de Givens $G^l_{i,k}$ a la matriz
            $R(1{:}k+1, k{:}k+1) \leftarrow G^l_{i,k} R(k{:}k+1, k{:}n)$
    Fin para
    $i \leftarrow i - 1$
Fin Mientras

Este algoritmo requiere $o(2mn^2 + 7n^3/2 - 25nr^2/6)$ *flops* si únicamente se calcula R (nótese que el procedimiento puede detenerse cuando se detecta que la matriz $R(1{:}i, 1{:}i)$ tiene rango completo y que el vector singular por la derecha puede estimarse con un coste aproximado de $o(i^2)$ *flops*).

La descomposición QR reveladora de rango (QRRR)

Si la matriz A es de rango deficiente, la descomposición QR no produce una base ortonormal de $lin(A)$. Del mismo modo, la descomposición QR así obtenida no revela el rango numérico de la matriz. En consecuencia, en la referencia[5] se introduce una estrategia de pivotamiento de columnas que sí proporciona esta información. El algoritmo para la descomposición QR con pivotamiento de columnas (o descomposición QRP) se puede describir del siguiente modo.

Algoritmo QRP

Para $j = 1{:}n$
  Calcular k tal que $\|A(j{:}m,k)\|_2 = \max_{i=j:n} \|A(j{:}m,i)\|_2$
  Calcular Π_j tal que $A\Pi_j$ tiene permutadas las columnas j-ésima y k-ésima
  $A \leftarrow A\Pi_j$
  Calcular un reflector de Householder, $Q_j \in \mathbb{R}^{m-j+1\times m-j+1}$, tal que
    $Q_jA(j+1{:}m,j) = (\|A(j+1{:}m,j)\|_2, 0 \ldots, 0)^T$
  Aplicar un reflector de Householder Q_j a la matriz:
    $R(1{:}j,j) \leftarrow Q_jA(1{:}j,j)$
    $A(j{:}m,j+1{:}n) \leftarrow Q_jA(j{:}m,j+1{:}n)$
Fin para

Este algoritmo requiere $o(4mnr - 2r^2(m+n) + 4r^3/3)$ *flops* si únicamente se construye R. Actualmente existe una implementación de la descomposición QRP en la librería LAPACK. Sin embargo, el empleo de esta descomposición para el cálculo del rango numérico matricial no es completamente fiable. Aunque en la práctica proporciona resultados correctos, desde hace algunos años se utiliza un tipo de matrices, conocidas como matrices de Kahan[17], para las cuales esta descomposición falla al revelar el rango.

Recientemente se han propuesto dos algoritmos de refinamiento para la descomposición QRP en las referencias[6,18]. En estos trabajos se describen aproximaciones teóricas al problema del cálculo del rango matricial de matrices triangulares mediante descomposiciones QRRR, pero no se realiza ninguna implementación de los algoritmos y se carece de un estudio experimental de las prestaciones mínimamente significativo. Ambos algoritmos de refinamiento parten de una descomposición ortogonal de la forma $Q^TA = R$ con o sin pivotamiento. Sin embargo, la utilización de la estrategia de pivotamiento es conveniente, puesto que reduce el trabajo de refinamiento que debe realizarse.

El algoritmo de Pan y Tang[18] tienen garantizada su finalización y a partir de una tupla de valores $(r,f), 0 < f \leq 1\sqrt{r+1}$ se obtiene una descomposición de la forma (6), donde

$$\begin{aligned} \sigma_{\min}(R_{11}) &\geq \sigma_r(A)f/\sqrt{r(n-r+1)} \\ \sigma_{\max}(R_{22}) &\leq \sigma_{r+1}(A)/\sqrt{(r+1)(n-r)/f} \end{aligned} \tag{7}$$

En este algoritmo, cuanto mayor es f, más estrictas son las cotas en (7) y los correspondientes valores singulares de R_{11} y R_{22} se aproximan más a los valores singulares r-ésimo y $r+1$-ésimo de A. Sin embargo, si f es demasiado grande, el algoritmo iterativo no se detiene debido a los errores de redondeo.

La implementación desarrollada en el presente trabajo incorpora diversas mejoras sobre el algoritmo original de Pan y Tang. En concreto, se realiza una reutilización cuidadosa de la información mediante la utilización de un estimador incremental de condición[15], que permite reducir considerablemente el número de operaciones del algoritmo. Así es posible obtener una estimación altamente fiable del menor valor singular de una matriz en $o(3n^2/2)$ *flops*, frente al orden cúbico del cálculo exacto del mismo valor singular. Además, se realiza un reordenamiento de las operaciones del algoritmo de Pan y Tang de modo que es necesario un menor número de iteraciones. Los resultados experimentales en la referencia[20] muestran que el nuevo algoritmo desarrollado resulta en la práctica entre 5 y 10 veces más rápido que el algoritmo original.

