Proyecto de Análisis Aplicado
Gradiente Conjugado Precondicionado
Dr. Zeferino Parada

1 Almacenamiento Comprimido por Renglones

Una matriz rala como la siguiente

$$A = \begin{pmatrix} 2 & -1 & 0 & 0 & 0 & 0 & 0 \\ -1 & 3 & -1 & 0 & 0 & 0 & 0 \\ 0 & -1 & 4 & -1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 5 & -1 & 0 & 0 \\ 0 & 0 & 0 & -1 & 6 & -1 & 0 \\ 0 & 0 & 0 & 0 & -1 & 7 & -1 \\ 0 & 0 & 0 & 0 & 0 & -1 & 8 \end{pmatrix},$$

se puede almacenar en una matriz de $2xm$ de la forma

$$AA = \begin{array}{|c|c||c||c|c|c||c||c|c|c|c||c||c|c||c||} \hline 2 & -1 & 0 & -1 & 3 & -1 & 0 & \dots & -1 & 7 & -1 & 0 & -1 & 8 & 0 \\ \hline 1 & 2 & 0 & 1 & 2 & 3 & 0 & \dots & 5 & 6 & 7 & 0 & 6 & 7 & 0 \\ \hline \end{array}$$

.

Cada columna diferente de cero de AA corresponde a la entrada, $a_{i,j} \neq 0$, tal que

$$\begin{array}{|c|} \hline a_{i,j} \\ \hline j \\ \hline \end{array}$$

Una columna cero en AA indica que en el renglón i se han guardado todos sus elementos diferentes de cero.

La matriz AA es de orden $2xm$ donde $m = nz + n$ y nz es el número de elementos diferentes de cero en A y n es la dimensión de A.

Para matrices densas, éste guardado no tiene sentido.

Escriba en Matlab la función :

function $[AA,\ ind] = almcomren(A)$
% Se guarda la matriz rala A por almacenamiento comprimido por renglones.
% El vector de índices, ind, contiene el número de las columnas de AA que son ceros.

El producto matriz por vector, $A * x$, puede realizarse con la matriz AA ya que

$$(A * x)_i = \sum_{j=1}^{n} a_{i,j} x_j = \sum_{a_{i,j} \neq 0} a_{i,j} x_j.$$

Construya un programa en Matlab de la forma

function $[v] = matporvec(AA,\ ind,\ \ x)$
% Se realiza el producto matriz por vector $v = A * x$ donde A se guarda
% en AA según la función *almcomren.m*

2 Proyecto

Crear una función en Matlab del algoritmo de gradiente conjugado y gradiente conjugado precondicionado donde la matriz A se almacena por compresión por renglones y el producto $A * x$ se hace con la función *matporvec.m.*

function $[x,\ iter,\ band,\ \ vecres] =$ GC $(AA,\ ind,\ b)$
% Método de gradiente conjugado con matriz almacenada por compresión
% en los renglones.
% El vector x es la aproximación a la solución.
% El índice $iter$ es el número de iteraciones.
% El vector $vecres$ es el vector de residuales $\|A * x_k - b\|_2$.
% La tolerancia para el residual es $tol = 1.e - 06$.
% El número máximo de iteraciones es $maxiter = n$.
% El valor $band$ indica

% $band == 1$ si y sólo si $\|A * x - b\|_2 \leq tol$.
% $band == 0$ si y sólo si $iter = maxiter$ y $\|A * x - b\|_2 > tol$.

function $[x,\ iter,\ band,\ vecres] =$ GCPre $(AA,\ ind,\ C,\ b)$
% Se resuelve el sistema lineal, $A * x = b$ por medio de gradiente conjugado.
% usando el precondicionador $M = C^T C$.

El Precondiconador será el de Jacobi. Puede usar el hecho de que la matriz C es diagonal y guardar solamente un vector VC y resolver los sistemas lineales

$$(C^T C)q = y,$$

en forma directa.

