



Universidad de Valladolid

Facultad de Ciencias

Trabajo Fin de Grado

Grado en Matemáticas

La exponencial matricial: cálculo eficiente y aplicaciones

Autor: Iván Corredor Mazo

Tutora: María Paz Calvo Cabrero

Introducción

La exponencial matricial aparece en diversos problemas de las matemáticas, entre otros, en la solución de ecuaciones diferenciales lineales o en el estudio de redes de información. El objetivo de este Trabajo Fin de Grado es el estudio de métodos numéricos eficientes para el cálculo de la exponencial de una matriz, prestando especial atención al caso en que la matriz es de gran dimensión pero dispersa.

Así, en el Capítulo 1 se ha definido la exponencial matricial, se han revisado sus principales propiedades y se han incluido un par de resultados para acotar la norma de la diferencia entre las exponenciales de dos matrices próximas.

En el Capítulo 2, tras una revisión de los distintos métodos numéricos existentes en la literatura para el cálculo de la exponencial de una matriz (Sección 2.1), se han estudiado con detalle el método del escalado y el cuadrado, basado a su vez en la aproximación de Padé a la función exponencial (Sección 2.2) y el método basado en los subespacios de Krylov (Sección 2.3).

En el Capítulo 3 se han descrito un par de aplicaciones en las que los métodos estudiados son de utilidad, y se han resuelto algunos ejemplos concretos, para lo que se han implementado dichos métodos en MATLAB.

El Apéndice incluye los programas que implementan los métodos numéricos descritos en el Capítulo 2 y que se han utilizado para obtener las gráficas y los resultados numéricos incluidos en la memoria.

Índice general

Índice general	5
1. LA EXPONENCIAL MATRICIAL	7
2. MÉTODOS NUMÉRICOS PARA EL CÁLCULO EFICIENTE DE LA EXPONENCIAL	17
2.1. Diversos métodos numéricos encontrados en la literatura	17
2.2. El escalado y el cuadrado	21
2.3. Método de los espacios de Krylov.	30
3. APLICACIONES	41
3.1. Propiedades de redes de comunicación	41
3.2. Resolución de sistemas diferenciales lineales con término fuente constante	47
Bibliografía	53
A. Programas de MATLAB	55
A.1. padeexponencial.m	55
A.2. escaladoycuadrado.m	56
A.3. Arnoldi.m	56
A.4. subespaciodekrylov.m	57
A.5. centralidad.m	58
A.6. aproximacionphi1f.m	59
A.7. ecuacionHeston.m	60

Capítulo 1

LA EXPONENCIAL MATRICIAL

Dada una matriz A , nuestro primer objetivo es dar sentido a la expresión e^A . Para ello vamos a hacer uso de la serie de potencias de la función exponencial escalar,

$$e^z = 1 + z + \frac{z^2}{2} + \cdots + \frac{z^n}{n!} + \cdots, \quad (1.1)$$

que sabemos que converge para todo $z \in \mathbb{C}$.

Definición 1.1 Sea $A \in \mathbb{C}^{n \times n}$. Se define la exponencial $e^A \in \mathbb{C}^{n \times n}$ como la suma de la serie matricial

$$e^A = I + A + \frac{A^2}{2} + \cdots + \frac{A^n}{n!} + \cdots. \quad (1.2)$$

Para que la definición anterior tenga sentido veamos que la sucesión de sumas parciales

$$S_k = I + A + \frac{A^2}{2} + \cdots + \frac{A^k}{k!}, \quad k = 1, 2, \dots$$

converge. Para ello, utilizando alguna norma matricial derivada de una norma vectorial obtenemos

$$\|S_k\| \leq 1 + \|A\| + \frac{\|A\|^2}{2} + \cdots + \frac{\|A\|^k}{k!} = \sum_{j=0}^k \frac{\|A\|^j}{j!} \leq e^{\|A\|}$$

y, tomando límites cuando $k \rightarrow \infty$ en la expresión anterior obtenemos

$$\|e^A\| = \lim_{k \rightarrow \infty} \|S_k\| \leq e^{\|A\|} < \infty$$

y hemos probado que la serie converge absolutamente y a su suma la llamamos exponencial de la matriz A .

Veamos a continuación algunas propiedades que tiene la exponencial matricial.

Proposición 1.2 Sean $A, B \in \mathbb{C}^{n \times n}$. Se verifican las siguientes propiedades:

- 1) Si A es la matriz nula, $e^A = I$.
- 2) Si A y B conmutan, entonces A conmuta con e^B , B conmuta con e^A y e^A conmuta con e^B .
- 3) Si A y B conmutan, se cumple que $e^{A+B} = e^A e^B$.
- 4) e^A es regular y, su inversa es e^{-A} .
- 5) Si A y B son semejantes, entonces e^A es semejante con e^B , con la misma matriz de paso.

Demostración:

La demostración de 1) es trivial de la Definición 1.1. La demostración de 2) se sigue de la serie que define la exponencial y haciendo uso de la conmutatividad.

Para demostrar 3) consideremos A, B matrices de tamaño $n \times n$ tales que conmutan entre sí. Entonces

$$\begin{aligned} e^A e^B &= \left(\sum_{k=0}^{\infty} \frac{A^k}{k!} \right) \cdot \left(\sum_{k=0}^{\infty} \frac{B^k}{k!} \right) = \sum_{k=0}^{\infty} \left(\sum_{j+i=k} \frac{1}{j!i!} A^j B^i \right) \\ &= \sum_{k=0}^{\infty} \frac{1}{k!} \left(\sum_{j+i=k} \binom{k}{i} A^j B^i \right) = \sum_{k=0}^{\infty} \frac{(A+B)^k}{k!} = e^{A+B}, \end{aligned}$$

donde la penúltima igualdad es consecuencia de la conmutatividad entre A y B .

La propiedad 4) se puede probar de diferentes formas pero la más sencilla es aprovechar las propiedades 3) y 1) y utilizar que A y $-A$ conmutan.

Por último demostremos 5). Para ello supongamos que $A = PBP^{-1}$ donde P es una matriz regular $n \times n$. Entonces

$$A^k = (PBP^{-1})^k = PB^kP^{-1} \quad \forall k$$

y, en consecuencia,

$$e^A = \sum_{k=0}^{\infty} \frac{A^k}{k!} = \sum_{k=0}^{\infty} \frac{PB^kP^{-1}}{k!} = P \left(\sum_{k=0}^{\infty} \frac{B^k}{k!} \right) P^{-1} = Pe^B P^{-1}.$$

□

Proposición 1.3 Sea $A \in \mathbb{C}^{n \times n}$ una matriz diagonalizable y sean $\lambda_1, \dots, \lambda_n$ sus autovalores no necesariamente distintos. Si $P^{-1}AP = D = \text{diag}(\lambda_1, \dots, \lambda_n)$, entonces

$$e^A = P \text{diag}(e^{\lambda_1}, \dots, e^{\lambda_n}) P^{-1}.$$

($\text{diag}(\lambda_1, \dots, \lambda_n)$ y $\text{diag}(e^{\lambda_1}, \dots, e^{\lambda_n})$ denotan matrices diagonales $n \times n$ cuyos elementos diagonales son $\lambda_1, \dots, \lambda_n$ y $e^{\lambda_1}, \dots, e^{\lambda_n}$ respectivamente.)

Demostración:

Si A es diagonalizable entonces A es semejante a una matriz diagonal D , es decir, existe una matriz regular P tal que $A = PDP^{-1} = P \text{diag}(\lambda_1, \dots, \lambda_n) P^{-1}$. Utilizando la propiedad 5) de la Proposición 1.2 se tiene que

$$e^A = Pe^D P^{-1}$$

y basta notar que de (1.2) se deduce que e^D es una matriz diagonal con las exponenciales de los elementos diagonales de D en su diagonal. □

Acabamos de estudiar qué ocurre si la matriz es diagonalizable. En el caso de que no sea diagonalizable hay que acudir a la forma canónica de Jordan. Por tanto hay que conocer cómo se calcula la exponencial de una matriz de Jordan, para la cual calculamos primero la exponencial de un bloque de Jordan.

Partimos de un bloque de Jordan J_k de tamaño $d_k \times d_k$. Podemos escribirlo como

$$J_k = \lambda_k I + N,$$

donde λ_k es autovalor de A , I denota la matriz identidad del tamaño adecuado y N es nilpotente con $N^{d_k} = 0$. Ahora sustituimos A por J_k en (1.2),

$$e^{J_k} = I + J_k + \frac{J_k^2}{2} + \dots + \frac{J_k^r}{r!} + \dots$$

y obtenemos

$$\begin{aligned}
e^{J_k} &= \sum_{p=0}^{\infty} \frac{J_k^p}{p!} = \sum_{p=0}^{\infty} \frac{(\lambda_k I + N)^p}{p!} \\
&= \sum_{p=0}^{\infty} \frac{1}{p!} \sum_{l=0}^p \binom{p}{l} \lambda_k^{p-l} N^l = \sum_{l=0}^{\infty} \left(\sum_{p=l}^{\infty} \frac{1}{p!} \binom{p}{l} \lambda_k^{p-l} \right) N^l. \quad (1.3)
\end{aligned}$$

Por lo tanto, utilizando la nilpotencia de N ,

$$e^{J_k} = \sum_{l=0}^{d_k-1} \left(\sum_{p=l}^{\infty} \frac{\lambda_k^{p-l}}{(p-l)!} \right) \frac{N^l}{l!}.$$

Renombrando $m = p - l$, la expresión anterior se convierte en

$$e^{J_k} = \sum_{l=0}^{d_k-1} \left(\sum_{m=0}^{\infty} \frac{\lambda_k^m}{m!} \right) \frac{N^l}{l!} = \left(\sum_{m=0}^{\infty} \frac{\lambda_k^m}{m!} \right) \left(\sum_{l=0}^{d_k-1} \frac{N^l}{l!} \right) = e^{\lambda_k} \left(\sum_{l=0}^{d_k-1} \frac{N^l}{l!} \right).$$

Teniendo en cuenta que por la estructura de N la matriz N^l tiene todas sus entradas nulas salvo las de la superdiagonal l -ésima que valen 1, se concluye que

$$e^{J_k} = e^{\lambda_k} \begin{bmatrix} 1 & 1 & \frac{1}{2!} & \cdots & \frac{1}{(d_k-1)!} \\ 0 & 1 & 1 & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \frac{1}{2!} \\ 0 & \cdots & \ddots & 1 & 1 \\ 0 & \cdots & \cdots & 0 & 1 \end{bmatrix},$$

que coincide con

$$\begin{bmatrix} f(\lambda_k) & f'(\lambda_k) & \frac{f''(\lambda_k)}{2!} & \cdots & \frac{f^{d_k-1}(\lambda_k)}{(d_k-1)!} \\ 0 & f(\lambda_k) & f'(\lambda_k) & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \frac{f''(\lambda_k)}{2!} \\ 0 & \cdots & \ddots & f(\lambda_k) & f'(\lambda_k) \\ 0 & \cdots & \cdots & 0 & f(\lambda_k) \end{bmatrix},$$

cuando $f(z) = e^z$.

Ya estamos en condiciones de enunciar la siguiente generalización de la Proposición 1.3.

Proposición 1.4 Sea $A \in \mathbb{C}^{n \times n}$ y sea J unas de sus formas canónicas de Jordan. Entonces, si $A = PJP^{-1}$,

$$e^A = Pe^J P^{-1} \in \mathbb{C}^{n \times n}, \quad (1.4)$$

donde e^J es la matriz diagonal por bloques, cuyos bloques diagonales son iguales a las matrices e^{J_k} descritas anteriormente.

Demostración:

La demostración de este resultado hace uso de la propiedad 5) de la Proposición 1.2, y teniendo en cuenta además que la forma canónica de Jordan es diagonal por bloques y cómo es la exponencial de cada bloque de Jordan. \square

Nota: El cálculo de e^{tA} , $t \in \mathbb{R}$, suele ser más común en la práctica que el de e^A y se calcula de la misma forma teniendo en cuenta que calculamos e^B donde $B = tA$.

Veremos a continuación cómo se comporta e^{tA} bajo derivación y la hallaremos para una la matriz A de tamaño 2×2 que nos permite ilustrar las dificultades a las que hay que enfrentarse cuando se trabaja en aritmética finita.

Teorema 1.5 Sea $A \in \mathbb{C}^{m \times n}$. Se cumple

$$\frac{d}{dt} e^{tA} = A e^{tA}$$

y, en consecuencia, $x(t) = e^{tA} x_0$ es la solución del sistema diferencial de primer orden

$$x'(t) = Ax(t),$$

con condición inicial $x(0) = x_0$.

Demostración:

Para demostrar el resultado basta con derivar en la serie, ya que al ser convergente la serie que se obtiene al derivar, se puede intercambiar derivada con sumatorio,

$$\begin{aligned} \frac{d}{dt} (e^{tA}) &= \frac{d}{dt} \left(\sum_{k=0}^{\infty} \frac{t^k A^k}{k!} \right) = \sum_{k=1}^{\infty} \frac{k t^{k-1} A^k}{k!} = \sum_{k=1}^{\infty} \frac{t^{k-1} A^k}{(k-1)!} \\ &= A \left(\sum_{k=1}^{\infty} \frac{t^{k-1} A^{k-1}}{(k-1)!} \right) = A \sum_{l=0}^{\infty} \frac{t^l A^l}{l!} = A e^{tA} \end{aligned}$$

\square

Ejemplo 1.6 Consideramos la matriz

$$A = \begin{bmatrix} \lambda & \alpha \\ 0 & \mu \end{bmatrix} \in \mathbb{C}^{2 \times 2}, \quad \alpha \neq 0.$$

Si $\mu \neq \lambda$ sabemos que A es diagonalizable con autovalores λ y μ , y la matriz de cambio de base P tal que $P^{-1}AP$ es diagonal se puede tomar igual a

$$P = \begin{bmatrix} 1 & \alpha \\ 0 & \mu - \lambda \end{bmatrix}.$$

La Proposición 1.3 nos dice que

$$e^{tA} = Pe^{tD}P^{-1},$$

luego

$$e^{tA} = \begin{bmatrix} e^{t\lambda} & \alpha \frac{e^{t\lambda} - e^{t\mu}}{\lambda - \mu} \\ 0 & e^{t\mu} \end{bmatrix}. \quad (1.5)$$

Si $\mu = \lambda$, A tiene un único autovalor λ de multiplicidad 2. Sabemos que en este caso A no es diagonalizable, pues $\text{Ker}(A - \lambda I)$ está generado por el vector $(1, 0)$ y, en consecuencia, tiene dimensión 1. Para calcular su exponencial vamos a calcular su forma canónica de Jordan.

