MÉTODOS DE REDUCCIÓN Y MÉTODO DE BUNEMAN

Dr. OSCAR ROJO JERALDO*

1. INTRODUCCION.

Consideremos el sistema de ecuaciones lineales

$$(1.1) \quad Mx = b,$$

donde

$$M = \begin{bmatrix} A & T & & & \bigcirc \\ T & A & T & & \\ & T & A & T & \\ & & T & A & T \\ \bigcirc & & & T & A \end{bmatrix} \quad \text{de orden } (pq) \times (pq),$$

con A y T matrices reales simétricas de orden $p \times p$ y tales que conmutan, es decir, $AT = TA$.

* Profesor Departamento de Matemáticas. Facultad de Ciencias.
Universidad del Norte.

A este tipo de sistemas de ecuaciones conduce la discretización de ciertas ecuaciones diferenciales elípticas. En particular, en el caso de la ecuación de Poisson en un rectángulo y con condiciones de Dirichlet en la frontera, el uso de la fórmula de cinco puntos para aproximar al laplaciano conduce a un sistema de ecuaciones en el cual

$$A = \begin{bmatrix} -4 & 1 & & & & & \bigcirc \\ 1 & -4 & 1 & & & & \\ & 1 & -4 & 1 & & & \\ & & \cdot & \cdot & \cdot & & \\ \bigcirc & & & \cdot & \cdot & 1 & \\ & & & & 1 & -4 & \end{bmatrix} \quad \text{y } T = I_p$$

El presente trabajo tiene por finalidad exponer el algoritmo FACR (ℓ), el método de reducción cíclica par e impar y el método de Buneman. Estos métodos son directos y han sido desarrollados para la resolución de (1.1). Se caracterizan por el hecho de realizar ciclos de reducción en el sistema. En el algoritmo FACR (ℓ), los ciclos realizados son seguidos por el método de descomposición matricial para resolver el sistema de ecuaciones reducido. En cambio el método de reducción cíclica par e impar y el método de Buneman agotan los ciclos de reducción. Estudios teóricos y prácticos han demostrado que el método de reducción cíclica par e impar presenta severos errores de redondeo. El método de Buneman, matemáticamente equivalente al proceso de reducción del método de reducción cíclica, presenta la ventaja de ser estable desde el punto de vista numérico. Este trabajo finaliza con una aplicación del método de Buneman a la ecuación de Poisson con condiciones de Dirichlet en la frontera.

2. CICLO DE REDUCCION.

Sea $q = m-1$ con $m = 2^{k+1}$. Particionando adecuadamente, el sistema (1.1) puede escribirse en la forma:

$$(2.2) \quad \begin{bmatrix} A & T & & & \\ T & A & T & & \\ & T & A & T & \\ & & & \ddots & \\ \bigcirc & & & & T \\ & & & T & A \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ \vdots \\ x_{m-1} \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ \vdots \\ b_{m-1} \end{bmatrix}$$

Definiendo $x_0 = x_m = 0$, de (2.2) y para $2 \leq j \leq m-2$, podemos escribir

$$(2.3) \quad \begin{cases} Tx_{j-2} + Ax_{j-1} + Tx_j & = b_{j-1} \\ Tx_{j-1} + Ax_j + Tx_{j+1} & = b_j \\ Tx_j + Ax_{j+1} + Tx_{j+2} & = b_{j+1} \end{cases}$$

Multiplicando la primera y tercera ecuación de (2.3) por T y la ecuación central por $-A$, después de sumar los productos obtenemos.

$$(2.4) \quad T^2 x_{j-2} + (2T^2 - A^2)x_j + T^2 x_{j+2} = T(b_{j-1} + b_{j+1}) - Ab_j$$

Considerando valores pares para j , es decir, $j = 2, 4, 6, \dots, m-2$ resulta de (2.4) el sistema de ecuaciones siguiente:

$$(2.5) \quad \begin{bmatrix} (2T^2 - A^2) & T^2 & & & \bigcirc \\ & T^2 & (2T^2 - A^2) & T^2 & \\ & & & \ddots & \\ & & & & T^2 \\ \bigcirc & & & T^2 & (2T^2 - A^2) \end{bmatrix} \begin{bmatrix} x_2 \\ x_4 \\ \vdots \\ \vdots \\ x_{m-2} \end{bmatrix} = \begin{bmatrix} T(b_1 + b_3) - Ab_2 \\ T(b_3 + b_5) - Ab_4 \\ \vdots \\ \vdots \\ T(b_{m-3} + b_{m-1}) - Ab_{m-2} \end{bmatrix}$$

