Resolución para X del sistema $AX - XB = C$

Javier Jorge Cano

19 de febrero de 2018

1. Problema propuesto

El problema a resolver es el siguiente:

$$AX - XB = C; A, B, C, X \in \mathbb{R}^{nxn} \tag{1}$$

Pare resolverlo, podemos descomponer B como sigue mediante la factorización de Schur:

$$B = QTQ^T; QQ^T = I, T \in \text{tr.sup o } T \in \text{Forma Real de Schur} \tag{2}$$

Sustituyendo en la fórmula original:

$$AX - XQTQ^T = C \tag{3}$$
$$(AX - XQTQ^T)Q = CQ \tag{4}$$
$$= AXQ - XQTQ^TQ = CQ \tag{5}$$
$$= AXQ - XQT = CQ \tag{6}$$
$$\tag{7}$$

Si consideramos que $Y = XQ$ y $D = CQ$, obtenemos:

$$AY - YT = D \tag{8}$$

Que se trata de un problema semejante al primero pero en este caso T puede tomar dos formas:

1. T es triangular superior.
2. T está en la forma de Schur.

Consideramos estas dos opciones por separado a continuación.

1.1. Opción 1: T es triangular superior

Si T es triangular superior, entonces obtendríamos un esquema cómo el que se muestra a continuación para matrices $\mathbb{R}^{4x4}$:

$$\begin{pmatrix} a_{1,1} & a_{1,2} & a_{1,3} & a_{1,4} \\ a_{2,1} & a_{2,2} & a_{2,3} & a_{2,4} \\ a_{3,1} & a_{3,2} & a_{3,3} & a_{3,4} \\ a_{4,1} & a_{4,2} & a_{4,3} & a_{4,4} \end{pmatrix} \begin{pmatrix} y_{1,1} & y_{1,2} & y_{1,3} & y_{1,4} \\ y_{2,1} & y_{2,2} & y_{2,3} & y_{2,4} \\ y_{3,1} & y_{3,2} & y_{3,3} & y_{3,4} \\ y_{4,1} & y_{4,2} & y_{4,3} & y_{4,4} \end{pmatrix} - \begin{pmatrix} y_{1,1} & y_{1,2} & y_{1,3} & y_{1,4} \\ y_{2,1} & y_{2,2} & y_{2,3} & y_{2,4} \\ y_{3,1} & y_{3,2} & y_{3,3} & y_{3,4} \\ y_{4,1} & y_{4,2} & y_{4,3} & y_{4,4} \end{pmatrix} \begin{pmatrix} t_{1,1} & t_{1,2} & t_{1,3} & t_{1,4} \\ 0 & t_{2,2} & t_{2,3} & t_{2,4} \\ 0 & 0 & t_{3,3} & t_{3,4} \\ 0 & 0 & 0 & t_{4,4} \end{pmatrix} = \begin{pmatrix} d_{1,1} & d_{1,2} & d_{1,3} & d_{1,4} \\ d_{2,1} & d_{2,2} & d_{2,3} & d_{2,4} \\ d_{3,1} & d_{3,2} & d_{3,3} & d_{3,4} \\ d_{4,1} & d_{4,2} & d_{4,3} & d_{4,4} \end{pmatrix} \tag{9}$$

Considerando los valores en función de la primera columna de Y (donde $Y_{\cdot,i}$ representa la columna i-ésima completa de Y), podemos obtener la siguiente ecuación para la primera columna de D:

$$\begin{align} AY_{\cdot,1} - Y_{\cdot,1}t_{1,1} = D_{\cdot,1} \tag{10} \\ (A - t_{1,1}I)Y_{\cdot,1} = D_{\cdot,1} \tag{11} \\ \tag{12} \end{align}$$