La forma canónica de Jordan está formada por un solo bloque de tamaño 2×2 . En este caso sencillo para calcular la matriz P de la forma canónica de Jordan basta con observar que podemos llevar la matriz A al bloque

$$J = \begin{bmatrix} \lambda & 1 \\ 0 & \lambda \end{bmatrix}$$

mediante transformaciones elementales por filas y por columnas. Concretamente, multiplicando la primera fila por α^{-1} y posteriormente multiplicando la primera columna de la matriz resultante por α , obtenemos

$$J = \begin{bmatrix} \alpha^{-1} & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \lambda & \alpha \\ 0 & \lambda \end{bmatrix} \begin{bmatrix} \alpha & 0 \\ 0 & 1 \end{bmatrix},$$

de donde se sigue que

$$P = \begin{bmatrix} \alpha & 0 \\ 0 & 1 \end{bmatrix} \quad \text{y} \quad J = \begin{bmatrix} \lambda & 1 \\ 0 & \lambda \end{bmatrix}.$$

En consecuencia, utilizando la Proposición 1.4 obtenemos

$$e^{tA} = Pe^{tJ}P^{-1} = \begin{bmatrix} \alpha & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} e^{t\lambda} & te^{t\lambda} \\ 0 & e^{t\lambda} \end{bmatrix} \begin{bmatrix} \frac{1}{\alpha} & 0 \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} e^{t\lambda} & \alpha te^{t\lambda} \\ 0 & e^{t\lambda} \end{bmatrix}.$$

Para ilustrar el tipo de dificultades con las que nos encontramos en la práctica cuando queremos calcular la exponencial de una matriz, consideramos la matriz (1.5) y pensamos en la implementación de los cálculos en aritmética finita. Nos damos cuenta de que tenemos una fuente potencial de error en el término $\alpha \frac{e^{t\lambda} - e^{t\mu}}{\lambda - \mu}$, pues, aunque λ y μ sean distintos, si la diferencia $\lambda - \mu$ es pequeña, al tener que dividir por dicha diferencia podríamos obtener errores grandes.

Por último, para cerrar el capítulo, incluimos a continuación dos resultados que nos permiten acotar la diferencia relativa entre las exponenciales de dos matrices próximas. Se han seguido [9] y [3] respectivamente.

Proposición 1.7 Sean $A, E \in \mathbb{C}^{n \times n}$ tales que conmutan entre sí y $t \geq 0$. Entonces si $\|\cdot\|$ denota una norma matricial derivada de una norma vectorial

$$\frac{\|e^{t(A+E)} - e^{tA}\|}{\|e^{tA}\|} \leq t\|E\|e^{t\|E\|}. \quad (1.6)$$

Demostración:

Si A y E conmutan, tA y tE también y por el apartado 5) de la Proposición 1.2 tenemos que $e^{t(A+E)} = e^{tA}e^{tE}$. En consecuencia,

$$\begin{aligned} \|e^{t(A+E)} - e^{tA}\| &= \|e^{tA}e^{tE} - e^{tA}\| = \|e^{tA} \cdot (e^{tE} - I)\| \\ &\leq \|e^{tA}\| \|e^{tE} - I\|, \end{aligned}$$

y

$$\frac{\|e^{t(A+E)} - e^{tA}\|}{\|e^{tA}\|} \leq \|e^{tE} - I\|.$$

Ahora usando que

$$e^{tE} - I = \sum_{k=1}^{\infty} \frac{(tE)^k}{k!}$$

y tomando $l = k - 1$, tenemos

$$e^{tE} - I = \sum_{l=0}^{\infty} \frac{(tE)^{l+1}}{(l+1)!} = tE \left(\sum_{l=0}^{\infty} \frac{1}{l+1} \cdot \frac{(tE)^l}{l!} \right)$$

y, por lo tanto,

$$\begin{aligned}
\frac{\|e^{t(A+E)} - e^{tA}\|}{\|e^{tA}\|} &\leq \left\| tE \left(\sum_{l=0}^{\infty} \left(\frac{1}{l+1} \right) \frac{(tE)^l}{l!} \right) \right\| \leq \|tE\| \left\| \sum_{l=0}^{\infty} \left(\frac{1}{l+1} \right) \frac{(tE)^l}{l!} \right\| \\
&\leq \|tE\| \left(\sum_{l=0}^{\infty} \left(\frac{1}{l+1} \right) \frac{\|tE\|^l}{l!} \right) \leq \|tE\| \left(\sum_{l=0}^{\infty} \frac{\|tE\|^l}{l!} \right) \\
&= \|tE\| e^{\|tE\|} = t\|E\| e^{t\|E\|}.
\end{aligned}$$

La penúltima desigualdad es consecuencia de que $\frac{1}{l+1} \leq 1$ para todo $l \geq 0$. \square

Proposición 1.8 Dadas $A, E \in \mathbb{C}^{n \times n}$ y $t \geq 0$, si $\|\cdot\|$ denota una norma matricial derivada de una norma vectorial se tiene

$$\|e^{t(A+E)} - e^{tA}\| \leq t\|E\| e^{t \max(\|A\|, \|A+E\|)}. \quad (1.7)$$

Demostración:

En primer lugar vamos a probar la siguiente igualdad válida para todo $k \geq 1$,

$$(A + E)^k - A^k = A^{k-1}E + \sum_{j=0}^{k-2} A^j E (A + E)^{k-j-1},$$

entendiendo que cuando $k = 1$ el último sumatorio es vacío. Razonaremos por inducción.

Para $k = 1$ la igualdad se convierte en $(A + E)^1 - A^1 = E$, que trivialmente es cierta.

Suponemos la igualdad cierta para $k = n$, veamos que se cumple para $k = n + 1$.

$$\begin{aligned}
(A + E)^{n+1} &= (A + E)^n (A + E) \\
&= \left(A^n + A^{n-1}E + \sum_{j=0}^{n-2} A^j E (A + E)^{n-j-1} \right) (A + E) \\
&= A^n (A + E) + A^{n-1}E (A + E) + \sum_{j=0}^{n-2} A^j E (A + E)^{n-j} \\
&= A^{n+1} + A^n E + A^{n-1}E (A + E) + \sum_{j=0}^{n-2} A^j E (A + E)^{n-j-1}.
\end{aligned}$$

Teniendo en cuenta que si $j = n - 1$, $A^j E(A + E)^{n-j} = A^{n-1} E(A + E)$, tenemos que

$$(A + E)^{n+1} = A^{n+1} + A^n E + \sum_{j=0}^{n-1} A^j E(A + E)^{(n+1)-j-1}$$

y, en consecuencia

$$(A + E)^{n+1} - A^{n+1} = A^n E + \sum_{j=0}^{n-1} A^j E(A + E)^{n-j}.$$

Para la demostración de la proposición vamos a usar la igualdad anterior.

$$\begin{aligned} e^{t(A+E)} - e^{tA} &= \sum_{k=1}^{\infty} \left(\frac{(t(A+E))^k}{k!} - \frac{(tA)^k}{k!} \right) = \sum_{k=1}^{\infty} \frac{t^k ((A+E)^k - A^k)}{k!} \\ &= \sum_{k=1}^{\infty} \frac{t^k}{k!} \left(A^{k-1} E + \sum_{j=0}^{k-2} A^j E(A+E)^{k-j-1} \right). \end{aligned}$$

Tomando normas obtenemos

$$\begin{aligned} \|e^{t(A+E)} - e^{tA}\| &= \left\| \sum_{k=1}^{\infty} \frac{t^k}{k!} \left(A^{k-1} E + \sum_{j=0}^{k-2} A^j E(A+E)^{k-j-1} \right) \right\| \\ &\leq \sum_{k=1}^{\infty} \frac{t^k}{k!} \left(\|A\|^{k-1} \|E\| + \sum_{j=0}^{k-2} \|A\|^j \|E\| \|A+E\|^{k-j-1} \right) \\ &\leq \sum_{k=1}^{\infty} \frac{t^k \|E\|}{k!} \left(\|A\|^{k-1} + \sum_{j=0}^{k-2} \|A\|^j \|A+E\|^{k-j-1} \right) \end{aligned}$$

y, si llamamos $M = \max(\|A\|, \|A+E\|)$,

$$\begin{aligned} \|e^{t(A+E)} - e^{tA}\| &\leq \sum_{k=1}^{\infty} \frac{t^k \|E\|}{k!} \left(M^{k-1} + \sum_{j=0}^{k-2} M^j M^{k-j-1} \right) \\ &= t \|E\| \sum_{k=1}^{\infty} \frac{t^{k-1}}{k!} (M^{k-1} + (k-1)M^{k-1}) \\ &= t \|E\| \sum_{k=1}^{\infty} \frac{(tM)^{k-1}}{(k-1)!} = t \|E\| e^{tM}. \end{aligned}$$

□

Capítulo 2

MÉTODOS NUMÉRICOS PARA EL CÁLCULO EFICIENTE DE LA EXPONENCIAL

Aunque en el capítulo anterior hemos visto cómo se puede hallar la exponencial de cualquier matriz en aritmética exacta, también hemos indicado que cuando se trabaja con la aritmética finita del ordenador dichos procedimientos pueden ser inadecuados ya sea porque pueden llevar a errores grandes o porque requieren un costo computacional excesivamente elevado.

En este capítulo revisaremos brevemente diversos métodos numéricos que han aparecido en la literatura para calcular la exponencial de una matriz y estudiaremos con más profundidad los dos métodos más usados: el escalado y el cuadrado, y el método de los espacios de Krylov.

Por simplicidad, todos los resultados que se van a estudiar a lo largo de este capítulo van a ser enunciados en \mathbb{R} . Sin embargo, se pueden extender con cambios mínimos a \mathbb{C} .

2.1. Diversos métodos numéricos encontrados en la literatura

A lo largo del siglo pasado han surgido diversos métodos numéricos para aproximar la exponencial matricial. Muchos de estos métodos han pasado inadvertidos desde el punto de vista práctico, pero todos

ellos tienen su importancia al menos desde el punto de vista teórico. En esta sección nos limitaremos a explicar brevemente algunos de estos métodos, agrupándolos en cinco categorías distintas atendiendo a las técnicas numéricas que involucran. Hemos seguido la descripción hecha en [6].

1) Métodos basados en la idea de aproximación

Estos métodos están basados en la aplicación directa al caso de matrices de técnicas clásicas de aproximación para la función escalar e^x . Dentro de estos métodos el primer ejemplo es el método basado en la serie de Taylor, que hace uso directo de la Definición 1.1, sumando tantos términos de la serie como sean necesarios para que al añadir un término adicional no cambie el valor de la suma ya calculada. Un segundo método está basado en la aproximación de Padé a la exponencial, que explicaremos con detalle en la Sección 2.2 puesto que, combinada con la técnica del escalado y el cuadrado, constituye uno de los métodos más comúnmente utilizados para el cálculo de la exponencial de una matriz. Es, de hecho, el método que está implementado en la función `expm` de MATLAB.

El principal problema de estos métodos no es tanto el número de términos de la serie que hay que sumar, sino las dificultades originadas por las posibles cancelaciones que pueden surgir al efectuar dichas sumas, como consecuencia de la aritmética finita del ordenador. En [6] se pueden ver algunos ejemplos.

2) Métodos basados en la integración numérica de EDO's

En el Teorema 1.5 hemos visto que la matriz e^{tA} es solución de la ecuación diferencial matricial

$$\frac{d}{dt}e^{tA} = Ae^{tA}$$

y, en particular, $e^A x_0$ es la solución en $t = 1$ del problema de valores iniciales

$$x'(t) = Ax(t), \quad x(0) = x_0. \quad (2.1)$$

Es posible, por tanto, calcular la columna k -ésima de la matriz e^A hallando la solución en $t = 1$ de (2.1), tomando como condición inicial x_0 la columna k -ésima de la matriz identidad. La solución de (2.1) se puede aproximar utilizando integradores numéricos de carácter general diseñados para sistemas diferenciales más generales

$$x'(t) = f(t, x(t)), \quad x(0) = x_0. \quad (2.2)$$

Entre los métodos numéricos para integrar (2.2) se encuentran, entre otros, los métodos de Taylor, los métodos Runge-Kutta y los métodos lineales multipaso. Varios de estos integradores están implementados en librerías de software y pueden utilizarse para aproximar e^{tA} para distintos valores de t y, en particular, para $t = 1$.

Sin embargo, este tipo de métodos no son los más apropiados para calcular la exponencial matricial pues, son excesivamente costosos ya que, por un lado, si la matriz A es grande hay que resolver muchos problemas con las distintas condiciones iniciales y, por otro, para obtener en la integración numérica de (2.1) precisiones próximas al tamaño de los errores de redondeo habría que tomar pasos de integración muy pequeños con métodos de orden muy alto.

3) Métodos basados en polinomios

El teorema de Cayley-Hamilton afirma que toda matriz es raíz de su polinomio característico y como consecuencia obtenemos que si $A \in \mathbb{R}^{n \times n}$ y $k \in \mathbb{N}$, entonces:

$$A^k = \sum_{j=0}^{n-1} \beta_{kj} A^j$$

para ciertos coeficientes $\beta_{kj} \in \mathbb{R}$. Es decir, para generar cualquier potencia de A basta combinar adecuadamente las primeras $n - 1$. Esto implica que

$$\begin{aligned} e^{tA} &= \sum_{k=0}^{\infty} \frac{t^k A^k}{k!} = \sum_{k=0}^{\infty} \frac{t^k}{k!} \left(\sum_{j=0}^{n-1} \beta_{kj} A^j \right) \\ &= \sum_{j=0}^{n-1} \left(\sum_{k=0}^{\infty} \beta_{kj} \frac{t^k}{k!} \right) A^j = \sum_{j=0}^{n-1} \alpha_j(t) A^j, \end{aligned} \quad (2.3)$$

es decir, que e^{tA} es un polinomio en A de grado $\leq n - 1$ cuyos coeficientes son funciones de t . El cálculo de dichos coeficientes se podría hacer de forma recursiva [6], pero es muy sensible a los errores de redondeo. No obstante, el principal problema del método descrito reside en que es preciso conocer el polinomio característico de la matriz A , y esto no siempre es viable en la práctica, en particular, si la matriz A es de gran tamaño. Otras alternativas basadas en reemplazar las potencias de A que aparecen en (2.3) por una base más adecuada del subespacio matricial que generan, incluyen, por ejemplo, la interpolación de Lagrange

$$e^{tA} = \sum_{j=0}^{n-1} e^{\lambda_j t} \prod_{k=1, k \neq j}^n \frac{(A - \lambda_k I)}{(\lambda_j - \lambda_k)}$$

o la forma de Newton de dicho interpolante,

$$e^{tA} = e^{\lambda_1 t} I + \sum_{j=2}^n [\lambda_1, \dots, \lambda_j] \prod_{k=1}^{j-1} (A - \lambda_k I)$$

donde $[\lambda_1, \dots, \lambda_j]$ son las diferencias divididas de orden $j - 1$ que están definidas recursivamente por

$$\begin{aligned} [\lambda_1, \lambda_2] &= \frac{e^{\lambda_1 t} - e^{\lambda_2 t}}{\lambda_1 - \lambda_2}, \\ [\lambda_1, \dots, \lambda_{k+1}] &= \frac{[\lambda_1, \dots, \lambda_k] - [\lambda_2, \dots, \lambda_{k+1}]}{\lambda_1 - \lambda_{k+1}} \quad k \geq 2. \end{aligned}$$

Sin embargo, es necesario conocer los autovalores de la matriz y, por tanto, en la práctica no son la mejor elección.

