Universidad Central de Venezuela

Facultad de Ciencias

Escuela de Computación

Asignatura: Cálculo Científico (6105)

Estudiante: Naranjo Sthory Alexanyer Antonio

Cédula de identidad: V – 26.498.600

Tarea 2: Factorización de Cholesky y Sistemas Especiales

A continuación, se presentan las respuestas de cada una de las preguntas indicadas para la actual asignación, donde se ha separado cada una según sus respectivos incisos, además de adjuntar la correspondiente justificación a la respuesta de cada una de las preguntas.

Se destaca también la carpeta cuyo nombre es ***Codes***, que contiene el código fuente de la resolución de aquellos ejercicios que lo requieren. De igual manera, en el presente informe se indican aquellas preguntas que se solventaron a partir de una implementación en Matlab/Octave y también se adjuntan imágenes de aquellos fragmentos de código relevantes para la justificación de la respuesta.

Respuestas 2)

Sea ***A*** una matriz simétrica y positivo definida. Demuestre:

a) ***A*** es no singular.

Demostración: Recordemos que una matriz no singular es aquella la cual $det(A) > 0$, entonces, es suficiente demostrando que el determinante de una matriz ***A*** simétrica y positiva es mayor a cero (0), para concluir que no es singular.

Tomemos una matriz ***A*** simétrica y positivo definida. Digamos que λ es un autovalor de ***A***. Entonces, para todo autovector **x** que pertenece a λ, tenemos que,

$$x^T A x = \lambda x^T x = \lambda ||x||^2$$

Así,

$$\lambda = \frac{x^T A x}{||x||^2} > 0$$

Por tanto, todos los autovalores pertenecientes a ***A*** deben ser positivos (demostrado en el inciso anterior de la actual pregunta). El producto de los autovalores de una matriz es igual al determinante. Como todos los valores propios de A son positivos, su producto debe ser positivo, por lo que,

$$det(A) \ > \ 0$$

Concluyendo que, si ***A*** es una matriz simétrica y positivo definida, entonces ***A*** no es singular.

b) $a_{ii} > 0$ para todo i (los elementos de la diagonal de ***A*** son todos positivos).

Demostración: Para un valor de i dado, tomemos $x = (x_j)$ el cual se encuentra definido por $x_i = 1$ y $x_j = 0$, si $j \neq i$. Donde $x \neq 0$.

Por tanto,

$$0 < x^t A x = a_{ii}.$$

Concluyendo que $a_{ii} > 0$, para cada $i = 1,2, \dots, n$, es decir, todos los elementos de la diagonal de ***A*** son todos positivos.

Referencia: Numerical Analysis, Richard L. Burden y J. Douglas Faire (Página 415 del libro o 431 del PDF).

c) ***A*** tiene autovalores reales y positivos.

Demostración: Primero, procedamos a demostrar que los autovalores de una matriz ***A*** Hermitiana, son reales, recordando que una matriz Hermitiana es similar a una matriz simétrica pero donde sus entradas son número complejos. La demostración que realizaremos a continuación será válida también para matrices simétricas.

Entonces, tomemos λ de un autovalor arbitratio de una matriz Hermitiana A, y tomemos un autovector **x** correspondiente al autovalor λ elegido anteriormente.

Teniendo así, lo siguiente,

$$Ax = \lambda x \ (*)$$

Multiplicamos por $\bar{x}^T$ desde el lado izquierdo, obteniendo así,

$$\bar{x}^T(Ax) = \bar{x}^T(\lambda x)$$

$$= \lambda \bar{x}^T x$$

$$= \lambda \left|\left|x\right|\right|$$

Ahora tomamos la transposición conjugada de ambos lados y obtenemos,

$$\bar{x}^T A^{-T} x = \bar{\lambda} \left|\left|x\right|\right| \ (**)$$

Donde ***A*** es una matriz Hermitiana, tenemos que $A^{-T} = A$.

Entonces, el lado izquierdo se convierte en,

$$\bar{x}^T A^{-T} x = \bar{x}^T A x$$

$$= \bar{x}^T \lambda x \ (Aplicamos\ (*))$$

$$= \lambda \left|\left|x\right|\right| \ (***)$$

Por tanto, comparando (**) y (***), obtenemos lo siguiente,

$$\lambda \left|\left|x\right|\right| = \bar{\lambda} \left|\left|x\right|\right|$$

Donde **x** es un autovector no nulo y su longitud $||x|| \neq 0$.

Dividiendo por la longitud $||x||$, obtenemos que $\lambda = \bar{\lambda}$, y esto implica que es λ es un número real. Donde λ es un autovalor arbitrario de ***A***, podemos concluir que para todo autovalor de una matriz Hermitiana ***A***, este es un valor real.

Con el resultado anterior, ahora demostremos que toda matriz simétrica y positiva, posee autovalores positivos.