Chandrasekaran e Ipsen desarrollan en la referencia[6] un algoritmo de refinamiento iterativo que finaliza y devuelve una descomposición QRRR de la forma (6), con las siguientes cotas

$$\begin{aligned}\sigma_{\min}(R_{11}) &\geq \sigma_r(A)/\sqrt{r(n-r+1)}\\ \sigma_{\max}(R_{22}) &\leq \sigma_{r+1}(A)\sqrt{(r+1)(n-r)}\end{aligned} \tag{8}$$

En este trabajo se ha desarrollado una versión modificada, que presenta diversas mejoras respecto al algoritmo original de Chandrasekaran e Ipsen. Por ejemplo, se introduce un factor f, $0 < f \leq 1$, que realiza la misma función descrita en el algoritmo de Pan y Tang y que evita en consecuencia los problemas de terminación del algoritmo original. En la referencia[19] se demuestra que la versión modificada del algoritmo finaliza y devuelve una descomposición QRRR de la forma (6) con las siguientes cotas

$$\begin{aligned}\sigma_{\min}(R_{11}) &\geq \sigma_r(A)f^2/\sqrt{r(n-r+1)}\\ \sigma_{\max}(R_{22}) &\leq \sigma_{r+1}(A)\sqrt{(r+1)(n-r)/f^2}\end{aligned} \tag{9}$$

Además se introducen otras mejoras como la reutilización de la información, el empleo de estimadores para reducir el coste por iteración del algoritmo y la eliminación de iteraciones innecesarias mediante una reordenación del procedimiento, que reducen considerablemente el coste del algoritmo original[3]. En el estudio experimental en la referencia[20] el nuevo algoritmo ha resultado ser entre 2 y 40 veces más rápido que el algoritmo original, dependiendo del rango de la matriz estudiada y de su dimensión.

ESTUDIO EXPERIMENTAL DE LOS ALGORITMOS

En esta sección se presenta el estudio experimental comparativo de los algoritmos secuenciales para el cálculo del rango numérico implementados. En este estudio se incluyen los siguientes algoritmos:

- SVD: Algoritmo de Golub-Reinsch de la librería LAPACK para el cálculo de la descomposición SVD
- URV: Nueva implementación de la descomposición URV
- QRP: Factorización QR con pivotamiento de columnas de la librería LAPACK
- QP1: Nueva factorización QRRR compuesta por la descomposición QR con pivotamiento de columnas, seguida por el algoritmo de postprocesamiento de matrices triangulares de Chandrasekaran e Ipsen
- QP2: Nueva factorización QRRR compuesta por la descomposición QR con pivotamiento de columnas, seguida por el algoritmo de postprocesamiento de matrices triangulares de Pan y Tang.

En los experimentos se se ha utilizado un procesador segmentado ACE del ordenador Alliant FX/80. Los cálculos se han realizado en aritmética de simple precisión (en esta arquitectura, la precisión de la máquina es $\epsilon \approx 1,1 \times 10^{-8}$).

En el análisis de los algoritmos se han empleado matrices con un rango r que varía entre 1 y $n-1$. Los valores singulares de estas matrices están divididos en dos conjuntos bien separados. Los primeros r valores singulares se hallan distribuidos geométricamente entre 1 y 10^{-2}, mientras que los últimos $n-r$ valores singulares están distribuidos de la misma forma entre 10^{-5} y 10^{-7}.

Además, si $r = 1$, entonces $\sigma_1 = 1$; mientras que si $r = n - 1$, entonces $\sigma_n = 10^{-5}$. En consecuencia, el rango numérico de estas matrices está bien definido y vale r. Los vectores singulares han sido generados aleatoriamente y son, por tanto, los que hacen a las matrices diferentes unas de otras. Las matrices han sido construidas mediante las rutinas de generación de la librería LAPACK[1].

Estudio comparativo de las prestaciones

En primer lugar se presenta el tiempo medio (en segundos) de cinco ejecuciones sobre matrices diferentes de dimensión 200×200. Los resultados únicamente reflejan el tiempo necesario para el cálculo del rango matricial a partir del factor diagonal (descomposición SVD) o los factores triangulares (descomposiciones URV y QRRR). Así, no se ha tenido en cuenta el tiempo necesario para construir las matrices U, V o Q de estas descomposiciones.