4) Métodos basados en descomposiciones matriciales

En los problemas en los que nuestra matriz A es de gran tamaño y se precisan evaluaciones de e^{tA} para distintos valores de t , los métodos que podrían parecer más eficientes son los métodos basados en distintas factorizaciones o descomposiciones de la matriz A . Además cuando A es simétrica pueden producir algoritmos muy eficientes. Estos métodos se basan en encontrar una matriz B semejante con A , para la que sea más sencillo y eficiente calcular e^{tB} . Si P es una matriz regular tal que

$$A = PBP^{-1} \tag{2.4}$$

como consecuencia de la propiedad 5) de la Proposición 1.2

$$e^{tA} = Pe^{tB}P^{-1}.$$

Un primer ejemplo de estos métodos estaría basado en el cálculo de los autovectores de la matriz A si fuese diagonalizable, o, en su defecto, en el cálculo de la forma canónica de Jordan J . Como vimos en el primer capítulo, calcular e^{tJ} es muy sencillo y barato. El problema de este método es que calcular la forma canónica de Jordan en aritmética finita no es posible, pues el mínimo error de redondeo puede provocar errores incluso en la propia estructura de los bloques diagonales que intervienen en la matriz de Jordan.

Una alternativa a la forma canónica de Jordan es la factorización de Schur que puede ser bien calculada numéricamente al involucrar

productos por matrices ortogonales. Si A es real calcula una matriz ortogonal Q y una matriz triangular (o casi triangular) T de manera que

$$Q^T A Q = T.$$

Como Q es ortogonal, $Q^{-1} = Q^T$ y obtenemos que en la diagonal de T aparecen los autovalores de la matriz A y permite calcular e^{tA} como $Q e^{tT} Q^T$.

Esta opción tiene sus inconvenientes, pues, en general, el cálculo de e^{tT} , es costoso y calcular la exponencial de una matriz triangular cuando esta es grande puede generar errores.

5) Métodos de escisión

Estos métodos se fundamentan en escribir $A = B + C$ de tal modo que sea sencillo y eficiente calcular con mucha precisión la exponencial de las matrices B y C . Aunque la igualdad

$$e^{t(B+C)} = e^{tB} e^{tC}$$

sólo es cierta si B y C conmutan (ver apartado 3) de la Proposición 1.2), las exponenciales de B y C se relacionan con la exponencial de $B + C$ mediante la igualdad

$$e^{(B+C)} = \lim_{m \rightarrow \infty} (e^{B/m} e^{C/m})^m.$$

En la literatura se han considerado escisiones con $B = \frac{(A+A^T)}{2}$ y $C = \frac{(A-A^T)}{2}$, de modo que las propiedades de simetría de B y C facilitan el cálculo de $e^{B/m}$ y $e^{C/m}$ de manera eficiente y precisa. La idea de dividir B y C por m y elevar después a la potencia m la matriz producto de las exponenciales calculadas está relacionada con la idea del escalado y cuadrado que vamos a describir en la siguiente sección.

2.2. El escalado y el cuadrado

El método del escalado y el cuadrado está basado a su vez en el método que llamamos método de la aproximación de Padé, al que ya nos hemos referido en la Sección 2.1 al hablar de los métodos basados en la idea de aproximación. Comenzamos explicando dicho método

Dada una función $f : \mathbb{R} \rightarrow \mathbb{R}$ analítica en un entorno de $x = 0$, podemos encontrar un polinomio $P(x)$ de forma que aproxime tanto como queramos a la función f . Para buscar este polinomio podemos hacer uso de valores conocidos de la función o de sus derivadas y usar las técnicas de la interpolación de Lagrange, de Taylor, de Hermite, etc... Sin embargo, en muchos casos los polinomios no son la mejor alternativa para aproximar funciones. Esto es particularmente cierto, por ejemplo, si las funciones tienen asíntotas (ya sean verticales u horizontales). Para este tipo de funciones una mejor alternativa puede ser aproximar por una función racional que consiste en un cociente de polinomios de grados p y q respectivamente.

Definición 2.1 *Dados $f : \mathbb{R} \rightarrow \mathbb{R}$ una función analítica en un entorno de $x = 0$ y $p, q \in \mathbb{N}$, llamamos aproximación (p, q) de Padé de la función f , a la función racional*

$$R_{pq}(x) = \frac{N(x)}{D(x)} = \frac{a_0 + a_1x + \cdots + a_px^p}{b_0 + b_1x + \cdots + b_qx^q}$$

que cumple

$$f(x) - R_{pq}(x) = \frac{f(x)D(x) - N(x)}{D(x)} = O(x^{p+q+1}) \text{ cuando } x \rightarrow 0. \quad (2.5)$$

Observaciones:

- 1) En la práctica, para hallar los coeficientes de $N(x)$ y $D(x)$ se aproxima previamente $f(x)$ por su polinomio de Taylor y se buscan $N(x)$ y $D(x)$ para igualar el mayor número de términos posible de dicho polinomio.
- 2) A lo sumo podemos igualar $p+q$ términos del desarrollo de Taylor de f , por tanto $k = p+q$ es el orden de la aproximación.
- 3) Si sacamos factor común a b_0 tanto en el numerador como en el denominador, la función $R_{pq}(x)$ es la misma y conseguimos eliminar uno de los parámetros del problema. En la práctica tomaremos directamente $b_0 = 1$, que es equivalente a que $D(0) = 1$.
- 4) Si tomamos $D(x) = 1$, $R(x)$ es un polinomio y tenemos una aproximación polinomial de grado p , es decir, $R_{p0}(x) = T_p(x)$, siendo $T_p(x)$ el polinomio de Taylor de grado p de f en $x = 0$. La aproximación de Padé, por tanto, es una generalización de la aproximación de Taylor.
- 5) La aproximación de Padé es la mejor aproximación a f por una función racional, fijados los grados del numerador y denominador, en el sentido de que coincide con la serie de potencias de f hasta los términos en x^{p+q} incluidos.

Para poder aplicar todo esto a e^A previamente tenemos que aplicarlo a la función $f(x) = e^x$, que es una función analítica en $x = 0$.

Teorema 2.2 *El aproximante (p, q) de Padé de $f(x) = e^x$ está dado por*

$$R_{pq}(x) = \frac{N(x)}{D(x)},$$

donde

$$N(x) = \sum_{j=0}^p \frac{(p+q-j)!p!}{(p+q)!j!(p-j)!} x^j \quad (2.6)$$

$$D(x) = \sum_{j=0}^q \frac{(p+q-j)!q!}{(p+q)!j!(q-j)!} (-x)^j \quad (2.7)$$

Demostración:

Para demostrar este teorema hemos seguido [2]. Vamos a considerar la función auxiliar $h(t)$ dada por

$$h(t) = t^q(1-t)^p,$$

donde p, q son los grados de $N(x)$ y $D(x)$, respectivamente. Es claro que $h^{(r)}(t) = 0$ si $r > p + q$ y para $r \leq p + q$ la fórmula de derivación de Leibniz para la función $h(t)$ nos dice que

$$h^{(r)}(t) = \sum_{k=0}^r \binom{r}{k} [t^q]^{(k)} [(1-t)^p]^{(r-k)}.$$

Teniendo en cuenta ahora que

$$[t^q]^{(k)} = \begin{cases} \binom{q}{k} k! t^{q-k} & \text{si } k \leq q, \\ 0 & \text{si } k > q, \end{cases} \quad (2.8)$$

y que

$$[(1-t)^p]^{(r-k)} = \begin{cases} \binom{p}{r-k} (r-k)! (1-t)^{p-r+k} & \text{si } r-k \leq p, \\ 0 & \text{si } r-k > p, \end{cases} \quad (2.9)$$

se tiene, en particular, que si $r \leq p + q$

$$h^{(r)}(0) = \begin{cases} (-1)^{r-q} \binom{p}{r-q} r! & \text{si } q \leq r \leq p + q, \\ 0 & \text{si } r < q, \end{cases} \quad (2.10)$$

y

$$h^{(r)}(1) = \begin{cases} (-1)^p \binom{q}{r-p} r! & \text{si } p \leq r \leq p+q, \\ 0 & \text{si } r < p. \end{cases} \quad (2.11)$$

Por otro lado, integrando por partes la integral

$$\int_0^1 e^{tx} h(t) dt$$

obtenemos

$$\int_0^1 e^{tx} h(t) dt = \frac{e^x h(1)}{x} - \frac{h(0)}{x} - \int_0^1 \frac{e^{tx}}{x} h^{(1)}(t) dt,$$

e integrando por partes una vez más,

$$\begin{aligned} \int_0^1 e^{tx} h(t) dt &= e^x \sum_{r=0}^1 \frac{(-1)^r h^{(r)}(1)}{x^{r+1}} - \sum_{r=0}^1 \frac{(-1)^r h^{(r)}(0)}{x^{r+1}} \\ &\quad + \frac{(-1)^{r+1}}{x^2} \int_0^1 e^{tx} h^{(2)}(t) dt. \end{aligned}$$

En general, dado un entero $s > 0$, repitiendo $s + 1$ veces la integración por partes, llegamos a la igualdad

$$\begin{aligned} \int_0^1 e^{tx} h(t) dt &= e^x \sum_{r=0}^s (-1)^r \frac{h^{(r)}(1)}{x^{r+1}} - \sum_{r=0}^s (-1)^r \frac{h^{(r)}(0)}{x^{r+1}} \\ &\quad + \frac{(-1)^{s+1}}{x^{s+1}} \int_0^1 e^{tx} h^{(s+1)}(t) dt. \end{aligned}$$

Si tomamos en la expresión anterior $s = p + q$, el integrando del último sumando se anula y multiplicando a los dos lados de dicha igualdad por el término $(-1)^s x^{s+1}$ obtenemos

$$\begin{aligned} (-1)^s x^{s+1} \int_0^1 e^{tx} h(t) dt &= (-1)^s e^x \sum_{r=0}^s (-1)^r h^{(r)}(1) x^{s-r} \\ &\quad - (-1)^s \sum_{r=0}^s (-1)^r h^{(r)}(0) x^{s-r}. \end{aligned}$$

Haciendo tender $x \rightarrow 0$ y llamando $j = s - r = p + q - r$, vemos que el término de la izquierda tiende a 0 como $O(x^{p+q+1})$ y obtenemos

$$O(x^{p+q+1}) = e^x \sum_{j=0}^s (-1)^j h^{(p+q-j)}(1) x^j - \sum_{j=0}^s (-1)^j h^{(p+q-j)}(0) x^j, \quad (2.12)$$

donde hemos usado $(-1)^{s+r} = (-1)^{j+2r} = (-1)^j$, y que $s = p + q$.
 Por último, acudiendo a (2.10) y (2.11), con $r = p + q - j$ obtenemos

$$h^{(p+q-j)}(0) = \begin{cases} (-1)^{p-j} \binom{p}{p-j} (p+q-j)! & \text{si } j \leq p, \\ 0 & \text{si } j > p, \end{cases} \quad (2.13)$$

y

$$h^{(p+q-j)}(1) = \begin{cases} (-1)^p \binom{q}{q-j} (p+q-j)! & \text{si } j \leq q, \\ 0 & \text{si } j > q. \end{cases} \quad (2.14)$$

Substituyendo en (2.12) se llega a

$$O(x^{p+q+1}) = e^x \sum_{j=0}^q (-1)^{p+j} \frac{(p+q-j)!q!}{(q-j)!j!} x^j - \sum_{j=0}^p (-1)^p \frac{(p+q-j)!p!}{(p-j)!j!} x^j.$$

Por último, multiplicando todo por la constante $\frac{(-1)^p}{(p+q)!}$ llegamos a la expresión

$$O(x^{p+q+1}) = e^x \sum_{j=0}^q \frac{(p+q-j)!q!}{(p+q)!j!(q-j)!} (-x)^j - \sum_{j=0}^p \frac{(p+q-j)!p!}{(p+q)!j!(p-j)!} x^j,$$

que se corresponde con el numerador de (2.5) para $f(x) = e^x$ con $N(x)$ y $D(x)$ dados por (2.6) y (2.7), respectivamente. El denominador satisface $D(0) = 1$. \square

Como consecuencia del teorema anterior tomamos como aproximantes de Padé de e^A ,

$$R_{pq}(A) = D(A)^{-1} \cdot N(A)$$

donde

$$N(A) = \sum_{j=0}^p \frac{(p+q-j)!p!}{(p+q)!j!(p-j)!} A^j$$

y

$$D(A) = \sum_{j=0}^q \frac{(p+q-j)!q!}{(p+q)!j!(q-j)!} (-A)^j.$$

Nótese que en la definición de $R_{pq}(A)$ aparece el término $D(A)^{-1}$. En la práctica calcular la inversa de una matriz y multiplicar por dicha inversa a una matriz dada es muy costoso, por lo que se recurre a resolver n sistemas lineales cuya matriz es $D(A)$ y los n términos independientes son las n columnas de la matriz $N(A)$.

A priori, podemos escoger los grados p y q arbitrariamente. Sin embargo, cuando la norma de la matriz no es muy grande, como en el caso en que vamos a aplicar el método, vamos a preferir los aproximantes de Padé con $p = q$, llamados *aproximantes diagonales*. La razón para ello es la siguiente; si suponemos que $p < q$ (similar para $p > q$), evaluar $R_{pq}(A)$ requiere del orden de qn^3 operaciones y el orden de la aproximación es $p + q$, mientras que evaluar $R_{qq}(A)$ requiere el mismo costo computacional y la aproximación tiene orden $2q > p + q$. Para los aproximantes diagonales se tiene que

$$N(A) = \sum_{j=0}^p \frac{(2p-j)!p!}{(p)!j!(p-j)!} A^j,$$

y $D(A) = N(-A)$, lo que los convierte además en fáciles de implementar.