Vemos que (2.5) involucra sólo incógnitas de índice par. El proceso seguido para reducir las ecuaciones es conocido como ciclo de reducción. Observamos que en este proceso se ha hecho uso de la conmutatividad de las matrices A y T. Las ecuaciones restantes, llamadas ecuaciones eliminadas, dan origen al sistema de ecuaciones:

$$(2.6) \quad \begin{bmatrix} A & 0 & \bigcirc \\ 0 & A & \vdots \\ \vdots & \vdots & \vdots \\ \bigcirc & \vdots & A \end{bmatrix} \begin{bmatrix} x_1 \\ x_3 \\ \vdots \\ x_{m-1} \end{bmatrix} = \begin{bmatrix} b_1 - Tx_2 \\ b_3 - Tx_2 - Tx_4 \\ \vdots \\ b_{m-1} - Tx_{m-2} \end{bmatrix}$$

El sistema de ecuaciones (2.6) puede escribirse abreviadamente como sigue:

$$(2.7) \quad Ax_j = b_j - Tx_{j-1} - Tx_{j+1}, \quad j = 1, 3, 5, \dots, m-1$$

Después de resolver el sistema de ecuaciones (2.5), las restantes incógnitas pueden ser obtenidas resolviendo los sistemas de ecuaciones p-dimensionales dados en (2.7).

Dado que $m=2^{k+1}$, el sistema (2.5) tiene una dimensión en bloques igual a 2^{k-1} . Además, claramente la conmutatividad de las matrices A y T implica que las matrices $(2T^2 - A^2)$ y T^2 conmutan. Se muestra así que el sistema (2.5) satisface las mismas hipótesis que (1.1). Podemos entonces aplicar ahora al sistema (2.5) el ciclo de reducción antes empleado.

3. METODOS DE REDUCCION CICLICA PAR E IMPAR.

El proceso de reducción antes descrito puede claramente ser repetido, obteniendo después de k ciclos de reducción un sistema de ecuaciones con un único bloque y cuya incógnita x_{2^k} puede ser obtenida resolviendo este sistema de ecuaciones. Después se sustituyen las incógnitas ya calculadas en los restantes sistemas de ecuaciones (ecuaciones eliminadas de cada ciclo de reducción). El método entero puede ser descrito recursivamente como sigue:

$$\begin{aligned} \text{Sean } A^{(0)} &= A, \quad T^{(0)} = T \\ b_j^{(0)} &= b_j \quad \text{para } j = 1, 2, \dots, m-1 \end{aligned}$$

Para $r = 1, 2, \dots, k$ y $j = 2^r, 2 \cdot 2^r, 3 \cdot 2^r, \dots, 2^{k+1} - 2^r$, se definen

$$(3.1) \quad \begin{cases} A^{(r)} &= 2(T^{(r-1)})^2 - (A^{(r-1)})^2 \\ T^{(r)} &= (T^{(r-1)})^2 \\ b_j^{(r)} &= T^{(r-1)} \left(b_{j-2^{r-1}}^{(r-1)} + b_{j+2^{r-1}}^{(r-1)} \right) - A^{(r-1)} b_j^{(r-1)} \end{cases}$$

Después, para $r = k, k-1, k-2, \dots, 0$, resolver sucesivamente:

$$(3.2) \quad \begin{aligned} A^{(r)} x_j &= b_j^{(r)} - T^{(r)} (x_{j+2^r} + x_{j-2^r}) \\ \text{para } j &= 2^r, 3 \cdot 2^r, 5 \cdot 2^r, \dots, 2^{k+1} - 2^r. \end{aligned}$$

Este método es conocido como método de reducción cíclica par e impar o también como el algoritmo COR.

Si A es una matriz de banda (por ejemplo, una matriz tridimensional), el ancho de las matrices de banda $A^{(r)}$ crece con r . Entonces, en tal caso la determinación directa de estas matrices requiere de

multiplicaciones matriciales extensas. Para evitar esto las matrices $A^{(r)}$ son expresadas como un producto de matrices más simples. Para este propósito, comenzamos observando que $A^{(1)}$ es un polinomio de grado dos en las matrices A y T . Por inducción puede ser probado que $A^{(r)}$ es un polinomio de grado 2^r en A y T , polinomio que puede escribirse en la forma

$$(3.3) \quad A^{(r)} = P_{2^r}(A, T) = \sum_{j=0}^{2^{r-1}} c_{2^j}^{(r)} A^{2^j} T^{2^r-2^j}$$