Los resultados obtenidos con problemas de otras dimensiones han sido similares.

r	QRP	QP1	QP2	URV	SVD
1	8,032	8,095	8,466	30,766	21,997
2	8,085	8,220	8,460	30,775	22,170
25	8,103	8,258	8,400	30,485	22,060
50	8,103	8,172	8,339	29,545	22,084
75	8,118	8,190	8,288	27,922	22,089
100	8,099	8,166	8,254	25,578	22,071
125	8,115	8,182	8,222	22,411	22,185
150	8,116	8,202	8,199	18,419	22,098
175	8,032	8,137	8,187	13,566	21,926
198	8,077	8,208	8,206	8,282	22,047
199	8,077	8,209	8,191	8,036	22,054

Tabla I. Tiempo medio de cinco ejecuciones para matrices 200×200 con $\sigma_1 = 1$, $\sigma_r = 10^{-2}$ $(r \neq 1)$, $\sigma_{r+1} = 10^{-5}$ y $\sigma_n = 10^{-7}$ $(r \neq n-1)$

Los algoritmos basados en las descomposiciones SVD (`SVD`) y URV (`URD`) presentan tiempos de ejecución muy elevados, entre 2 y 4 veces mayores que los tiempos correspondientes a la descomposición QR con pivotamiento de columnas (QRP).

Un análisis más detallado revela que las prestaciones del algoritmo SVD no dependen en absoluto del rango.

En cambio, los tiempos de ejecución del algoritmo URV resultan mucho menores para matrices con un rango alto. Esto es debido a que en tal caso el coste computacional de la descomposición URV es notablemente menor (por ejemplo, si la matrix tiene rango completo, entonces el coste de esta factorización es del mismo orden que en la descomposición QR con pivotamiento de columnas). Puede observarse que, incluso para $r = n - 1$, el algoritmo URV resulta más eficiente que las descomposiciones QRRR.

Todos los métodos basados en las descomposiciones RRQR ofrecen unas prestaciones notablemente reducidas. Además, la etapa de postprocesamiento necesaria en los algoritmos QP1 y QP2 no supone un incremento importante en el tiempo de ejecución de los algoritmos.

Estudio comparativo de la precisión

En el estudio de precisión se han considerado, además del anterior tipo de matrices, las matrices de Hilbert, matrices con elementos distribuidos aleatoriamente entre 0 y 1, matrices de enteros, etc. En todos estos casos, las diferentes descomposiciones estudiadas han revelado correctamente el rango numérico matricial.

En segundo lugar, se han empleado en el análisis de la precisión un tipo de matriz considerado como conflictivo: las matrices de Kahan[17] ($r = n - 1$).

La siguiente tabla muestra los resultados para dos matrices de Kahan, 50×50 ($c = 0,2$) y 100×100 ($c = 0,1$). En concreto, se muestran los elementos diagonales $R(r,r)$ y $R(r+1, r+1)$, que deben revelar el rango numérico matricial.

	50×50		50×50	
Algoritmo	$\lvert R(49,49)\rvert$	$\lvert R(50,50)\rvert$	$\lvert R(99,99)\rvert$	$\lvert R(100,100)\rvert$
SVD	0,442	$0,9290 \times 10^{-4}$	0,6409	$0,9489 \times 10^{-4}$
URV	0,4504	$0,9290 \times 10^{-4}$	0,6722	$0,9489 \times 10^{-4}$
QRP	0,3754	0,3678	0,6111	0,6080
QP1	0,4505	$1,6808 \times 10^{-4}$	0,7759	$2,2780 \times 10^{-4}$
QP2	0,4505	$1,6808 \times 10^{-4}$	0,7759	$2,2780 \times 10^{-4}$

Tabla II. Precisión de los algoritmos para matrices de Kahan

Esta tabla muestra que la descomposición QR con pivotamiento de columnas (QRP), aunque construye una matriz triangular con elementos diagonales decrecientes, falla al revelar el rango numérico. En cambio, los nuevos algoritmos desarrollados en este trabajo (QP1 y QP2) y los algoritmos para las descomposiciones SVD y URV revelan correctamente el rango.

CONCLUSIONES

En este trabajo se ha realizado un estudio experimental de la descomposición SVD, una implementación propia para el cálculo de la descomposición URV, la descomposición QR con pivotamiento de columnas y dos nuevas rutinas para el cálculo de la descomposición QRRR.

Los resultados experimentales muestran que las nuevas descomposiciones QRRR son en la práctica tan fiables como las descomposiciones SVD y URV. Además, estas descomposiciones QRRR presentan la ventaja fundamental de un notable menor coste computacional.