El método de Padé ha sido implementado en MATLAB (ver Sección A.1 del apéndice) y para ver las ventajas que aporta este método sobre los descritos en el Capítulo 1, cuando se utiliza aritmética finita, se ha considerado la familia de matrices $A_n \in \mathbb{R}^{n \times n}$ dada por

$$A_n = \begin{bmatrix} n-1 & 1 & 0 & 0 & \cdots & 0 \\ -(n-1) & n-3 & 2 & 0 & \cdots & 0 \\ 0 & -(n-2) & n-5 & 3 & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & 0 \\ 0 & \cdots & \ddots & -2 & -(n-3) & n-1 \\ 0 & \cdots & \cdots & 0 & -1 & -(n-1) \end{bmatrix}. \quad (2.15)$$

Se ha procedido a calcular e^{A_n} para $n = 4, \dots, 128$, utilizando aproximantes de Padé con $p = q = 12$ y calculando los autovalores con la función `eig` de MATLAB y utilizando la Proposición 1.3.

En la Figura 2.1 hemos representado en escala doblemente logarítmica los errores relativos de ambas aproximaciones medidos en norma euclídea, frente a la dimensión de la matriz. Se ha utilizado el color verde para los resultados obtenidos haciendo uso de los autovalores y el color azul para los generados con la aproximación de Padé. Como solución de referencia para calcular dichos errores, hemos tomado la que proporciona la función `expm` de MATLAB. Observamos que los errores correspondientes a la aproximación de Padé son, en general, bastante inferiores a los obtenidos calculando los autovalores.

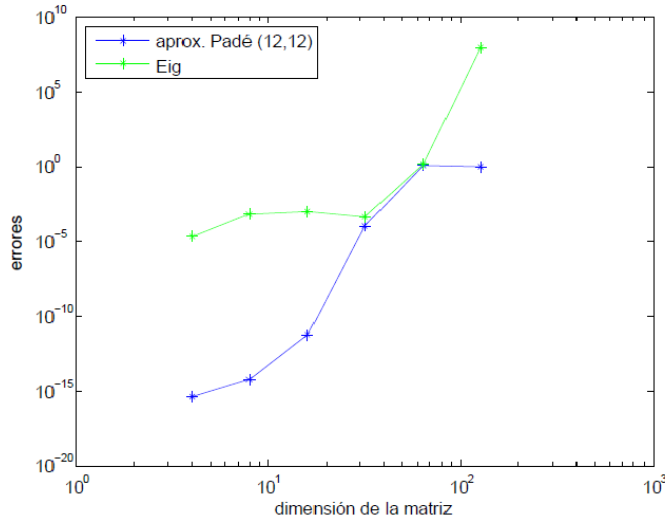


Figura 2.1: Comparación entre los errores relativos al aproximar e^{A_n} mediante aproximantes de Padé de orden 12 y utilizando el comando eig.

El mal comportamiento del procedimiento basado en el cálculo de los autovalores es debido a que éstos no son correctamente calculados cuando se trabaja en aritmética finita. La función eig de MATLAB proporciona una matriz diagonal semejante a la matriz A_n , cuyos elementos diagonales son aproximaciones al 0 con errores inferiores a 10^{-3} para $n = 4$ pero que aumenta al aumentar n , mientras que se puede comprobar que las matrices A_n dadas por (2.15) son nilpotentes de orden n y, por tanto, su único autovalor exacto es el 0 y no diagonalizan.

Observamos también en la Figura 2.1 que los errores relativos de las aproximaciones de Padé crecen al aumentar la dimensión de la matriz y son de tamaño unidad para $n \geq 64$, lo que hace que las correspondientes aproximaciones no sean de utilidad. Como ya se comentó en la sección anterior, este método de aproximación puede ser usado en la práctica cuando la norma de la matriz no es muy grande y en nuestro ejemplo, $\|A_n\|$ crece al aumentar n . Este hecho no debe sorprendernos pues la aproximación de Padé es una buena aproximación cerca del origen. La combinación de los aproximantes de Padé con la idea del escalado y el cuadrado que describimos a continuación permitirá obtener buenas aproximaciones a la función exponencial para cualquier matriz,

con independencia del valor de su norma.

Para evitar el problema que tenemos cuando la norma de la matriz es grande basta con aplicar la siguiente propiedad de la exponencial matricial

$$e^A = (e^{\frac{A}{m}})^m, \quad m \in \mathbb{N} \quad (2.16)$$

ya que $\|\frac{A}{m}\| = \frac{1}{m} \cdot \|A\| \ll \|A\|$, si m es grande.

El método del escalado y el cuadrado explota la propiedad anterior, determinando m para que la aproximación de Padé para la matriz $\frac{A}{m}$ sea eficiente y precisa. Es decir, al estar $\|\frac{A}{m}\|$ suficientemente próxima al 0, no serán necesarios p y q demasiado altos para obtener buenas aproximaciones de $e^{\frac{A}{m}}$. Posteriormente hay que elevar a la potencia m que corresponda, el resultado obtenido. En la práctica se suele escoger $m = 2^j$ de forma que

$$\frac{\|A\|}{2^j} < 1. \quad (2.17)$$

Desde el punto de vista computacional se calcula $\|A\|$, se va dividiendo sucesivamente por 2 hasta que se consigue (2.17) y se divide A por 2^j (de aquí el 'escalado' que aparece en el nombre del método). Una vez hallada $e^{\frac{A}{2^j}}$ utilizando el aproximante de Padé adecuado, para recuperar e^A se multiplica $e^{\frac{A}{2^j}}$ por sí misma para obtener $e^{\frac{A}{2^{j-1}}}$ y se itera el proceso hasta obtener e^A como producto de $e^{\frac{A}{2}}$ por sí misma (de aquí el 'cuadrado' que aparece en el nombre del método).

Este método también ha sido implementado en MATLAB (ver Sección A.2 del apéndice) y se ha llevado a cabo la comparación de este método con el de la aproximación de Padé para las matrices A_n definidas en (2.15).

La Figura 2.2 muestra los errores relativos obtenidos en la aproximación de Padé $R_{12,12}$ (azul) y los que se obtienen cuando se utiliza el escalado y el cuadrado (rojo). Se observa que estos últimos son mucho menores que los correspondientes a la aproximación de Padé y que se mantienen prácticamente del tamaño de los errores de redondeo para todas las dimensiones consideradas. Hay que notar que la línea azul es la misma que se observa en la Figura 2.1, aunque con un considerable cambio de escala.

En la Figura 2.3 se ha comparado el costo computacional (medido en tiempo de CPU) de los dos procedimientos. Aunque el escalado y el cuadrado requiere un poco más de esfuerzo computacional, la diferencia entre ambos métodos no es grande y, en conclusión, el método

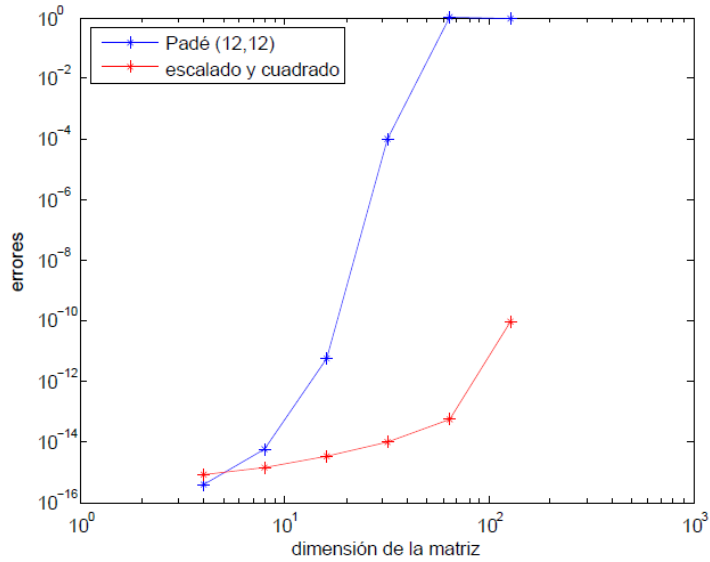


Figura 2.2: Errores relativos al aproximar e^{A_n} mediante aproximantes de Padé de orden (12, 12) y el método del escalado y el cuadrado.

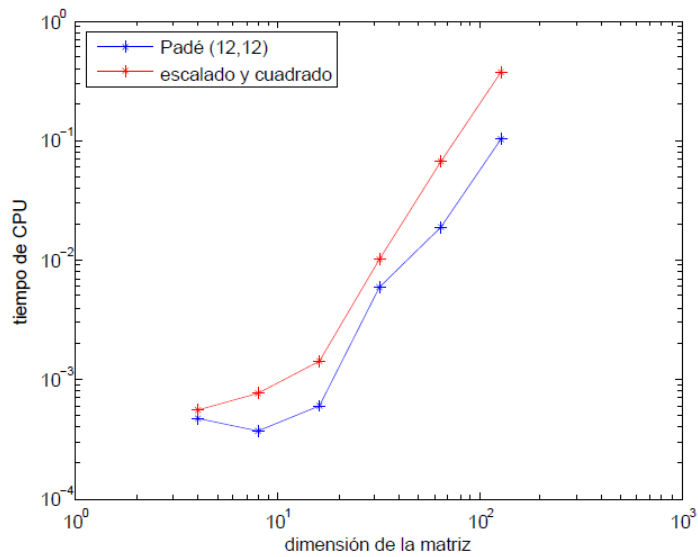


Figura 2.3: Tiempos de CPU al aproximar e^{A_n} mediante aproximantes de Padé de orden (12, 12) y el método del escalado y el cuadrado.

del escalado y el cuadrado cumple su objetivo de mejorar al método de Padé y conseguir buenos resultados de aproximación, lo que hacen de él un método muy atractivo en la práctica.

2.3. Método de los espacios de Krylov.

Si queremos resolver el sistema de ecuaciones diferenciales lineales

$$x'(t) = Ax(t),$$

con condición inicial $x(t_0) = x_0$ y matriz A de tamaño $n \times n$, debemos calcular

$$e^{(t-t_0)A}x_0. \quad (2.18)$$

Si la matriz A es dispersa y de gran tamaño, el cálculo de la matriz exponencial de $(t - t_0)A$ además de ser muy costoso va a producir una matriz, en general, llena, por la que hay que multiplicar después a la condición inicial. Será, por tanto, preferible utilizar un método que proporcione directamente el producto (2.18) y que aproveche el carácter disperso de la matriz A . Para ello vamos a estudiar el método basado en los subespacios de Krylov. Aunque en un principio queremos estudiar este tipo de métodos para la exponencial, podría utilizarse también para otras funciones matriciales.

Dada una función $f(x)$, $x \in \mathbb{R}$, el método de los subespacios de Krylov permite aproximar directamente el producto $f(A)b$, donde $b \in \mathbb{R}^n$, sin hallar previamente la matriz $f(A)$.

Nótese que en nuestro caso estamos interesados en $f(A) = e^A$. Si consideramos $f(A) = A^{-1}$, hallar $f(A)b$ sería equivalente a resolver el sistema lineal $Ax = b$ sin calcular A^{-1} , por lo que este método, además de permitirnos aproximar la exponencial también puede ser tratado como un método de aproximación de soluciones de grandes sistemas dispersos de ecuaciones lineales.

Para poder estudiar este método, previamente debemos introducir qué son los espacios de Krylov y estudiar el *proceso de Arnoldi*.

Definición 2.3 Sean $A \in \mathbb{R}^{n \times n}$, $b \in \mathbb{R}^n$ no nulo y $k \in \mathbb{N}$. Definimos el k -ésimo subespacio de Krylov como el subespacio vectorial de \mathbb{R}^n generado por los vectores $b, Ab, \dots, A^{k-1}b$, y lo denotamos por

$$\mathcal{K}_k(A, b) = \langle b, Ab, A^2b, \dots, A^{k-1}b \rangle. \quad (2.19)$$

En virtud del Teorema de Cayley-Hamilton, A es raíz de su polinomio característico, es decir, si $p(x) = x^n + a_{n-1}x^{n-1} + \dots + a_1x + a_0$ es el polinomio característico de A , entonces $A^n = -a_{n-1}A^{n-1} - \dots - a_1A - a_0I$. En consecuencia $A^nb \in \mathcal{K}_n(A, b)$ y para $m > n$, $A^mb \in \mathcal{K}_n(A, b)$. No tiene sentido, por tanto, considerar en (2.19) valores $k > n$.

En la práctica, como n será grande, queremos considerar valores de k mucho menores que n .

Teorema 2.4 Sean $A \in \mathbb{R}^{n \times n}$ y $b \in \mathbb{R}^n$ tal que $b \neq 0$. Si denotamos por $q_{A,b}$ al único polinomio mónico de grado mínimo m , tal que $q_{A,b}(A)b = 0$, entonces m es el mínimo entero positivo tal que

$$\mathcal{K}_m(A, b) = \mathcal{K}_{m+1}(A, b). \quad (2.20)$$

Demostración:

Primero vamos a ver que efectivamente si m es el grado de $q_{A,b}$ entonces se cumple (2.20).

Si $q_{A,b} = x^m + a_{m-1}x^{m-1} + \dots + a_1x + a_0$, como $q_{A,b}(A)b = 0$, entonces

$$(A^m + a_{m-1}A^{m-1} + \dots + a_1A + a_0I)b = 0$$

y, en consecuencia,

$$A^mb = - \sum_{j=0}^{m-1} a_j A^j b \in \mathcal{K}_m(A, b).$$

Se tiene por tanto que

$$\mathcal{K}_{m+1}(A, b) \subseteq \mathcal{K}_m(A, b),$$

y como por la construcción de los subespacios de Krylov se tiene que $\mathcal{K}_m(A, b) \subseteq \mathcal{K}_{m+1}(A, b)$, queda probada la igualdad.

Veamos ahora que m es el mínimo entero positivo con la propiedad (2.20). Para ello vamos a razonar por reducción al absurdo y supongamos que existe $s < m$ de forma que $\mathcal{K}_s(A, b) = \mathcal{K}_{s+1}(A, b)$. Entonces $A^s b \in \mathcal{K}_s(A, b)$ y necesariamente deben existir escalares c_0, \dots, c_{s-1} tales que

$$A^s b = \sum_{j=0}^{s-1} c_j A^j b.$$

Esto implica que

$$\left(A^s - \sum_{j=0}^{s-1} c_j A^j \right) b = 0$$

y, en consecuencia, existe un polinomio mónico, $\tilde{q}_{A,b}$ de grado $s < m$ con $\tilde{q}_{A,b}(A)b = 0$. Esto contradice el hecho de que $q_{A,b}$ era el polinomio mónico de grado mínimo m con la propiedad $q_{A,b}(A)b = 0$. \square

A continuación debemos detallar el *proceso de Arnoldi* que será la base del método basado en subespacios de Krylov.