Donde A, D y $t_{1,1}$ con conocidos, por lo tanto $(A - t_{1,1}I)$ es conocido, resultando en un sistema de ecuaciones con la siguiente forma:

$$Zx = b \tag{13}$$

Donde $Z = (A - t_{1,1}I)$, $x = Y_{\cdot,1}$ and $b = D_{\cdot,1}$. Resolviendo para $Y_{\cdot,1}$, ya podemos calcular de forma iterativa las siguientes columnas, por ejemplo, la segunda:

$$\begin{align} AY_{\cdot,2} - t_{1,2}Y_{\cdot,1} - t_{2,2}Y_{\cdot,2} = D_2 \tag{14} \\ (A - t_{2,2}I)Y_{\cdot,2} = D_2 + t_{1,2}Y_{\cdot,1} \tag{15} \\ \tag{16} \end{align}$$

Donde, de nuevo, $(A - t_{2,2}I)$ y el lado derecho de la ecuación son dados. Iterando sobre la matriz y resolviendo los sistemas, solamente quedaría deshacer el cambio, considerando que Q es ortogonal y por lo tanto $QQ^T = I$ y $Q^{-1} = Q^T$.

$$\begin{align} Y = XQ \tag{17} \\ YQ^T = XQQ^T \tag{18} \\ YQ^T = X \tag{19} \\ \tag{20} \end{align}$$

Obteniendo finalmente el valor para X.

1.2. Opción 2: T está en forma real de Schur

En el caso de que T este en forma real de Schur, el sistema podría tener la siguiente estructura:

$$A\begin{pmatrix} y_{1,1} & y_{1,2} & y_{1,3} & y_{1,4} \\ y_{2,1} & y_{2,2} & y_{2,3} & y_{2,4} \\ y_{3,1} & y_{3,2} & y_{3,3} & y_{3,4} \\ y_{4,1} & y_{4,2} & y_{4,3} & y_{4,4} \end{pmatrix} - \begin{pmatrix} y_{1,1} & y_{1,2} & y_{1,3} & y_{1,4} \\ y_{2,1} & y_{2,2} & y_{2,3} & y_{2,4} \\ y_{3,1} & y_{3,2} & y_{3,3} & y_{3,4} \\ y_{4,1} & y_{4,2} & y_{4,3} & y_{4,4} \end{pmatrix} \begin{pmatrix} t_{1,1} & t_{1,2} & t_{1,3} & t_{1,4} \\ 0 & t_{2,2} & t_{2,3} & t_{2,4} \\ 0 & t_{3,2} & t_{3,3} & t_{3,4} \\ 0 & 0 & 0 & t_{4,4} \end{pmatrix} = \begin{pmatrix} d_{1,1} & d_{1,2} & d_{1,3} & d_{1,4} \\ d_{2,1} & d_{2,2} & d_{2,3} & d_{2,4} \\ d_{3,1} & d_{3,2} & d_{3,3} & d_{3,4} \\ d_{4,1} & d_{4,2} & d_{4,3} & d_{4,4} \end{pmatrix} \tag{21}$$

Encontrando elementos no nulos por debajo de la diagonal principal, formando bloques de $2x2$ como se muestra con los elementos $t_{2,2}, t_{2,3}, t_{3,2}$ y $t_{3,3}$. Estos bloques podrían aparecer en cualquier posición de la diagonal, obligando a considerar estos casos para cada posición. Por lo tanto, en este caso podríamos representar el problema como sigue en función de la primera columna de Y como en el caso anterior:

$$AY_{\cdot,1} - Y_{\cdot,1}t_{1,1} = D_{\cdot,1} \tag{22}$$
$$(A - t_{1,1}I)Y_{\cdot,1} = D_{\cdot,1} \tag{23}$$
$$\tag{24}$$

Sin embargo, al obtener la segunda columna observamos que es necesaria también la tercera. Si detallamos el desarrollo de la multiplicación YT, la dependencia es clara:

$$\begin{pmatrix} y_{1,1} & y_{1,2} & y_{1,3} & y_{1,4} \\ y_{2,1} & y_{2,2} & y_{2,3} & y_{2,4} \\ y_{3,1} & y_{3,2} & y_{3,3} & y_{3,4} \\ y_{4,1} & y_{4,2} & y_{4,3} & y_{4,4} \end{pmatrix} \begin{pmatrix} t_{1,1} & t_{1,2} & t_{1,3} & \dots \\ 0 & t_{2,2} & t_{2,3} & \dots \\ 0 & t_{3,2} & t_{3,3} & \dots \\ 0 & 0 & 0 & \dots \end{pmatrix} = \begin{pmatrix} y_{1,1}t_{1,1} & y_{1,1}t_{1,2} + y_{1,2}t_{2,2} + y_{1,3}t_{3,2} & \dots \\ 0 & y_{2,1}t_{1,2} + y_{2,2}t_{2,2} + y_{2,3}t_{3,2} & \dots \\ 0 & y_{3,1}t_{1,2} + y_{3,2}t_{2,2} + y_{3,3}t_{3,2} & \dots \\ 0 & y_{4,1}t_{1,2} + y_{4,2}t_{2,2} + y_{4,3}t_{3,2} & \dots \end{pmatrix} \tag{25}$$

$$\begin{pmatrix} y_{1,1} & y_{1,2} & y_{1,3} & y_{1,4} \\ y_{2,1} & y_{2,2} & y_{2,3} & y_{2,4} \\ y_{3,1} & y_{3,2} & y_{3,3} & y_{3,4} \\ y_{4,1} & y_{4,2} & y_{4,3} & y_{4,4} \end{pmatrix} \begin{pmatrix} t_{1,1} & t_{1,2} & t_{1,3} & t_{1,4} \\ 0 & t_{2,2} & t_{2,3} & t_{2,4} \\ 0 & t_{3,2} & t_{3,3} & t_{3,4} \\ 0 & 0 & 0 & t_{4,4} \end{pmatrix} = \begin{pmatrix} y_{1,1}t_{1,1} & \cdots & y_{1,1}t_{1,3} + y_{2,2}t_{2,3} + y_{3,3}t_{3,3} & t_{1,4} \\ 0 & \cdots & y_{2,1}t_{1,3} + y_{2,2}t_{2,3} + y_{3,3}t_{3,3} & t_{2,4} \\ 0 & \cdots & y_{3,1}t_{1,3} + y_{2,2}t_{2,3} + y_{3,3}t_{3,3} & t_{3,4} \\ 0 & \cdots & y_{4,1}t_{1,3} + y_{2,2}t_{2,3} + y_{3,3}t_{3,3} & t_{4,4} \end{pmatrix} \tag{26}$$

Por lo tanto, necesitamos para las columnas $D_{\cdot,2}$ y $D_{\cdot,3}$ obtener estos valores:

$$Y_{\cdot,1}t_{1,2} + Y_{\cdot,2}t_{2,2} + Y_{\cdot,3}t_{3,2} \to D_{\cdot,2} \tag{27}$$
$$Y_{\cdot,1}t_{1,3} + Y_{\cdot,2}t_{2,3} + Y_{\cdot,3}t_{3,3} \to D_{\cdot,3} \tag{28}$$

Si lo reformulamos incluyendo la multiplicación por la matriz A, obtenemos un sistema de ecuaciones como el que se muestra a continuación:

$$AY_{\cdot,2} - Y_{\cdot,1}t_{1,2} - Y_{\cdot,2}t_{2,2} - Y_{\cdot,3}t_{3,2} = D_2 \tag{29}$$
$$AY_{\cdot,3} - Y_{\cdot,1}t_{1,3} - Y_{\cdot,2}t_{2,3} - Y_{\cdot,3}t_{3,3} = D_3 \tag{30}$$

Desarrollando:

$$(A - t_{2,2}I)Y_{\cdot,2} - Y_{\cdot,1}t_{1,2} - Y_{\cdot,3}t_{3,2} = D_2 \tag{31}$$
$$AY_{\cdot,3} - Y_{\cdot,1}t_{1,3} - Y_{\cdot,2}t_{2,3} - Y_{\cdot,3}t_{3,3} = D_3 \tag{32}$$

$$(A - t_{2,2}I)Y_{\cdot,2} - Y_{\cdot,1}t_{1,2} - D_2 = Y_{\cdot,3}t_{3,2} \tag{33}$$
$$AY_{\cdot,3} - Y_{\cdot,1}t_{1,3} - Y_{\cdot,2}t_{2,3} - Y_{\cdot,3}t_{3,3} = D_3 \tag{34}$$