donde

$$c_{2^r}^{(r)} = -1. \quad \text{Sea}$$

$$p_{2^r}(a, t) = \sum_{j=0}^{2^{r-1}} c_{2^j}^{(r)} a^{2^j} t^{2^r-2^j}$$

Deseamos expresar este polinomio como un producto de factores lineales en a y t . Para $t \neq 0$ definimos:

$$\frac{a}{t} = 2 \cos \theta$$

Probaremos a continuación, usando inducción en r , que

$$(3.4) \quad p_{2^r}(a, t) = -2t^{2^r} \cos 2^r \theta$$

Para $r = 1$ tenemos

$$\begin{aligned} p_2(a, t) &= 2t^2 - a^2 = t^2 \left(2 - \frac{a^2}{t^2} \right) = \\ &= t^2 (2 - 4 \cos^2 \theta) = -2t^2 (2 \cos^2 \theta - 1) \\ &= -2t^2 \cos 2\theta \end{aligned}$$

Supongamos ahora que (3.4) se cumpla. De las relaciones

$$A^{(r+1)} = 2(T^{(r)})^2 - (A^{(r)})^2$$

$$T^{(r+1)} = (T^{(r)})^2$$

se sigue que

$$p_{2^{r+1}}(a, t) = 2t^{2^{r+1}} - (p_{2^r}(a, t))^2$$

Entonces,

$$\begin{aligned} p_{2^{r+1}}(a, t) &= 2t^{2^{r+1}} - (-2t^{2^r} \cos 2^r \theta)^2 = \\ &= -2t^{2^{r+1}} (2 \cos^2 2^r \theta - 1) = \\ &= -2t^{2^{r+1}} \cos 2^{r+1} \theta. \end{aligned}$$

Con lo cual queda probado la validez de (3.4) para todo r . Por lo tanto,

$$p_{2^r}(a, t) = 0 \text{ si y sólo si } \cos 2^r \theta = 0.$$

En consecuencia, p_{2^r} se anula si y sólo si

$$\theta = \theta_j^{(r)} = (2j-1) \frac{\pi}{2^{r+1}} \text{ con } j = 1, 2, 3, \dots, 2^r.$$

De este modo hemos establecido que

$$p_{2^r}(a, t) = -t^{2^r} \pi \sum_{j=1}^{2^r} \left(\frac{a}{t} - (-2 \cos \theta_j^{(r)}) \right).$$

Es decir,

$$p_{2^r}(a, t) = - \pi \sum_{j=1}^{2^r} (a + 2t \cos \theta_j^{(r)}).$$

Queda así probado que

$$A^{(r)} = - \pi \sum_{j=1}^{2^r} (A + 2 \cos \theta_j^{(r)} T), \quad r = 1, 2, 3, \dots, k,$$

$$\text{con } \theta_j^{(r)} = (2j-1) \frac{\pi}{2^{r+1}}$$

La importancia de esta factorización radica en el hecho de que ella puede ser empleada para resolver los sistemas de ecuaciones dados en (3.2) y en el cómputo de $b_j^{(r)}$ de (3.1). De este modo, la resolución de los sistemas de ecuaciones como también las multiplicaciones matriciales son limitadas a matrices de banda especiales (por ejemplo, tri diagonales especiales). El método recién descrito es llamado método de reducción par e impar con factorización o, más brevemente, como el algoritmo CORF.

Queremos a continuación mostrar como emplear la factorización de $A^{(r)}$ en la resolución de los sistemas de ecuaciones dados en (3.2). Por ejemplo, para el sistema

$$A^{(k)} x_{2^k} = b_{2^k}^{(k)},$$

después de definir $G_j^{(k)} = A + 2 \cos \theta_j^{(k)} T$ y $z_1 = -b_{2^k}^{(k)}$, resolver iterativamente los sistemas especiales

$$G_j^{(k)} z_{j+1} = z_j, \quad j = 1, 2, \dots, 2^k,$$

resultando $z_{2^k+1} = x_{2^k}$.

Un algoritmo similar puede ser usado para resolver los sistemas de las ecuaciones eliminadas con

$$A^{(r)} = -\pi \sum_{k=1}^{2^r} G_j^{(r)}.$$

Como indicáramos anteriormente la factorización de $A^{(r)}$ puede también ser usada para computar $b^{(r)}$ dado en (3.1). Sin embargo, un procedimiento alternativo puede ser encontrado al hacer uso de la naturaleza recursiva de los polinomios $p_{2^r}^s(a, t)$. Sea

$$p_s(a, t) = -2t^s \cos(s\theta), \quad \frac{a}{t} = -2 \cos \theta, \quad t \neq 0$$

Entonces, $p_0(a, t) = -2$ y $p_1(a, t) = a$.