Definición 2.5 Dada $H \in \mathbb{R}^{n \times n}$, se dice que H tiene forma de Hessenberg superior si $h_{i,j} = 0$ siempre que $i > j + 1$, es decir, si es de la forma

$$H = \begin{bmatrix} h_{1,1} & h_{1,2} & h_{1,3} & \cdots & h_{1,n} \\ h_{2,1} & h_{2,2} & h_{2,3} & \cdots & h_{2,n} \\ 0 & h_{3,2} & h_{3,3} & \cdots & h_{3,n} \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & h_{n,n-1} & h_{n,n} \end{bmatrix}.$$

Es conocido [10], que dada $A \in \mathbb{R}^{n \times n}$ siempre se puede encontrar una matriz $Q \in \mathbb{R}^{n \times n}$ ortogonal de forma que

$$Q^T A Q = H, \quad (2.21)$$

donde H tiene forma de Hessenberg superior. Se dice que A se ha reducido a forma de Hessenberg superior mediante transformaciones ortogonales. Dada $A \in \mathbb{R}^{n \times n}$ el proceso de Arnoldi calcula su reducción a forma de Hessenberg.

Para describir el método vamos a reescribir la relación (2.21) en el formato equivalente $AQ = QH$, por ser Q ortogonal.

Si denotamos por q_1, \dots, q_n los vectores columna de Q y reescribimos en función de éstos la relación anterior, obtenemos

$$Aq_k = \sum_{i=1}^{k+1} h_{ik} q_i, \quad 1 \leq k \leq n-1. \quad (2.22)$$

Supongamos ahora que $h_{k+1,k} \neq 0$, $1 \leq k \leq n-1$. De la relación (2.22) se concluye el siguiente resultado.

Teorema 2.6 En las condiciones anteriores, para $1 \leq k \leq n-1$ se tiene

$$\langle q_1, \dots, q_k \rangle = \langle q_1, Aq_1, A^2q_1, \dots, A^{k-1}q_1 \rangle.$$

Demostración:

Razonemos por inducción sobre k .
Para $k = 1$ claramente se tiene la relación,

$$\langle q_1 \rangle = \langle q_1 \rangle.$$

Supongamos cierta la identidad para $k = s$ y veamos que se cumple para $k = s + 1$.

Como, por hipótesis de inducción,

$$\langle q_1, \dots, q_s \rangle = \langle q_1, Aq_1, A^2q_1, \dots, A^{s-1}q_1 \rangle,$$

basta probar que $q_{s+1} \in \langle q_1, Aq_1, A^2q_1, \dots, A^s q_1 \rangle$ y que $A^s q_1 \in \langle q_1, \dots, q_{s+1} \rangle$.
De la relación (2.22) se tiene

$$h_{s+1,s}q_{s+1} = Aq_s - \sum_{i=1}^s h_{is}q_s.$$

El sumatorio claramente está en $\langle q_1, \dots, q_s \rangle = \langle q_1, Aq_1, A^2q_1, \dots, A^{s-1}q_1 \rangle$ y $Aq_s \in \langle Aq_1, A^2q_1, \dots, A^s q_1 \rangle \subseteq \langle q_1, Aq_1, A^2q_1, \dots, A^s q_1 \rangle$, luego al ser $h_{s+1,s} \neq 0$,

$$q_{s+1} \in \langle q_1, Aq_1, A^2q_1, \dots, A^s q_1 \rangle.$$

Por otro lado, como $A^{s-1}q_1 \in \langle q_1, \dots, q_s \rangle$ entonces $A^s q_1 = A(A^{s-1}q_1) \in \langle Aq_1, \dots, Aq_s \rangle$. Utilizando de nuevo (2.22) concluimos que

$$Aq_l \in \langle q_1, \dots, q_{l+1} \rangle \text{ para } 1 \leq l \leq s,$$

lo cual implica que $A^s q_1 \in \langle q_1, \dots, q_{s+1} \rangle$. \square

A los vectores q_1, \dots, q_k se los conoce como *vectores de Arnoldi*, que por ser Q ortogonal forman una base ortonormal de $\mathcal{K}_k(A, q_1)$.

Volviendo de nuevo a (2.22), si despejamos el último término de la suma obtenemos

$$h_{k+1,k}q_{k+1} = Aq_k - \sum_{i=1}^k h_{ik}q_i, \quad 1 \leq k \leq n-1. \quad (2.23)$$

Para que q_{k+1} sea ortogonal a q_1, \dots, q_k tienen que ser $h_{ik} = (Aq_k)^T q_i$, para todo i .

Definiendo

$$r_k = Aq_k - \sum_{i=1}^k h_{ik}q_i$$

entonces, si $r_k = 0$, $Aq_k \in \langle q_1, \dots, q_k \rangle$, $\mathcal{K}_{k+1}(A, q_1) = \mathcal{K}_k(A, q_1)$ y $\{q_1, \dots, q_k\}$ constituyen una base ortonormal de $\mathcal{K}_{k+1}(A, q_1)$, mientras que en el caso en que $r_k \neq 0$, $h_{k+1,k} = \|r_k\|_2$ y $q_{k+1} = \frac{r_k}{h_{k+1,k}}$.

De este modo se calculan los k primeros vectores de Arnoldi que por (2.23) producen la factorización

$$AQ_k = Q_k H_k + h_{k+1,k} q_{k+1} e_k^T, \quad (2.24)$$

donde $Q_k = [q_1, \dots, q_k]$ y H_k es Hessemberg superior de orden $k \times k$ y está formada por los elementos de las k primeras filas y columnas de H .

Multiplicando por la izquierda por Q_k^T y teniendo en cuenta que q_{k+1} es ortogonal a las columnas de Q_k se tiene que

$$Q_k^T A Q_k = H_k + h_{k+1,k} Q_k^T q_{k+1} e_k^T = H_k.$$

Si el proceso anterior se repite hasta $k = n$ llegamos a la matriz H de (2.21). Sin embargo, en virtud del Teorema 2.4 si m es el grado mínimo del polinomio mónico $q_{A,b}$ tal que $q_{A,b}(A)b = 0$ entonces el proceso de Arnoldi con $q_1 = \frac{b}{\|b\|_2}$ termina para $k = m$, donde obtendríamos que $r_m = 0$ y a lo sumo deberíamos calcular m vectores.

Observaciones:

- 1) Para cada k , a la vista de la igualdad $Q_k^T A Q_k = H_k$, decimos que la matriz H_k es la proyección de A sobre el subespacio de Krylov $\mathcal{K}_k(A, q_1)$.
- 2) En general, no es cierto que, $Q_k H_k Q_k^T = A$ para cada k . En efecto, tomemos (2.24) y multipliquemos por Q_k^T por la derecha, entonces

$$A = Q_k H_k Q_k^T + h_{k+1,k} q_{k+1} e_k^T Q_k^T = Q_k H_k Q_k^T + h_{k+1,k} q_{k+1} q_k^T.$$

El segundo sumando de la expresión anterior solo es 0 cuando $h_{k+1,k} = 0$, es decir, cuando $k = m$ y el proceso termina. En este caso $A = Q_m H_m Q_m^T$.

En resumen, podemos dar el siguiente algoritmo:

Dada $A \in \mathbb{R}^{n \times n}$ y $q_1 \in \mathbb{R}^n$ unitario, el siguiente algoritmo usa el proceso de Arnoldi para calcular la factorización $AQ_m = Q_m H_m$, donde $Q_m \in \mathbb{R}^{n \times m}$ ($m = \deg(q_{A,q_1})$) tiene columnas ortonormales y $H_m \in \mathbb{R}^{m \times m}$ es Hessemberg superior con todos los elementos subdiagonales distintos de 0.

```

1: for  $k = 1 : n$  do
2:    $z = Aq_k$ 
3:   for  $i = 1 : k$  do
4:      $h_{ik} = q_i^t z$ 
5:      $z = z - h_{ik}q_i$ 
6:   end for
7:    $h_{k+1,k} = \|z\|_2$ 
8:   if  $h_{k+1,k} = 0$  then
9:      $m = k$ , break
10:  end if
11:   $q_{k+1} = \frac{z}{h_{k+1,k}}$ 
12: end for

```

Observaciones:

- 1) El algoritmo anterior no es más que el algoritmo de Gram-Schmidt modificado para ortogonalizar Aq_k frente a los vectores q_1, \dots, q_k , aunque se podría usar el algoritmo clásico de Gram-Schmidt.
- 2) No es necesario conocer explícitamente la matriz A . Basta con poder calcular los productos Aq_k . Esto tiene especial relevancia cuando se trabaja con matrices dispersas.
- 3) Si la matriz A es simétrica muchos de los productos del proceso no hacen falta ser calculados por lo que el costo operativo se reduce y obtenemos el llamado *Proceso de Lanczos*. En este caso obtenemos que H_m es tridiagonal simétrica.

Ya hemos explicado en qué consiste el proceso de Arnoldi y su objetivo, luego estamos en condiciones de describir *el método para calcular $f(A)b$ basado en los subespacios de Krylov*, que no es más que aprovechar el proceso de Arnoldi. En estos métodos tratamos de proyectar un "problema m -dimensional" en subespacios de dimensión menor sobre los que es más sencillo y económico realizar los cálculos. El Teorema 2.4 nos aseguró que en algunos casos al realizar la proyección sobre estos subespacios vamos a tener esa reducción de dificultad del problema.

Dados $A \in \mathbb{R}^{n \times n}$ y $b \in \mathbb{R}^n$ con $b \neq 0$, tomamos $q_1 = \frac{b}{\|b\|_2}$ y realizamos el proceso de Arnoldi para calcular para cierto k , $H_k \in \mathbb{R}^{k \times k}$ y $Q_k \in \mathbb{R}^{n \times k}$ tales que $Q_k^T A Q_k = H_k$. Como aproximación a $f(A)b$ tomamos entonces

$$f_k b := Q_k f(H_k) Q_k^T b = \|b\|_2 Q_k f(H_k) e_1, \quad (2.25)$$

donde para obtener la segunda igualdad hemos tenido en cuenta que la

primera columna de Q_k es proporcional a b y que entonces $Q_k^T b = \|b\|_2 e_1$. En (2.25) estamos evaluando f en el subespacio de Krylov $\mathcal{K}_k(A, q_1)$ y posteriormente estamos extendiendo el resultado al espacio original \mathbb{R}^n . Claramente cuando $k = n$ obtenemos el valor exacto, pues al ser Q_n ortogonal, si f está definida por una serie de potencias convergente,

$$f(A)b = f(Q_n H_n Q_n^T) b = Q_n f(H_n) Q_n^T b.$$

Sin embargo, el proceso de Arnoldi acaba cuando $k = m = \deg(q_{A, q_1})$ por lo que vamos a ver que en este caso efectivamente seguimos obteniendo aproximaciones exactas.

Proposición 2.7 *Sea $A \in \mathbb{R}^{n \times n}$ y sean Q_k y H_k las matrices resultantes del proceso de Arnoldi en k pasos. Entonces, para todo polinomio $p(x)$ de grado $\leq k - 1$ tenemos una aproximación exacta, es decir, se cumple que*

$$p(A)q_1 = Q_k p(H_k) e_1.$$

Demostración:

Basta probar que se tiene aproximación exacta para las potencias A^j de grado $j \leq k - 1$. Para ello procederemos por inducción. Para $j = 0$, entonces claramente es cierto porque $q_1 = Q_k e_1$. Supongamos cierta la propiedad para $j - 1 \leq k - 2$ y veamos que se cumple para j .

$$\begin{aligned} A^j q_1 &= A \cdot (A^{j-1} q_1) = A Q_k H_k^{j-1} e_1 \\ &= (Q_k H_k + h_{k+1, k} q_{k+1} e_k^T) H_k^{j-1} e_1 \\ &= Q_k H_k^j e_1 + h_{k+1, k} q_{k+1} e_k^T H_k^{j-1} e_1 \\ &= Q_k H_k^j e_1, \end{aligned}$$

donde se ha hecho uso de la igualdad (2.24) y de que, por ser H_k de Hessenberg, la matriz H_k^{j-1} tiene ceros en la primera columna a partir del elemento $j + 1$ y al ser $j + 1 \leq k$ hace que $e_k^T H_k^{j-1} e_1 = 0$. \square

La Proposición 2.7 implica que si el proceso de Arnoldi se para tras k pasos, tenemos aproximación exacta para todos los polinomios de grado $\leq k$ cuando usamos el método basado en los subespacios de Krylov.

Teorema 2.8 *Sea $f(x)$ una función que puede escribirse como serie de potencias convergente para todo $x \in \mathbb{R}$ y $A \in \mathbb{R}^{n \times n}$. La expresión (2.25) da una aproximación exacta cuando $k = m$, donde m es el grado del polinomio mínimo $q_{A, b}$ tal que $q_{A, b}(A)b = 0$.*

Demostración:

Consideremos $f(x) = \sum_{j=0}^{\infty} \alpha_j x^j$ una serie de potencias convergente para todo $x \in \mathbb{R}$. Sean $A \in \mathbb{R}^{n \times n}$, $b \in \mathbb{R}^n$ no nulo y m el paso para el cual el proceso de Arnoldi concluye. Sabemos que para m se tiene que

$$A = Q_m H_m Q_m^T,$$

luego

$$\begin{aligned} f(A) &= \sum_{j=0}^{\infty} \alpha_j A^j = \sum_{j=0}^{\infty} \alpha_j (Q_m H_m Q_m^T)^j = \sum_{j=0}^{\infty} \alpha_j Q_m H_m^j Q_m^T \\ &= Q_m \left(\sum_{j=0}^{\infty} \alpha_j H_m^j \right) Q_m^T = Q_m f(H_m) Q_m^T \end{aligned}$$

y en consecuencia

$$f(A)b = Q_m f(H_m) Q_m^T b = f_m b.$$

□

El Teorema 2.8 nos da un resultado para funciones que se pueden escribir como serie de potencias convergente, lo que le resta generalidad al estudio que estamos realizando. Sin embargo, esto es suficiente para nuestros propósitos, pues la exponencial es una función de las mencionadas y como consecuencia el siguiente corolario está demostrado.

Corolario 2.9 *Dados $A \in \mathbb{R}^{n \times n}$ y $b \in \mathbb{R}^n$ no nulo, la expresión (2.25) produce aproximaciones exactas cuando $f(A) = e^A$ y $k = m$, donde m es el grado del polinomio mínimo $q_{A,b}$ tal que $q_{A,b}(A)b = 0$.*

Observaciones:

1) En nuestro caso, cuando k tiene un tamaño adecuado, es buena idea computar e^{H_k} mediante el método del escalado y el cuadrado ya estudiado en la sección anterior. Cuando lo anterior no es posible existen diversos métodos para aproximar el factor $e^{H_k} e_1$, que es realmente el que necesitamos conocer en la aproximación. La precisión de la aproximación depende de k .