$$Y_{\cdot,3} = \frac{(A - t_{2,2}I)Y_{\cdot,2} - Y_{\cdot,1}t_{1,2} - D_2}{t_{3,2}} \tag{35}$$
$$AY_{\cdot,3} - Y_{\cdot,1}t_{1,3} - Y_{\cdot,2}t_{2,3} - Y_{\cdot,3}t_{3,3} = D_3 \tag{36}$$

$$Y_{\cdot,3} = \frac{(A - t_{2,2}I)Y_{\cdot,2}}{t_{3,2}} - \frac{Y_{\cdot,1}t_{1,2}}{t_{3,2}} - \frac{D_2}{t_{3,2}} \tag{37}$$

Por simplicidad, reorganizaremos los términos, considerando:

$$K = \frac{(A - t_{2,2}I)}{t_{3,2}} \tag{38}$$
$$S = \frac{Y_{\cdot,1}t_{1,2}}{t_{3,2}} - \frac{D_2}{t_{3,2}} \tag{39}$$
$$\tag{40}$$

Obtenemos:

$$Y_{\cdot,3} = KY_{\cdot,2} - S \tag{41}$$
$$\tag{42}$$

$$AY_{\cdot,3} - Y_{\cdot,1}t_{1,3} - Y_{\cdot,2}t_{2,3} - Y_{\cdot,3}t_{3,3} = D_3 \tag{43}$$
$$A(KY_{\cdot,2} - S) - Y_{\cdot,1}t_{1,3} - Y_{\cdot,2}t_{2,3} - (KY_{\cdot,2} - S)t_{3,3} = D_3 \tag{44}$$
$$AKY_{\cdot,2} - AS - Y_{\cdot,1}t_{1,3} - Y_{\cdot,2}t_{2,3} - KY_{\cdot,2}t_{3,3} - St_{3,3} = D_3 \tag{45}$$
$$(AK - Kt_{3,3} - t_{1,3}I)Y_{\cdot,2} - AS - Y_{\cdot,1}t_{1,3} - St_{3,3} = D_3 \tag{46}$$
$$(AK - Kt_{3,3} - t_{1,3}I)Y_{\cdot,2} = D_3 + AS + Y_{\cdot,1}t_{1,3} + St_{3,3} \tag{47}$$
$$\tag{48}$$

Donde, salvo $Y_{\cdot,2}$, todo lo demás es conocido. Resolviendo el sistema de ecuaciones obtenemos la segunda columna. Con esto, solamente queda sustituir en la segunda ecuación el valor obtenido para $Y_{\cdot,2}$:

$$
\begin{aligned}
AY_{\cdot,3} - Y_{\cdot,1}t_{1,3} - Y_{\cdot,2}t_{2,3} - Y_{\cdot,3}t_{3,3} &= D_3 && (49)\\
AY_{\cdot,3} - Y_{\cdot,3}t_{3,3} &= D_3 + Y_{\cdot,1}t_{1,3} + Y_{\cdot,2}t_{2,3} && (50)\\
AY_{\cdot,3} - Y_{\cdot,3}t_{3,3} &= D_3 + P && (51)\\
& && (52)
\end{aligned}
$$

Donde $P = Y_{\cdot,1}t_{1,3} + Y_{\cdot,2}t_{2,3}$, donde todo es conocido, resultando en una ecuación como en el caso de obtener una T triangular superior.

Por último, deshacemos el cambio de variable como en el caso de la triangular superior, obteniendo el valor para X.