Para ello, tomemos un autovalor (real) λ de A y un vector **x** correspondiente a un autovector real de la matriz ***A***. De esta manera, tenemos que,

$$Ax = \lambda x$$

Ahora, multipliquemos ambos miembros de la igualdad por x^T, obteniendo así,

$$x^T A x = \lambda x^T x$$

$$= \lambda ||x||^2$$

El lado izquierdo es positivo ya que ***A*** es definida positiva y **x** es un vector no nulo ya que es un vector propio.

Donde la longitud de $||x||^2$ es positiva, debemos de tener que λ es positiva.

Por lo tanto, todo autovalor λ de ***A*** es real y positiva.

Respuestas 3)

Suponiendo que la factorización LU de ***A*** existe, demuestre que,

a) (**Factorización** LDU) ***A*** se puede escribir como

$$A = LDU_1,$$

Donde D es diagonal y L y U_1 son triangular inferior con unos en la diagonal, y triangular superior respectivamente.

Demostración: Consideremos LDU y $L'D'U'$ como dos factorizaciones del tipo que estamos tratando actualmente, tendríamos como resultado la siguiente igualdad,

$$LDU = L'D'U',$$

Si multiplicamos por los inversos en los lugares apropiados, obtenemos lo siguiente,

$$L'^{-1}L = D'U'U^{-1}D^{-1}$$

Detallando que tanto U como D son invertibles dado a que cada uno ya se encuentra reducido por filas.

Obsérvese que el lado izquierdo es triangular inferior con unos en la diagonal, por lo que el lado derecho también debe serlo. Pero el lado derecho es triangular superior. La única matriz que es a la vez triangular inferior y superior es aquella matriz que sea diagonal, por lo que el lado izquierdo es la matriz diagonal con unos en la diagonal (la matriz identidad). Por tanto, tenemos que,

$$L = L'$$

Entonces, podemos poner las D en un lado y las U en el otro, y ahora ambas den ser I, por lo que las U son iguales, al igual que las D.

Quedando demostrado de esta manera que,

$$A = LDU\ (\boldsymbol{Factorizaci}ó\boldsymbol{n}\ LDU).$$

b) (**Factorización** LDL^T) Si $A = A^T$ entonces

$$A = LDL^T$$

Demostración: Supongamos que ***A*** tiene una descomposición de la forma LDL^T donde,

$$D = \begin{pmatrix} d_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & d_n \end{pmatrix}$$

Entonces, podemos definir lo siguiente,

$$\sqrt{D} = \begin{pmatrix} \sqrt{d_1} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \sqrt{d_n} \end{pmatrix}$$

Teniendo así que $(\sqrt{D})^2 = D$, y si tomamos $L_1 = L\sqrt{D}$. Entonces,

$$L_1 L_1^T = L(\sqrt{D})(\sqrt{D})^T L^T = LDL^T = A,$$

De esta manera, $L_1L_1^T$ es la descomposición de Cholesky de la matriz ***A***.

Por el contrario, teniendo la descomposición de Cholesky $A = L_1L_1^T$, podemos escribir que $L_1 = LD'$, donde D' es la matriz diagonal con las mismas entradas diagonales que la matriz L_1; entonces $L = L_1D'^{-1}$ es la matriz triangular inferior que obtenemos de L_1 al dividir cada columna por su entrada diagonal. Si hacemos $D = D'^2$, tenemos como resultado que,

$$A = (LD')(LD')^T = LD'^2L^T = LDL^T,$$

El cuál es la descomposición LDL^T de la matriz ***A***.

c) Usando lo anterior, establezca que, si ***A*** es simétrica y definida positiva, entonces

$$A = HH^T,$$

Donde H es triangular inferior con diagonal positiva. (**Esta es otra manera de deducir la factorización de Cholesky**).

Demostración: La prueba a realizar a continuación es por inducción en el orden de la matriz ***A.*** El resultado a obtener será cierto para matrices uno por uno, ya que a_{11} es positivo.

Supongamos que la afirmación es cierta para matrices de orden $n - 1$. Sea ***A*** una matriz simétrica definida positiva de orden n. Se puede dividir de la siguiente manera,

$$A = \begin{pmatrix} d & v^T \\ v & \bar{L} \end{pmatrix}$$

Donde d es un escalar positivo y $\bar{L}$ es una submatriz $n - 1$ por $n - 1$.