2.1 Matrices de Prueba

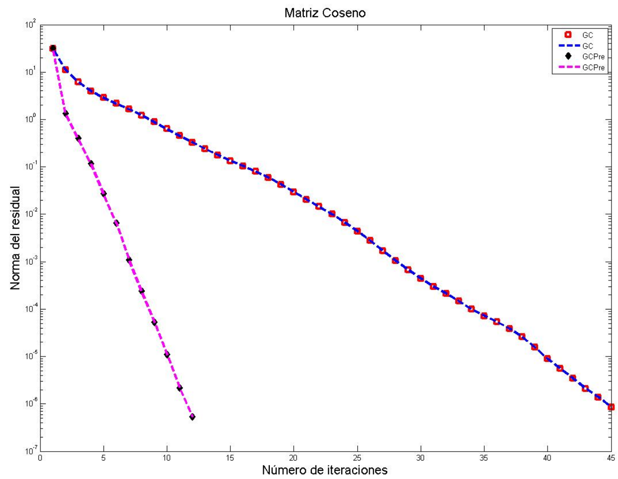
1. Matriz coseno.

$$a_{i,j} = \begin{cases} \sqrt{i} & \text{si } i = j \\ cos\ i & \text{si } j = i + 10 \ \text{ o } i = j + 10 \end{cases}$$

con $n = 1000$.

El vector del lado derecho es $b = ones(n.\ 1)$.

El script **pruebamatcoseno.m** tiene por salida la siguiente gráfica de comparación de los vectores residuales:

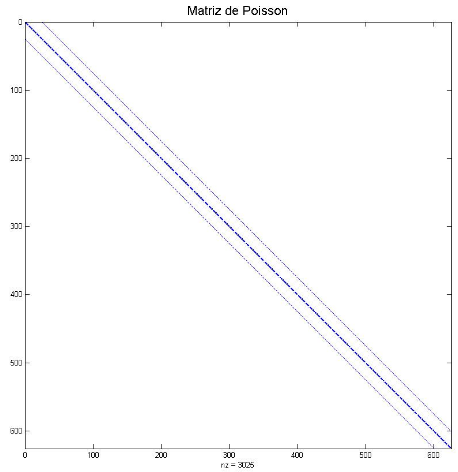


El precondicinamiento resuelve el problema en doce iteraciones. La escala sobre el vector residual es logarítmica.

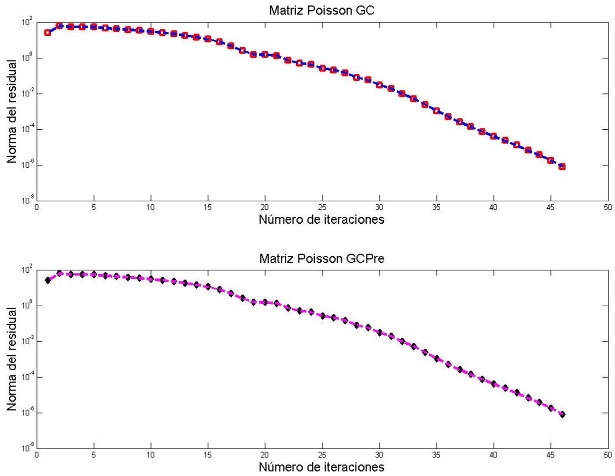
Los tiempos de máquina para la matriz coseno son :

almcomren.m	0.051306 seconds.
GC.m	0.216907 seconds.
GCPre.m	1.0153041 seconds

2. Matrices de Poisson con $q = 25$, recuerde que en este caso $n = q^2 = 625$.



La gráfica que se obtiene es:



En este caso el precondicionador C no ayuda en la solución.

Los tiempos de máquina para la matriz de Poisson son :

almcomren.m	0.059956 seconds.
GC.m	0.353699 seconds.
GCPre.m	0.287787 seconds

Qué entregar.

1. Funciones **almcomren.m, matporvec.m, GC.m, GCPre.m** y los script files **pruebapoisson.m**, y **pruebamatcoseno** Los script deben tener como salida: la gráfica y los tiempos de máquina. de cada parte.

2. Los programas computacionales se empaquetan y se envían al correo electrónico: **zeferino@itam.mx**.

3. **Fecha y hora final de entrega:** 22 de mayo de 2019.