AGRADECIMIENTOS

Este trabajo ha sido subvencionado por el proyecto CICYT TIC-96-1062-C03-03.

REFERENCIAS

1 E. Anderson, Z. Bai, C.H. Bischof, J. Demmel, J. Dongarra, J. DuCroz, A. Greenbaum, S. Hammerling, A. McKenney, S. Ostrouchov y D. Sorensen, *LAPACK User's Guide Release 2.0, SIAM*, Philadelphia, (1994).

2 Z. Baiy J. Demmel, "Design of a parallel nonsymmetric enginroutine toolbox", Part 1, Tech. Report UCB//CSD-92-718, Dept. of Computer Science, University of California, Berkeley.

3 C.H. Bischof y G. Quintana Ortí, "Computing rank-revealing QR factorizations of dense matrices", *ACM Transactions on Mathematical Software*, Vol. **24**,N° 2, pp. 226–253, (1998).

4 C.H. Bischof y G.M. Shroff, "*On updating signal subspaces*", Preprint MCS-P101-0989, Argonne National Laboratory, Mathematics and Computer Science Division, (1989).

5 P.A. Businger y G.H. Golub, "Linear least square solutions by Householder transformations", *Numer. Math.*, Vol. **7**, pp. 269–276, (1965).

6 S. Chandrasekaran e I. Ipsen, "On rank-revealing QR factorizations", *SIAM J. Mat. Anal. Appl.*, Vol. **15**, pp. 592–622, (1994).

7 J. Demmel y W. Kahan, "Accurate singular values of bidiagonal matrices", *SIAM J. Sci. Stat. Comp.*, Vol. **11**, pp. 873–912, (1990).

8 L. Eldén y R. Schreiber, "An application of systolic arrays to linear discrete ill-posed problems", *SIAM J. Sci. Stat. Comp.*, Vol. **7**, pp. 892–903, (1986).

9 G.H. Golub y W. Kahan, "Calculating the singular values and pseudo-inverse of a matirx", *SIAM J. Numerical Analysis*, Vol. **2**, pp. 205–224, (1965).

10 G.H. Golub, V. Klema y G.W. Stewart, "Rank degeneracy and last squares problems", Tech. Report STAN-CS-76-559, Computer Science Department, Stanford University, (1976).

11 G.H. Golub, C.F. Van Loan, "*Matrix computations*", The John Hopkins University Press, Baltimore, (1989).

12 T.A. Grandine, "An iterative method for computing multivariate C^1 piecewise polynomial interpolants", *Computer Aided Geom. Design*, Vol. **4**, pp. 307–319, (1987).

13 T.A. Grandine, "Rank deficient interpolation and optimal design: An example", Tech. Report SCA-TR-113, Boeing Computer Services, Engineering and Scientific Services Division, Seattle, Washington, (1989).

14 P.C. Hansen, "Truncated SVD solutions to discrete ill-posed problems with ill-determined numerical ranks", *SIAM J. Sci. Stat. Comp.*, Vol. **11**, pp. 503–518, (1990).

15 N.J. Higham, "A survey of condition number estimation of triangular matrices", *SIAM Rev.*, Vol. **29**, pp. 575–596, (1987).

16 S.F. Hsieh, K.J.R. Liu y K. Yao, "*Comparisons of truncated QR and SVD methods for AR spectrals estimations*", pp. 403–418, Elsevier Science Publishers, (1991).

17 W. Kahan, "Numerical linear algebra", *Canadian Math. Bull.*, Vol. **9**, pp. 757–801, (1966).

18 C.T. Pan, P.T.P. Tang, "Bounds on singular values revealed by QR factorizations", Tech. Report ANL-MCS-P322-1092, Argonne National Laboratory, Mathematics and Computer Science Division, (1992).

19 G. Quintana Ortí y E.S. Quintana Ortí, "Guaranteeing termination of Chandrasekaran and Ipsen's algorithm for computing rank-revealing QR factorizations", Tech. Report MCS-P564-0196, Argonne National Laboratory, Mathematics and Computer Science Division, (1996).

20 G. Quintana Ortí y E.S. Quintana Ortí, "Métodos secuenciales para el cálculo de la factorización RRQR", Informe técnico DI 2-1/96, Depto. de Informática, Universidad Jaime I, Castellón, (1996).

21 G.W. Stewart, "An updating algorithm for subspace tracking", *IEEE Trans. on Signal Processing*, Vol. **40**, pp. 1535–1541, (1992).

22 X. Sun y E.S. Quintana Ortí, "Spectral division methods for block generalized Schur decomposition", Tech. Report CS-1996-13, Computer Science Department, Duke University, (1996).