2) Aunque dados $A \in \mathbb{R}^{n \times n}$ y $b \in \mathbb{R}^n$, estos métodos aproximen $f(A)b$, siempre podemos tomar $b = e_i$ para cada $i \in \{1, \dots, n\}$ y recuperar la matriz $f(A)$, lo que nos da un método para la aproximación de $f(A)$, aunque, en general, no va a ser muy eficiente.

3) Si no se requieren aproximaciones muy precisas de $e^A b$, es suficiente aproximar e^A , por su polinomio de Taylor de grado $\leq k$ e iterar el proceso de Arnoldi sólo k veces. En ese caso se puede demostrar [8] que si $\rho = \|A\|_2$ y $\beta = \|b\|_2$, entonces

$$\|e^A b - \beta Q_k e^{H_k} e_1\|_2 \leq 2\beta \frac{\rho^k e^\rho}{k!},$$

de manera que podemos estimar el grado k del polinomio de Taylor que necesitamos considerar para obtener una precisión determinada.

Hemos programado el método basado en los subespacios de Krylov en MATLAB y hemos llevado a cabo una serie de experimentos.

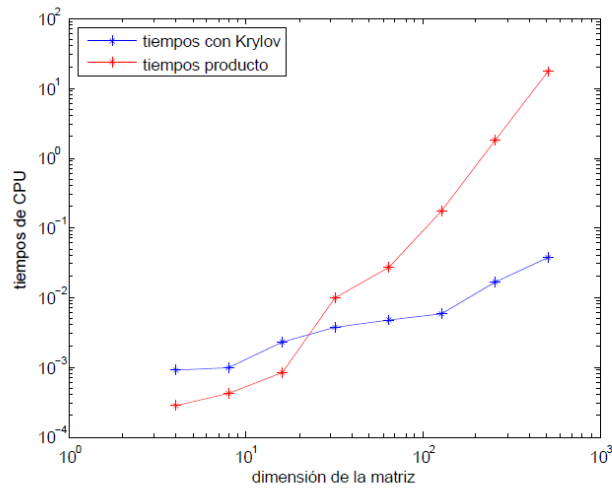


Figura 2.4: Comparación entre el tiempo de CPU para el método de Krylov y el producto directo cuando se calcula $e^A b$

En el primero se han generado matrices aleatorias A_n de tamaño $n \times n$ con $n = 4, 8, \dots, 256$ y vectores aleatorios $b_n \in \mathbb{R}^n$ para los mismos valores de n y se ha calculado $e^{A_n} b_n$ utilizando subespacios de Krylov parando el proceso de Arnoldi con $k = 20$ y calculando e^{A_n} con el método del escalado y el cuadrado y después multiplicando la exponencial calculada por b_n .

En la Figura 2.4 se han representado en escala doblemente logarítmica los tiempos de CPU utilizados frente a la dimensión del problema. Observamos que para dimensiones pequeñas el método del escalado es menos costoso, pero al aumentar la dimensión, el método basado en el proceso de Arnoldi es mucho más eficiente.

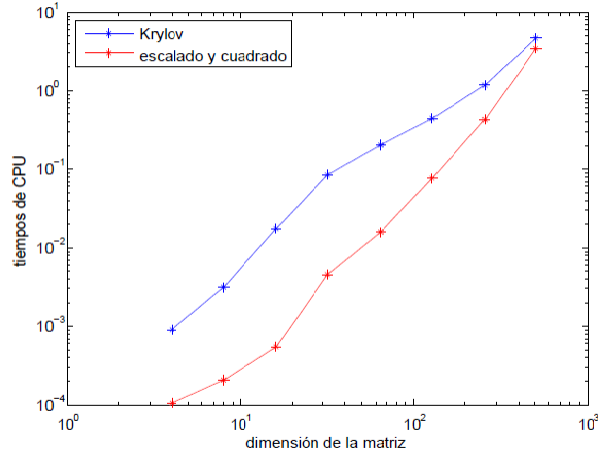


Figura 2.5: Comparación entre el tiempo de CPU para los método de Krylov y del escalado y el cuadrado cuando se calcula e^A

Se ha tomado $k = 20$ porque de ese modo los errores relativos en la aproximación a $e^{A_n}b_n$ con ambos procedimientos son del mismo tamaño y se mantienen por debajo de 5×10^{-13} .

El segundo experimento que hemos realizado consiste en hallar la matriz e^{A_n} , $n = 4, 8, \dots, 256$, ejecutando el método basado en los subespacios de Krylov n veces tomando en cada ejecución como vector b , la j -ésima columna de la matriz identidad de tamaño $n \times n$. Se ha utilizado el mismo valor $k = 20$, y se han representado en la Figura 2.5 los tiempos de CPU utilizados por este procedimiento y por el método del escalado y el cuadrado frente a la dimensión de la matriz. Los errores en ambos métodos son comparables, y observamos que como método de aproximación de la matriz exponencial, el método basado en los espacios de Krylov es menos eficiente que el escalado y el cuadrado, por lo que el método no es bueno para aproximar la exponencial.

Capítulo 3

APLICACIONES

En este último capítulo vamos a describir un par de situaciones donde los métodos explicados en el Capítulo 2 son de aplicación, y vamos a resolver algunos ejemplos de las mismas mediante los programas implementados en MATLAB (ver apéndice).

3.1. Propiedades de redes de comunicación

La primera aplicación está relacionado con el estudio de algunas propiedades de redes de comunicación o interconexión.

Al modelar problemas de las matemáticas en los que intervienen redes de comunicación recurrimos a los grafos para representar las conexiones entre los nodos de las mismas. Esto sucede por ejemplo, cuando estudiamos la World Wide Web, cuyos nodos son las paginas Web, al estudiar las interrelaciones entre los animales de un zoo o cuando estudiamos las conexiones de nuestro cerebro, entre otros. A lo largo de esta sección vamos a seguir [5] y [1] para revisar algunos conceptos relacionados con el grafo que representa a una red y mostrar el papel que juega la exponencial de la matriz de adyacencia de dicho grafo.

Definición 3.1 *Un grafo G se define como un par (V, E) , donde $V = \{1, \dots, n\}$ es el conjunto de vértices o nodos y E es un subconjunto de pares no ordenados de nodos que reciben el nombre de aristas. Dado un grafo llamamos matriz de adyacencia del mismo a una matriz $A \in \mathbb{R}^{n \times n}$, simétrica, tal que $a_{ij} = 1$ si el nodo i está conectado con el nodo j ($(i, j) \in E$), y $a_{ij} = a_{ji} = 0$ en caso contrario. Se tomará $a_{ii} = 0$, para $1 \leq i \leq n$.*

Al estudiar un grafo y el funcionamiento de la red asociada al mismo, conocer el nodo con mas conexiones puede ayudar a determinar cuál es el nodo que posee mayor importancia, es decir, cual es el que tiene mas influencia en el paso de información a través de él.

Dado un nodo, la forma más simple de medir su importancia es contar el número de conexiones que tiene. Esto es lo que se conoce como *grado* de un nodo y se calcula trivialmente como

$$deg_i = \sum_{k=1}^n a_{ik} = (A \cdot \mathbf{1})_i,$$

donde $\mathbf{1} = (1)_{1 \leq i \leq n}$ y $(\cdot)_i$ denota la componente i -ésima.

Aunque el grado nos da una forma de medir la importancia de un nodo basándose en el número de conexiones que posee, pudiera ser que un nodo tuviera pocas conexiones, pero que los nodos a los que está conectado tuviesen un grado alto, por lo que, indirectamente, la importancia del primer nodo podría ser bastante mayor de lo que indica el valor de su grado. Entra en juego el concepto de camino entre dos nodos que se define del siguiente modo.

Definición 3.2 *Llamamos camino de longitud s entre dos nodos i y j a un conjunto de nodos $\{i, k_1, \dots, k_{s-1}, j\}$ todos distintos y de forma que cada uno está conectado con el siguiente, es decir,*

$$a_{ik_1} = \dots = a_{k_{s-1}j} = 1,$$

o, equivalentemente, $(i, k_1), (k_1, k_2), \dots, (k_{s-1}, j) \in E$.

En [1] se define la *centralidad* de un nodo como el número total de caminos que podemos tomar con origen en i y llegada a él mismo. Como

$$(A^s)_{ij} = \sum_{k_1=1}^n \dots \sum_{k_{s-1}=1}^n a_{ik_1} \dots a_{k_{s-1}j},$$

la diagonal de A^s es el número de caminos cerrados de longitud s que podemos tomar con origen en i . El número total de caminos cerrados de cualquier longitud con origen y final en el nodo i se puede calcular como el elemento diagonal i -ésimo de la matriz

$$\sum_{k=0}^{\infty} A^k.$$

Teniendo en cuenta que la información fluye mejor en los caminos cortos que en los largos (el riesgo de que se pierda es menor), podemos dar una serie de pesos a los distintos caminos en función de su longitud y si dichos pesos se eligen como $\frac{1}{s!}$ para los caminos de longitud s [1], la centralidad del nodo i se puede calcular como

$$(e^A)_{ii}.$$

En caso de que dos nodos tengan el mismo grado, la centralidad nos puede servir para deshacer el empate y saber cuál de los dos tiene mayor influencia en la red.

La idea de contar caminos se puede extender al caso de nodos $i, j \in V$ distintos y da lugar al concepto de *comunicabilidad* entre dichos nodos, que tiene en cuenta el número total de caminos que podemos tomar de uno a otro. Siguiendo los mismos argumentos que para la centralidad, la comunicabilidad entre los nodos i y j se puede calcular como [1]

$$(e^A)_{ij}.$$

Por último podemos estudiar una última propiedad de una red con la ayuda de la exponencial matricial. Si fijamos los nodos i, j, r , estudiaremos la *influencia* del nodo r en el paso de la información entre i y j . Es razonable preguntarse cómo cambia la comunicabilidad entre i y j cuando prescindimos del nodo r . Considerando la matriz $A - B(r)$, donde $B(r)$ es una matriz del mismo tamaño que A , que tiene entradas no nulas solo en la fila y la columna r y que tiene 1's donde los tiene A , el cambio relativo de la comunicabilidad se puede escribir como

$$\frac{1}{(N-1)^2 - (N-1)} \sum \sum_{i \neq j, i \neq r, j \neq r} \frac{((e^A)_{ij} - (e^{A-B(r)})_{ij})}{(e^A)_{ij}},$$

donde $(N-1)^2 - (N-1)$ es el número de términos totales de la suma.

La matriz $A - B(r)$ representa la matriz de adyacencia del grafo que se obtiene al eliminar todas las conexiones del nodo r en el grafo original.

En [1], además de la función exponencial, se han considerado también otras funciones matriciales, que corresponden a tomar pesos distintos de $\frac{1}{s!}$ para los caminos de longitud s .

Para cerrar la sección vamos a estudiar dos ejemplos en los que aplicaremos todo lo anterior utilizando los métodos numéricos estudiados

a lo largo de la Sección 2.2 para el cálculo de la exponencial de una matriz.

Como primer ejemplo vamos a considerar el representado por la matriz de adyacencia de la Figura 3.1, obtenida mediante el comando `spy` de MATLAB, y que se corresponde con el estudio de relaciones sociales entre 15 trabajadores mineros en Zambia, uno de los ejemplos estudiados en [1].

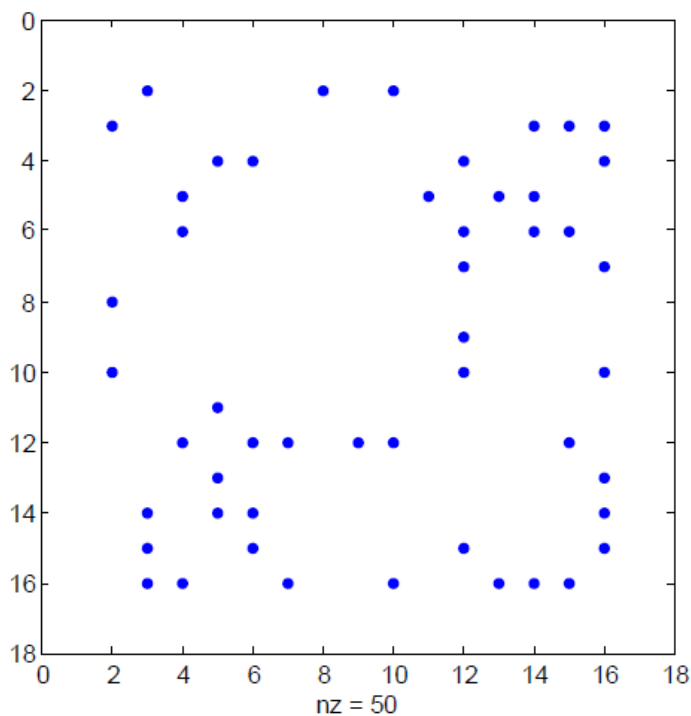


Figura 3.1: Matriz de adyacencia del primer ejemplo.

Aproximamos e^A mediante el método del escalado y el cuadrado y representamos en la Figura 3.2 la centralidad de los nodos en azul y el grado de los mismos en color rojo.

En la Figura 3.2 podemos ver cómo el grado de los nodos sí nos da una idea aproximada de qué nodo es el más importante, que en nuestro caso sería el nodo 12, que tiene 6 conexiones. Sin embargo, se producen empates por ejemplo entre los nodos 5 y 6 y necesariamente tenemos que recurrir a la centralidad de dichos nodos para discernir cuál es el

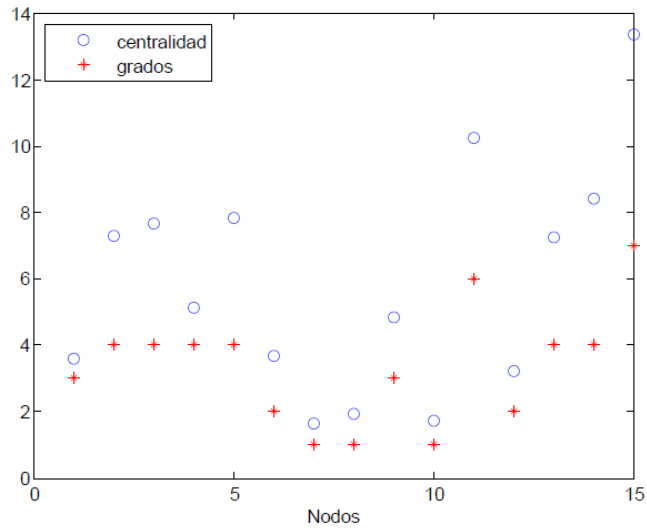


Figura 3.2: Centralidad y grado de cada nodo.