La matriz dividida, la podemos reescribir como un producto de matrices, quedando de la siguiente forma,

$$A = \begin{pmatrix} \sqrt{d} & 0 \\ \dfrac{v}{\sqrt{d}} & I_{n-1} \end{pmatrix} \begin{pmatrix} 1 & 0 \\ 0 & L \end{pmatrix} \begin{pmatrix} \sqrt{d} & \dfrac{v^T}{\sqrt{d}} \\ 0 & I_{n-1} \end{pmatrix}$$

Donde $L = \bar{L} - \frac{vv^T}{d}$. Es evidente que la matriz ***L*** es simétrica. También es definida positiva ya que para cualquier vector **x** no nulo de longitud $n-1$,

$$x^T L x = x^T(\bar{L} - \frac{vv^T}{d})x$$

$$= (-\frac{x^T v}{d}, x^T)\begin{pmatrix} d & v^T \\ v & \bar{L} \end{pmatrix}\begin{pmatrix} -\frac{x^T v}{d} \\ x \end{pmatrix}$$

Lo que implica que $x^T L x \geq 0$. Por la hipótesis de inducción, ***L*** tiene una factorización triangular $H_L H_L^T$ con diagonales positiva. Así, A puede llegar a ser expresada como,

$$A = \begin{pmatrix} \sqrt{d} & 0 \\ \frac{v}{\sqrt{d}} & I_{n-1} \end{pmatrix}\begin{pmatrix} 1 & 0 \\ 0 & H_L \end{pmatrix}\begin{pmatrix} 1 & 0 \\ 0 & H_L^T \end{pmatrix}\begin{pmatrix} \sqrt{d} & \frac{v^T}{\sqrt{d}} \\ 0 & I_{n-1} \end{pmatrix}$$

$$= \begin{pmatrix} \sqrt{d} & 0 \\ \frac{v}{\sqrt{d}} & L_H \end{pmatrix}\begin{pmatrix} \sqrt{d} & \frac{v^T}{\sqrt{d}} \\ 0 & H_L^T \end{pmatrix}$$

$$= HH^T$$

Concluyendo de esta manera que es cierto que,

$$A = HH^T.$$

Respuestas 4)

A continuación, se presentan las respectivas soluciones implementadas en el lenguaje de programación Octave, donde se explicará el proceso realizado para lleva a cabo cada una de las preguntas, además de adjuntar imágenes correspondientes a los fragmentos más relevantes del código fuente de cada uno de los ejercicios asignados para esta sección de la tarea.

a) Factorización de Cholesky:

Dividiremos esta subsección en dos (2) partes, una enfocada en la explicación de la función desarrollada para obtener la factorización de Cholesky y otra donde se mencionará a la función principal (**main**) en la cual se definen todas las matrices de ejemplo para demostrar el correcto funcionamiento del código fuente y sus respectivos resultados.

- Función **cholesky**: Como su nombre lo indica, esta función se encarga de realizar los cálculos necesarios para obtener la Factorización de Cholesky de una matriz ***A*** dada como entrada, junto con la **n** cantidad de filas (o columnas, dado a que estamos considerando que la matriz ***A*** es una matriz cuadrada nxn) que esta posea. Recordemos que esta factorización, busca reescribir la matriz ***A*** como una multiplicación de matrices de la siguiente manera,

  $$A = RR^T \ (*)$$

  Donde ***R*** es una matriz cuadrada triangular superior, es decir, una matriz nxn que solo tiene cero debajo de la diagonal principal, por la respectiva multiplicación de su transpuesta.

  Matemáticamente, si existe una matriz simétrica definida positiva, ***A***, entonces, existe una matriz simétrica triangular inferior, ***R***, de la misma dimensión que ***A***, resultando en lo que se indica en $(*)$.

  La matriz resultado, figura como la matriz de Cholesky de ***A***. Esta matriz actúa como la raíz cuadrada de la matriz ***A***. Sabemos que el dominio de la raíz cuadrada, es:

  $$\{x \in \mathcal{R}: x \geq 0\}$$

La cual está definida en todos los números reales no negativos. De la misma manera que la raíz cuadrada, la matriz de Cholesky solo existirá si la matriz está definida positiva.

De esta manera, y basándonos directamente en los diferentes pseudocódigos ofrecidos en las bibliografías ofrecidas en el material de estudio, se define la siguiente función **cholesky**, la cual retorna como resultado una matriz ***R***, que es el factor de la descomposición que nos encontramos trabajando.

```
function R = cholesky(A, n)
  % Create the empty R matrix
  R = zeros(n);
  % Fill the R matrix
  for i = 1:n
    for j = 1:n
      if(i == j)
        squares = 0;
        for k = 1:j-1
          squares = squares + (R(j,k))^2;
        endfor
        R(j,j) = sqrt(A(j,j) - squares);
      elseif(i > j)
        summation = 0;
        for k = 1:j-1
          summation = summation+ R(i,k)*R(j,k);
        endfor
        R(i,j) = ((A(i,j)-summation)/(R(j,j)));
      endif
    endfor
  endfor
endfunction
```