2. Implementación: Llamadas a BLAS/LAPACK

En primer lugar, se han utilizado llamadas las siguientes llamadas a BLAS de diferentes niveles:

Instrucción	**Función**
BLAS - Nivel 1	
dscal (nElem, escalarA, vectorX, incX)	$\mathbf{x} \leftarrow \mathbf{x} \cdot \alpha$
dcopy (nElem, vectorX, incX, vectorY, incY)	$\mathbf{y} \leftarrow \mathbf{x}$
daxpy (nElem, escalarA, vectorX, incX, vectorY, incY)	$\mathbf{y} \leftarrow \mathbf{y} + \alpha\mathbf{x}$
BLAS - Nivel 2	
dgemv (transp, elemM, elemN, escalarA, A, leadingA, vectorX, incX, escalarB, vectorY, incY)	$\mathbf{y} \leftarrow \alpha * A * \mathbf{x} + \beta * \mathbf{y}$
BLAS - Nivel 3	
dgemm (transpA, transpB, elemM, elemN, elemK, escalarA, A, leadingA, B, leadingB, escalarB, C, leadingC)	$C \leftarrow \alpha * A * B + \beta C$

Los términos que aparecen en las llamadas son los siguientes:

- nElem: Referencia a *integer*. Número de elementos en el vector.
- escalarX: Referencia a *double*. El valor del escalar x
- vectorX: Referencia a vector de valores *double*. Posición en memoria del vector $\mathbf{x}$.
- incX: Referencia a *integer*. Número de posiciones entre elementos consecutivos, proporcionando flexibilidad, por ejemplo, para multiplicar uno de cada dos elementos con un incremento de 2.
- transp{X}: Referencia a *char*. Valores N o T para no transponer o transponer ,respectivamente, la matriz X.

- elemM, elemN, elemK: Dimensiones en el caso de la multiplicación de matrices donde $C \leftarrow A * B$, $A \in \mathbb{R}^{mxn}, B \in \mathbb{R}^{nxk}$ y $C \in \mathbb{R}^{mxk}$.
- leadingX: Referencia a *integer*. Cantidad de elementos que separan una columna de otra.

Por otro lado, para rutinas más complejas, se ha requerido de las siguientes llamadas a LAPACK:

2.1. DGESV: Solución del sistema de ecuaciones lineales $AX = B$

Esta llamada computa el la solución del sistema de ecuaciones lineales con número reales. De acuerdo con la documentación, realiza la descomposición LU con pivotación parcial e intercambios de filas para facilitar la estabilidad numérica. Un ejemplo de llamada a esta rutina sería:

$$dgesv(elemN, NRHS, A, leadingA, IPIV, B, leadingB, info) \tag{53}$$

Donde:

- elemN: Tamaño de la matriz, como se ha descrito anteriormente.
- NRHS: Número de vectores b o columnas de la matriz B, considerando que se pueden proporcionar más de un vector de términos constantes.
- A: Referencia a vector de *double*: La matriz de coeficientes.
- ipiv: Referencia a vector de *integer*: Parámetro de salida que proporciona la matriz de pivotación.
- B: Referencia a vector de *double*: El vector/matriz de términos constantes.
- info: Referencia a *integer*: Valor de retorno con información sobre el proceso: errores, optimizaciones, etc.

2.2. DGEES: Factorización de Schur

Esta llamada computa, para una matriz general A, los *eigenvalores*, la forma real de Schur T y los vectores de Schur Z. En este caso, esta es la descomposición deseada para B con la forma $B = QTQ^T$, donde T es la matriz devuelta por el método y Q es la matriz con los vectores de Schur. Un ejemplo de llamada a esta función sería la siguiente:

$$dgees(jobvs, sort, select, elemN, A, leadingA, sdim, wr, \tag{54}$$
$$wi, VS, ldvs, work, lwork, bwork, info) \tag{55}$$

Donde:

- *jobvs*: $N|V$: Caracteres indicadores para computar o no computar los vectores de Schur.
- *sort*: $N|S$: Caracteres indicadores para ordenar o no los *eigenvalores*, si es S se puede especificar una función de ordenación con *select*.
- *select*: Función específica para la ordenación de *eigenvalores*.
- *elemN*: Elementos de la matriz.
- A: Matriz que será descompuesta. Será sobrescrita con la matriz en forma Real de Schur T.
- *leadingA*: *Leading dimension* de A.
- *sdim*: Número de *eigenvalores* devueltos si hemos especificado un orden con *sort*.
- *wr*: Vector con la parte real de los *eigenvalores*.
- *wi*: Vector con la parte imaginaria de los *eigenvalores*.
- VS: Matriz Z de Schur, que en este ejemplo corresponde con la Q.
- *ldvs*: *Leading dimension* de VS.
- *work*: Espacio de memoria de trabajo para los cálculos, se puede obtener información sobre cómo debe ser con el resto de parámetros: *lwork* y *bwork*.
- *info*: Parámetro de salida con información sobre el proceso.