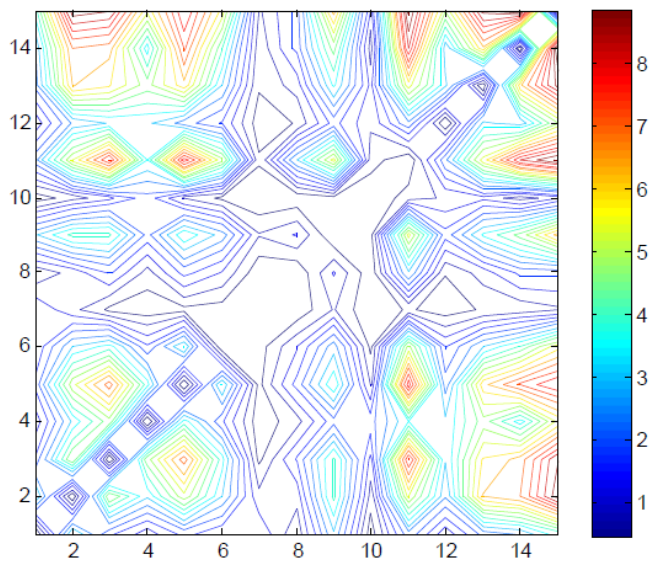


Figura 3.3: Comunicabilidad entre los nodos.

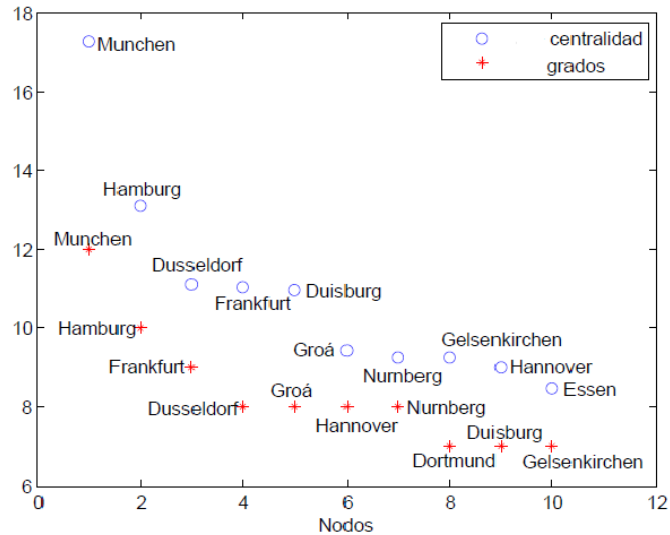


Figura 3.4: Ciudades ordenadas según el grado y la centralidad.

más importante de los dos. En este ejemplo, la centralidad del nodo 6 es bastante mayor que la del nodo 5, puesto que el nodo 6 está conectado al nodo 12 que es el nodo más importante de la red.

Al calcular la exponencial de la matriz de adyacencia no solo obtenemos la centralidad de cada nodo, sino, también la comunicabilidad entre los diferentes nodos. En la Figura 3.3 hemos representado con curvas de nivel los valores de los elementos no diagonales de la exponencial de la matriz de adyacencia. Los nodos que intercambian información con mejores resultados (o de mayor comunicabilidad), se corresponden con las curvas en color rojo. Por otra parte, entre los nodos que no hay comunicabilidad posible obtenemos espacios en blanco.

En el siguiente ejemplo, tomado de [4] vamos estudiar una matriz de dimensión 1168×1168 y muy dispersa. Corresponde a la matriz de adyacencia del grafo asociado a la red de autovías que conecta 1168 ciudades de Alemania. Se ha calculado la exponencial de dicha matriz con los métodos de la Sección 2.2 que hemos implementado en MATLAB y se han representado en la Figura 3.4 las 10 ciudades con mayor centralidad (en azul) y grado (en rojo). Observemos que en ambos casos los dos nodos "más importantes" coinciden.

3.2. Resolución de sistemas diferenciales lineales con término fuente constante

A lo largo de esta sección vamos a tomar como referencia [7].

Como ya se dijo en la Sección 2.3, el método basado en subespacios de Krylov para la aproximación del producto de la exponencial matricial por un vector resulta de utilidad para resolver sistemas de ecuaciones diferenciales lineales homogéneos, sobre todo en el caso de que la matriz A del sistema sea de grandes dimensiones y dispersa. En esta sección se ha considerado como aplicación un problema no homogéneo,

$$\begin{aligned} u'(t) &= Au(t) + f \\ u(t_0) &= u_0, \end{aligned} \quad (3.1)$$

donde $A \in \mathbb{R}^{n \times n}$, $f \in \mathbb{R}^n$, $t_0 > 0$ y $u_0 \in \mathbb{R}^n$.

Como sabemos, la fórmula de variación de las constantes proporciona la solución de (3.1),

$$u(t) = e^{(t-t_0)A}u_0 + \int_{t_0}^t e^{(t-s)A}f ds.$$

Teniendo en cuenta que f es constante y el Teorema 1.5 se concluye que

$$\begin{aligned} u(t) &= e^{(t-t_0)A}u_0 + \left(\int_{t_0}^t e^{(t-s)A} ds \right) f \\ &= \varphi_0((t-t_0)A)u_0 + (t-t_0)\varphi_1((t-t_0)A)f, \end{aligned} \quad (3.2)$$

donde

$$\varphi_0(x) = e^x, \quad \varphi_1(x) = \frac{e^x - 1}{x}.$$

Conocido u_0 , el término $\varphi_0((t-t_0)A)u_0$ puede ser aproximado mediante el método de los subespacios de Krylov como se indicó en la Sección 2.3. Sin embargo, tenemos que estudiar cómo aproximar eficientemente mediante dicho método el término $\varphi_1((t-t_0)A)f$, conocida f . Para ello vamos a considerar la siguiente matriz auxiliar

$$\tilde{H}_k = \left(\begin{array}{c|c} H_k & \mathbf{e}_1 \\ \hline \mathbf{0}^T & 0 \end{array} \right), \quad (3.3)$$

donde H_k es la matriz resultante del proceso de Arnoldi para la aproximación de Krylov que usaríamos al aproximar $\varphi_0((t-t_0)A)f$. Vamos a

calcular

$$e^{\tilde{H}_k} = \sum_{j=0}^{\infty} \frac{\tilde{H}_k^j}{j!},$$

para lo cual tenemos en cuenta que

$$\tilde{H}_k^2 = \left(\begin{array}{c|c} H_k & \mathbf{e}_1 \\ \mathbf{0}^T & 0 \end{array} \right) \cdot \left(\begin{array}{c|c} H_k & \mathbf{e}_1 \\ \mathbf{0}^T & 0 \end{array} \right) = \left(\begin{array}{c|c} H_k^2 & H_k \mathbf{e}_1 \\ \mathbf{0}^T & 0 \end{array} \right),$$

y, en general,

$$\tilde{H}_k^j = \left(\begin{array}{c|c} H_k^j & H_k^{j-1} \mathbf{e}_1 \\ \mathbf{0}^T & 0 \end{array} \right).$$

Como consecuencia de lo anterior

$$\begin{aligned} e^{\tilde{H}_k} &= \sum_{j=0}^{\infty} \frac{\tilde{H}_k^j}{j!} = I + \sum_{j=1}^{\infty} \frac{1}{j!} \left(\begin{array}{c|c} H_k^j & H_k^{j-1} \mathbf{e}_1 \\ \mathbf{0}^T & 0 \end{array} \right) \\ &= \left(\begin{array}{c|c} \sum_{j=0}^{\infty} \frac{1}{j!} H_k^j & \sum_{j=1}^{\infty} \frac{1}{j!} H_k^{j-1} \mathbf{e}_1 \\ \mathbf{0}^T & 0 \end{array} \right) \\ &= \left(\begin{array}{c|c} e^{H_k} & \varphi_1(H_k) \mathbf{e}_1 \\ \mathbf{0}^T & 0 \end{array} \right), \end{aligned}$$

pues

$$\varphi_1(x) = \frac{e^x - 1}{x} = \frac{\sum_{j=0}^{\infty} \frac{x^j}{j!} - 1}{x} = \frac{\sum_{j=1}^{\infty} \frac{x^j}{j!}}{x} = \sum_{j=1}^{\infty} \frac{x^{j-1}}{j!}.$$

Somos capaces entonces de aproximar el término $\varphi_1((t - t_0)A)f$ que aparece en la fórmula para la solución de (3.1) calculando mediante el proceso de Arnoldi la matriz H_k , considerando la matriz auxiliar \tilde{H}_k y calculando la exponencial matricial de nuestra matriz auxiliar con el método del escalado y el cuadrado.

Como ejemplo práctico vamos a considerar, siguiendo [7], el modelo de volatilidad estocástica de Heston, que está descrito mediante la siguiente ecuación en derivadas parciales de evolución

$$U_t = \frac{1}{2} v s^2 U_{ss} + \rho \lambda v s U_{vs} + \frac{1}{2} v \lambda^2 U_{vv} + (r_d - r_f) s U_s + \kappa(\eta - v) U_v - r_d U,$$

junto a la condición inicial

$$U(s, v, 0) = \max(s - K, 0),$$

y las condiciones frontera

$$\begin{aligned}
U(0, v, t) &= 0, \\
U(s, V, t) &= s, \\
U(S, v, t) &= 1, \\
U_t(s, 0, t) &= (r_f - r_d)sU_s(s, 0, t) - r_dU(s, 0, t) + \kappa\eta U(s, 0, t)_v.
\end{aligned}$$

Para los distintos parámetros involucrados en el modelo, se han fijado los considerados en [7]. Más precisamente $\rho = 0,8$, $\lambda = 0,3$, $r_d = 0,03$, $r_f = 0$, $\kappa = 2$, $\eta = 0,2$, $K = 100$. Nuestro objetivo es aproximar la solución en $t \geq 0$ para $(s, v) \in [0, 800] \times [0, 50]$. Discretizamos el dominio espacial usando una red rectangular de pasos Δ_s y Δ_v , respectivamente y usamos como en [7] las siguientes aproximaciones por diferencias finitas para las derivadas respecto de s y de v

$$\begin{aligned}
(U_s)_{i,j} &\approx \frac{U_{i+1,j} - U_{i-1,j}}{2\Delta_s}, \\
(U_{ss})_{i,j} &\approx \frac{U_{i+1,j} - 2U_{i,j} + U_{i-1,j}}{\Delta_s^2}, \\
(U_v)_{i,j} &\approx \frac{U_{i,j+1} - U_{i,j-1}}{2\Delta_v}, \\
(U_v)_{i,0} &\approx \frac{-3U_{i,0} + 4U_{i,1} - U_{i,2}}{2\Delta_v}, \\
(U_{vv})_{i,j} &\approx \frac{U_{i,j+1} - 2U_{i,j} + U_{i,j-1}}{\Delta_v^2}, \\
(U_{sv})_{i,j} &\approx \frac{U_{i+1,j+1} + U_{i-1,j-1} - U_{i-1,j+1} - U_{i+1,j-1}}{4\Delta_v\Delta_s}.
\end{aligned}$$

Llevándolo a nuestro problema obtenemos la siguiente relación para $U_{i,j}(t) \simeq U(s_i, v_j, t)$, donde $s_i = i\Delta_s$, $1 \leq i \leq N$, y $v_j = j\Delta_v$, $0 \leq j \leq M$, (por simplicidad se ha suprimido la dependencia explícita de U respecto de t). Aquí $\Delta_s = \frac{800}{N+1}$ y $\Delta_v = \frac{50}{M+1}$.

$$\begin{aligned}
(U_t)_{i,j} &= \frac{1}{2}v s_i^2 \left[\frac{U_{i+1,j} - 2U_{i,j} + U_{i-1,j}}{\Delta_s^2} \right] \\
&+ \rho \lambda v_j s_i \left[\frac{U_{i+1,j+1} + U_{i-1,j-1} - U_{i-1,j+1} - U_{i+1,j-1}}{4\Delta_v \Delta_s} \right] \\
&+ \frac{1}{2} \lambda^2 v \left[\frac{U_{i,j+1} - 2U_{i,j} + U_{i,j-1}}{\Delta_v^2} \right] \\
&+ (r_d - r_f) s_i \left[\frac{U_{i+1,j} - U_{i-1,j}}{2\Delta_s} \right] \\
&+ \kappa(\eta - v_j) \left[\frac{U_{i,j+1} - U_{i,j-1}}{2\Delta_v} \right] - r_d U_{i,j},
\end{aligned}$$

en los puntos (s_i, v_j) con $j \neq 0$, y

$$(U_t)_{i,0} = (r_d - r_f) s_i \left[\frac{U_{i+1,0} - U_{i-1,0}}{2\Delta_s} \right] + \kappa \eta \left[\frac{-3U_{i,0} + 4U_{i,1} - U_{i,2}}{2\Delta_v} \right] - r_d U_{i,0}$$

en los puntos (s_i, v_j) con $j = 0$. Estas ecuaciones se corresponden con un problema del tipo $u' = Au + f$, que sabemos resolver.

Hemos implementado en MATLAB un programa para aproximar $\varphi_1(\Delta_t A)f$ y lo hemos aprovechado para aproximar la solución del problema a lo largo del intervalo temporal $[0, 1]$. Para ello, se ha utilizado repetidamente la relación análoga a (3.2)

$$u_{k+1} = \varphi_0(\Delta_t A)u_k + \Delta_t \varphi_1(\Delta_t A)f,$$

con $t_k = k\Delta_t$, $0 \leq k \leq N_t - 1$, $\Delta_t = \frac{1}{N_t}$ que proporciona aproximaciones $u_k \simeq u(t_k)$.

En la Figura 3.5 se ha representado la restricción de la condición inicial al dominio $[0, 200] \times [0, 1]$, mientras que en la Figura 3.6 se ha representado la restricción de la solución numérica calculada en $t = 1$ utilizando $k = 200$ para la dimensión del espacio de Krylov con el que se ha trabajado y avanzando en tiempo 10 pasos de longitud $\Delta_t = 0,1$.

En la Figura 3.6 se puede observar cómo la solución tiene un comportamiento de suavizado ya que en $t = 1$ la solución es más regular que la condición inicial.