- Función **main**: La función **main** es el punto de partida de la ejecución del programa, la cual dentro de ella se define cuatro (4) matrices simétricas definida positiva para demostrar el funcionamiento adecuado de la función anteriormente mencionada. Además de ello, como se podrá observar, se realiza la impresión tanto de la matriz ***A***, como también la de la matriz ***R*** obtenida como también el resultado de realizar la operación $\boldsymbol{RR^T}$.

```
function main
  % EXAMPLE 1
  A = [1 0 1; 0 2 0; 1 0 3];

  % EXAMPLE 2
  %A = [6 15 55; 15 55 225; 55 225 979];

  % EXAMPLE 3
  %A = [25 15 -5; 15 18 0; -5 0 11];

  % EXAMPLE 4
  %format long
  %A = [5 1.2 0.3 -0.6; 1.2 6 -0.4 0.9; 0.3 -0.4 8 1.7; -0.6 0.9 1.7 10];

  disp("A = ");
  disp(A);

  R = cholesky(A, size(A,1));
  disp("R = ");
  disp(R);

  disp("RR' = ");
  disp(R * R');

endfunction
```

b) Factorización LU para un sistema $Ax = b$ en bandas diagonal dominante:

De manera similar a la pregunta anterior, se dividirá la explicación en dos (2) partes donde se explicará en una la función desarrollada **lu_solve** y la función **main**, de manera precisa y puntual de cómo trabajan cada una de estas, además adjuntar las correspondientes imágenes del fragmento del código fuente como apoyo a lo explicado.

- Función **lu_solve**: Como su nombre indica, esta función es la encargada de llevar a cabo la tarea de realizar la factorización LU de la matriz ***A*** pasada como argumento de entrada en la invocación de la función y retornará un vector de solución ***x*** del sistema de ecuaciones lineales que se esté procesando en ese instante, también se menciona que la función recibirá como segundo argumento de entrada, el vector ***b***, correspondiente al lado derecho de $Ax = b$.

  Dado a que estamos considerando que la matriz ***A*** indicada es de bandas diagonal dominante, la factorización LU, especialmente, el proceso de Eliminación Gaussiana, recibe algunas alteraciones en comparación a otras implementaciones que se han desarrollado en asignaciones anteriores de la materia.

  La principal alteración al procedimiento en comparación al realizado para matrices generales, es que, como se indica en la página 66 de ***Scientific Computing. An introductory Survery*** (*Michael T. Heath*), dado a que el pivotaje no es necesario para la estabilidad, lo que suele ser el caso de los sistemas tridiagonales que surgen en la práctica (por ejemplo, si la matriz es diagonalmente dominante o positiva definida), entonces la Eliminación Gaussiana se reduce al siguiente algoritmo sencillo:

$$
\begin{array}{l}
d_1 = b_1 \\
\textbf{for } i = 2 \textbf{ to } n \\
\qquad m_i = a_i / d_{i-1} \\
\qquad d_i = b_i - m_i c_{i-1} \\
\textbf{end}
\end{array}
$$

  Tomando en cuenta lo mencionado por el autor del libro, entonces, se desarrolló la función **lu_solve** de la siguiente manera,

```
function [L, U] = LU(A)
  % Obtain number of rows (should equal number of columns)
  n = size(A, 1);
  % Start L off as identity and populate the lower triangular half
slowly
  L = eye(n);
  for k = 1 : n
    % For each row k, access columns from k+1 to the end and divide by

    % the diagonal coefficient at A(k ,k)
    L(k + 1 : n, k) = A(k + 1 : n, k) / A(k, k);
    % For each row k+1 to the end, perform Gaussian elimination
    % In the end, A will contain U
    for l = k + 1 : n
        A(l, :) = A(l, :) - L(l, k) * A(k, :);
    end
  end
  U = A;
  y = L\b;
  x = U\y;
end
```

- Función **main**: Se toma como punto de partida para la ejecución del código fuente, donde en ella se encuentra comentada algunos ejemplos de sistemas de bandas diagonalmente dominante y así comprobar la correcta ejecución de la función explicada anteriormente. Además, se indica para imprimir por pantalla la matriz de coeficientes ***A*** y el vector derecho ***b***, para así, en especial para la matriz ***A*** comprobar que esta es una matriz de bandas diagonalmente dominante.

```
function main
  % EXAMPLE 1
  A = [5 2 1; 0 -3 2; -4 1 6];
  b = [1; 0; -1];

  % EXAMPLE 2
  %A = [10 2 3 0 0; 2 20 4 5 0; 3 4 30 6 7; 0 5 6 40 8; 0 0 7 8 50];
  %b = [2; 4 ; 5; -2; 1];

  % EXAMPLE 3
  %A = [7 2 0; 3 5 -1; 0 5 -6];
  %b = [1; 1; 1];

  disp("A = ");
  disp(A);
  disp("b = ");
  disp(b);

  x = LU(A, b);
  disp("Solution");
  disp("x = ");
  disp(x);

endfunction
```