Para mayor detalle del funcionamiento y de los parámetros que interviene, se remite a la documentación oficial [1].

3. Comparativa de tiempos

Tras la implementación, se ha realizado un análisis temporal evaluando la compilación utilizando el compilador `icc`. Se ha incluido posteriormente la compilación con las opciones para las librerías de *Intel Matrix Kernel Libraries* (MKL)[2], variando entre la opción secuencial y la opción paralela.

[1] `http://www.netlib.org/lapack/`
[2] `https://software.intel.com/en-us/mkl`

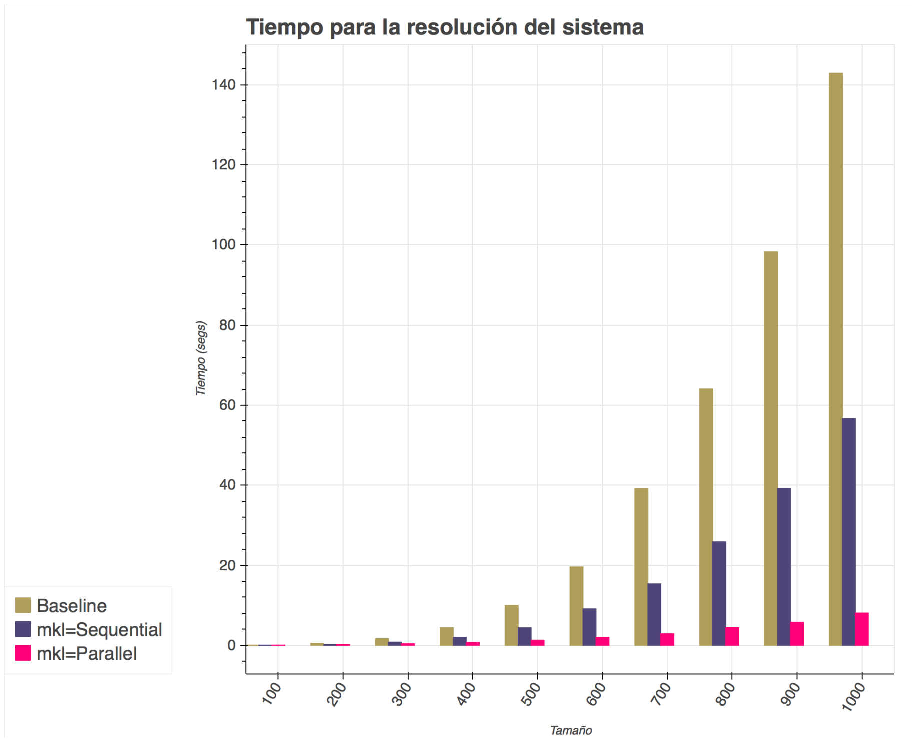

Figura 1: Comparación de los tiempos con diferentes opciones de compilación.

4. Conclusiones

Se ha implementado un algoritmo de resolución del sistemas de ecuaciones planteado, prototipado primero en *Matlab* y posteriormente implementado en C con llamadas a rutinas de los 3 niveles de $BLAS$ y dos rutinas avanzadas de $LAPACK$, para la resolución de ecuaciones y la obtención de la factorización de Schur.

Se han variado las opciones de compilación y se ha observado una mejora considerable en la reducción del tiempo de ejecución cuando se ha compilado con las librerías *MKL* de *Intel*, en ambas versiones, la secuencial y la paralela.

Como aporte personal, una vez se ha superado la primera barrera inicial para la implementación con *BLAS/LAPACK*, no es una tarea tan compleja y, sin embargo, el rendimiento obtenido al combinar estar rutinas con la compilación con las *MKL* es realmente interesante, abriendo la posibilidad a resolver problemas de tamaños que no podríamos considerar con la compilación básica.