Además, hemos recogido en la Tabla 3.1 los errores de las aproximaciones que hemos calculado para $k = 100$ y 200 y $N_t = 10, 20, 40$ y 80 . Igual que antes, k es la dimensión del espacio de Krylov con el que se

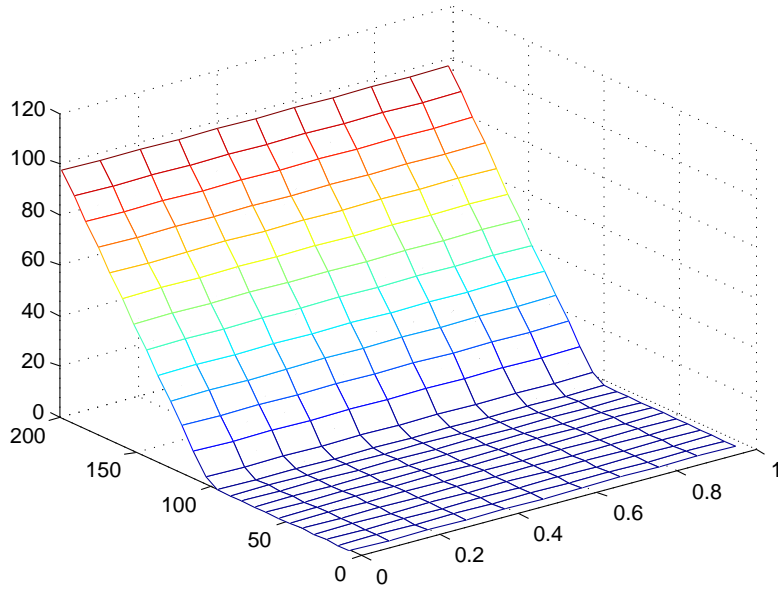


Figura 3.5: Condición inicial.

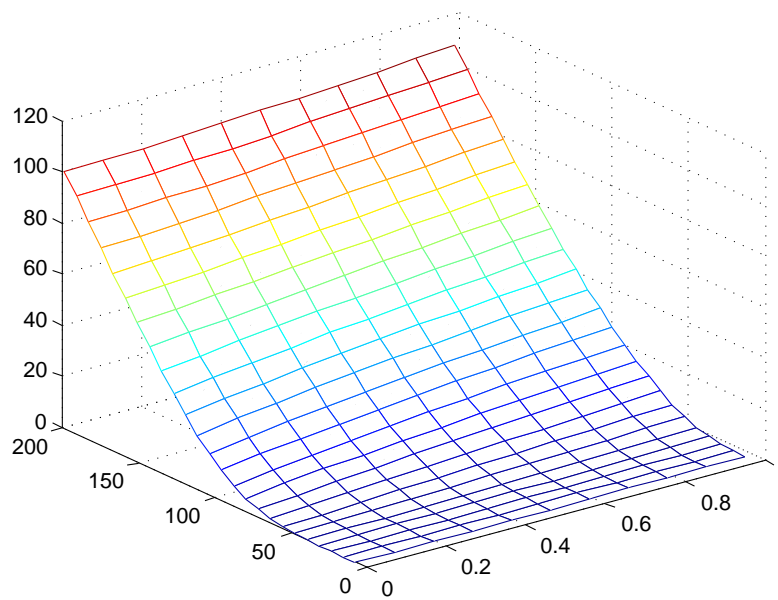


Figura 3.6: Solución en $t = 1$.

ha trabajado (el paso en el que detenemos el proceso de Arnoldi) y N_t es el número de pasos temporales que usamos para avanzar desde $t = 0$ hasta $t = 1$. Como solución exacta de referencia para calcular los errores se ha utilizado la hallada con los programas de Niessen y Wright [11].

	$k = 100$	$k = 200$
$Nt = 10$	–	$1,79 \times 10^{-1}$
$Nt = 20$	–	$3,29 \times 10^{-4}$
$Nt = 40$	$2,48 \times 10^{-1}$	$3,52 \times 10^{-9}$
$Nt = 80$	$8,92 \times 10^{-4}$	$5,30 \times 10^{-10}$

Cuadro 3.1: Máximo error en $t = 1$ en los nodos de la red restringida a $[0, 200] \times [0, 1]$.

Idealmente, si el proceso de Arnoldi se realizase hasta el final (o con k igual a la dimensión de la matriz), sería posible tomar $N_t = 1$ y se obtendría en un sólo paso temporal la solución exacta hasta la precisión del ordenador, pero el costo operativo sería muy elevado (la matriz con la que trabajamos tiene dimensión 5100×5100 , aunque es muy dispersa).

Cuando se fija en el proceso de Arnoldi $k = 100$, con 10 y 20 pasos temporales se obtienen errores de tamaño unidad (representados en la tabla con –), pero a partir de $N_t = 40$ los errores decrecen considerablemente. Si se eleva el valor de k hasta 200, vemos que los errores obtenidos con 20, 40 y 80 pasos temporales son muy razonables y se reducen mucho al duplicar N_t . Errores inferiores a 10^{-10} no son significativos puesto que la solución de referencia que se obtiene con los programas de [11] está calculada con esa precisión.

Como consecuencia de lo anterior, podemos concluir el método basado en los subespacios de Krylov es una buena opción para aproximar las soluciones de éste tipo de problemas.

Bibliografía

- [1] Ernesto Estrada & Desmond J. Higham *Network Properties Revealed through Matrix Functions*, SIAM Review, vol.52, No 4, (2010), 696–714.
- [2] Walter Gautschi, *Numerical Analysis. An introduction*, ed Birkhäuser (1997).
- [3] Nicholas J. Higham, *Functions of Matrices: Theory and Computation*, SIAM, 2008.
- [4] Marcus Kaiser & Claus C. Hilgetag , *Spatial growth of real-world networks*, PHYSICAL REVIEW E 69, 036103, (2004).
- [5] Jörg Liesen & Volker Mehrmann, *Linear Algebra*, Springer.
- [6] Cleve Moler, Charles Van Loan, *Nineteen dubious ways to compute the Exponential of a Matrix, Twenty-Five Years Later*, SIAM Review, vol.45, No 1, (2003),49.
- [7] Jitse Niesen & Will M. Wright , *Algorithm 919: A Krylov Subspace Algorithm for Evaluating the ϕ -Functions Appearing in Exponential Integrators*, ACM Trans. Math. Softw. 38, 3, Article 22, (2012).
- [8] Yousef Saad , *Analysis of some Krylov subspace approximations to the Matrix Exponential Operator*, SIAM J. Numer. Anal. vol. 29, No 1, (1992), 209–228.
- [9] Charles Van Loan, *The Sensitivity of the Matrix Exponential*, SIAM J. Numer. Anal. vol.14, No 6, (1997), 971-981.
- [10] David S. Watkins, *Fundamentals of Matrix Computations*, John Wiley and Sons, 2002.
- [11] <http://www.leeds.ac.uk/jitse/software.html>

Apéndice A

Programas de MATLAB

En este apéndice recopilamos los programas que hemos realizado en MATLAB sobre cada uno de los métodos numéricos que hemos estudiado a lo largo de la teoría.

A.1. `padeexponencial.m`

En este programa hemos implementado la aproximación diagonal de Padé para cualquier matriz A y para cualquier grado p .

```
function [aproxpade]=padeexponencial(p,A)
N=eye(size(A)); % Inicializamos la matriz  $N$ .
D=eye(size(A)); % Inicializamos la matriz  $D$ .
cn=1; % Inicializamos  $cn$  al coeficiente  $a_0$  del numerador
de la aproximación de Padé.
M=eye(size(A)); %  $M$  nos ayudara a realizar las potencias
de  $A$  sin perder los datos guardados.
for j=1:p
    cn=(p-j+1)*cn/(j*(2*p-j+1)); % Los coeficientes de la
aproximación cumplen esta relación recursiva.
    M=M*A;
    N=N+cn*M; % Actualizamos  $N$ .
    D=D+((-1)^j)*cn*M; % Actualizamos  $D$ .
end
aproxpade=D \ N; % Por último calculamos el aproximante de
Padé.
end
```

A.2. escaladoycuadrado.m

Implementamos el método del escalado y el cuadrado estudiado en el Sección 2,2, usando como método de aproximación para la exponencial de $\frac{A}{m}$ el método de Padé del programa anterior.

```
function [E]=escaladoycuadrado(p,A)

j=1;
m=2;

% Calculo de m.
while (norm(A)/m)>1
    m=2*m;
    j=j+1; % Exponente de 2 en m.
end
% Alternativamente este paso se puede realizar con el
comando [j,f]=log2(norm(A)) que nos da de una forma
más directa j.

Am=(1/m)*A;
[E]=padeexponencial(p,Am);

for i=1:j
    E=E*E;
end

end
```

A.3. Arnoldi.m

Implementamos el algoritmo estudiado para el proceso de Arnoldi en el Sección 2,3.

```
function [m,Q,H]=Arnoldi(A,q,w)

% Dados A y q, el programa va a realizar el proceso de
Arnoldi para obtener las matrices  $Q_k$  y  $H_k$  de la teoría,
w marca en que iteración vamos a parar.

[~,n]=size(A);
Q=zeros(n,w); % A lo sumo la matriz Q va a tener orden
n x w.
```



```

H=zeros(w,w); % De la misma forma a lo sumo  $H$  tiene orden
 $w \times w$ .
Q(:,1)=q; %El primer vector de Arnoldi es el vector  $q$ .
m=w; % A priori debemos pensar que vamos a realizar las
 $w$  iteraciones.

% Empezamos el bucle del proceso:
for k=1:m
    z=A*Q(:,k);
    for i=1:k
        H(i,k)=Q(:,i)'*z;
        z=z-H(i,k)*Q(:,i);
    end
    H(k+1,k)=norm(z,2);
    if H(k+1,k)==0
        m=k;
        break;
    end
    Q(:,k+1)=z/H(k+1,k);
end

% En este momento hay que tener en cuenta que quizás no
hayamos realizado  $w$  iteraciones, luego:
Q=Q(1:n,1:m);
H=H(1:m,1:m);

end

```

A.4. subespaciodekrylov.m

Implementamos una función que hace uso de los dos programas anteriores para aproximar $f(A)b$.

```

function [apx]=subespaciodekrylov(A,b,w)

% Grados del aproximante de Padé que vamos a usar.
p=12;
s=norm(b,2);
% Vector que inicia el proceso de Arnoldi.
q=b/s;
[m,Q,H]=Arnoldi(A,q,w);

```

```
[E]=escaladoycuadrado(p,H);
apx=s*Q*E(:,1);

end
```

A.5. centralidad.m

Dada una matriz de adyacencia y una lista de etiquetas de los nodos este programa calcula la centralidad y el grado de cada nodo generando una gráfica y nos devuelve la lista de etiquetas ordenadas en función de la centralidad.

```
function [d,deg,lorddeg,lordcentr]=centralidad(A,l)
% A matriz de adyacencia.
% l sera la lista de etiquetas que le asignamos a los
nodos.

[m,n]=size(A);
d=zeros(1,n);

[E]=escaladoycuadrado(12,A);

% Centralidad:
for i=1:n
    d(i)=E(i,i);
end
x=1:n;

% Grados
deg=zeros(1,n);
for j=1:n
    for i=1:n
        deg(j)=deg(j)+A(j,i);
    end
end

% Ordenamos las etiquetas.
[d,perm1]=sort(d,'descend');
[deg,perm2]=sort(deg,'descend');
[ ,s]=size(l);
lordcentr=zeros(s,1);
lordcentr=l(perm1);
lorddeg=zeros(s,1);
```

```

lorddeg=l(perm2);
% Gráficas, nos dibuja todos los grados y centralidades
ya ordenados.
figure(1)
clf
plot(x,d,'o b')
hold on
plot(x,deg,'* r')
xlabel('Nodos')
legend('centralidad','grados',2)
end

```

A.6. aproximacionphilf.m

Implementamos una función que aproxima mediante el método basado en subespacios de Krylov la función $\varphi_1(tA)f$ siguiendo las consideraciones hechas en la Sección 3.2 .

```

function [philf]=aproximacionphilf(A,b,w)
p=12;
% queremos aproximar  $\varphi_1(TA)*f$ .
beta=norm(b);
b=b/beta;
[Q,H]=Arnoldi(A,b,w); % Aplicamos el proceso de Arnoldi
para calcular la  $H_k$  original.
% Ampliamos a la  $H_k$  tilde:
Htilde=zeros(w+1,w+1);
Htilde(1:w,1:w)=H; %  $H$  siempre es  $w \times w$ .
Htilde(1,w+1)=1;
% Calculamos la exponencial de Htilde
E=escaladoycuadrado(p,Htilde);
% Aproximamos  $\varphi_1(A)*f$ .
philf=E(1:w,w+1);
philf=beta*Q*philf;
end

```

A.7. ecuacionHeston.m

Implementamos una función que haciendo uso de `aproximacion-philf.m` y de `subespaciodekrylov.m` nos calcula la solución del modelo de Heston en el intervalo $[0, T]$, donde además escogemos el número de pasos en tiempo que vamos a llevar a cabo (N_t) y la dimensión del subespacio de Krylov en el que vamos a realizar la aproximación (w).

```
function [maximo]=ecuacionHeston(T,Nt,w)

load exacta

% Discretizamos  $[0, T]$ .
dt=T/Nt;
t=0:dt:T;

%Malla:
%  $\Delta_s=h$  ,  $\Delta_v=k$ .
% Vamos a considerar  $N$  y  $M$  como 100 y 51
N=100;% Subintervalos en  $s$ .
M=50;% Subintervalos en  $v$ .
V=5;% Extremo del intervalo para  $v$ .
S=800;% Extremo del intervalo para  $s$ .
k=V/(M+1);% paso en la malla de  $v$ .
h=S/(N+1);% paso en la malla de  $s$ .
% red:
v=0:k:V-k;
s=h:h:S-h;
dimen=N*(M+1);

%El índice  $i$  será para las  $s$  y  $j$  para las  $v$ .

%Parámetros del modelo
rd=0.03;
rf=0.0;
eta=0.2;
kappa=2;
lambda=0.3;
rho=0.8;

[A, f] = FDHeston(h,k,N,M,rd,rf,eta,kappa,lambda,rho);
%Dadas las constantes obtenemos la matriz del
sistema y el termino fuente.
A=A*dt;
```

```

%Definimos la condición inicial restringida al dominio
pequeño:
K=100;
u0=zeros(N,M+1);
for j=1:(M+1)
    u0(:,j)=max(s-K,0);
end
%Dibujamos la condicion inicial.
mesh(v(1:11),s(1:25),u0(1:25,1:11))
axis([0 1 0 200 0 120])
pause
%pasamos la cond. ind. a vector:
u0=reshape(u0,dimen,1);

%Ahora vamos a aproximar la solución en los nodos que
%generamos en [0,T]:
condinic=u0;%Condicion inicial
for i=1:Nt
    apx1=subespaciodekrylov(A,condinic,w);
    philf=aproximacionphilf(A,f,w);
    u=apx1+dt*philf;
    condinic=u;
end

u=reshape(u,N,M+1);
%Al salir del bucle tenemos la solución aproximada en 1.
mesh(v(1:11),s(1:25),u(1:25,1:11))
pause
%Medimos errores en el recinto pequeño:
err=abs(SolExact(1:25,1:11)-u(1:25,1:11));
maximo=max(max(err));
end

```