Para $s \geq 2$ tenemos

$$\begin{aligned}
 -a p_{s-1}(a, t) - t^2 p_{s-2}(a, t) &= a 2t^{s-1} \cos(s-1)\theta + t^2 2t^{s-2} \cos(s-2)\theta = \\
 &= 2t^{s-1} (a \cos(s-1)\theta + t \cos(s-2)\theta) = \\
 &= 2t^{s-1} (-2t \cos\theta \cos(s-1)\theta + t \cos(s-2)\theta) = \\
 &= -2t^s (2\cos\theta \cos(s-1)\theta - \cos(s-2)\theta) = \\
 &= -2t^s (\cos(s\theta) + \cos(s-2)\theta - \cos(s-2)\theta) = \\
 &= -2t^s \cos(s\theta).
 \end{aligned}$$

Por lo tanto,

$$p_s(a, t) = -a p_{s-1}(a, t) - t^2 p_{s-2}(a, t) \text{ con}$$

$$p_0(a, t) = -2, p_1(a, t) = a$$

De (3.1) vemos que el cómputo de $b^{(r)}$ requiere del cómputo del producto $A^{(r-1)} b^{(r-1)}$. La fórmula de recursión recién encontrada para los polinomios $p_s(a, t)$ puede ser empleada en el cómputo de dichos productos. En efecto, haciendo

$$n_0 = -2b_j^{(r-1)}, n_1 = Ab_j^{(r-1)}$$

$$n_s = -An_{s-1} - T^2 n_{s-2}, s = 2, 3, \dots, 2^{r-1}$$

$$\text{resulta } n_{2^{r-1}} = P_{2^{r-1}}(A, T) b_j^{(r-1)} = A^{(r-1)} b_j^{(r-1)}.$$

Estudios teóricos y prácticos prueban que el cómputo de $b^{(r)}$ está sujeto a severos errores de redondeo. Sin embargo, las fórmulas hasta ahora encontradas son necesarias para el desarrollo del algoritmo

mo de Buneman, método estable desde el punto de vista numérico. Antes de estudiar el método de Buneman, deseamos decir algunas palabras relativas al algoritmo FACR (ℓ)

4. EL ALGORITMO FACR (ℓ)

Los métodos de reducción cíclica par e impar agotan los ciclos de reducción indicados en la sección 2, es decir, llevan el proceso de reducción hasta tener solo un bloque. En el algoritmo FACR (ℓ), en cambio, el proceso de reducción es detenido en algún momento, usando después el método de descomposición matricial para resolver el sistema de ecuaciones resultante. Después de r ciclos de reducción tenemos el sistema de ecuaciones

$$(4.1) \quad M^{(r)} Z^{(r)} = b^{(r)},$$

$$M^{(r)} = \begin{bmatrix} A^{(r)} & T^{(r)} & & & & & \bigcirc \\ T^{(r)} & A^{(r)} & & & & & \\ & & \ddots & & & & \\ & & & \ddots & & & \\ & & & & \ddots & & \\ & & & & & T^{(r)} & \\ \bigcirc & & & & & T^{(r)} & A^{(r)} \end{bmatrix}, \quad Z^{(r)} = \begin{bmatrix} x \\ 2^r \\ x \\ 2 \cdot 2^r \\ \vdots \\ x \\ j \cdot 2^r \\ \vdots \end{bmatrix}, \quad b^{(r)} = \begin{bmatrix} b^{(r)} \\ 2^r \\ b^{(r)} \\ 2 \cdot 2^r \\ \vdots \\ b^{(r)} \\ j \cdot 2^r \\ \vdots \end{bmatrix}$$

El uso del método de descomposición matricial para resolver el sistema tridiagonal en bloque (4.1) requiere de la determinación de los autovalores de las matrices $A^{(r)}$ y $T^{(r)}$. La conmutatividad de las matrices A y T implica la existencia de una matriz ortogonal Q tal que

$$(4.2) \quad Q^T A Q = \Lambda, \quad Q^T T Q = \Omega$$

donde las matrices Λ y Ω son matrices diagonales. Es decir, las matrices A y T tienen un conjunto de autovectores linealmente independientes en común, siendo los elementos de la diagonal de Λ los autovalores de A y aquellos en la diagonal de Ω los autovalores de T .

De (4.2) podemos escribir

$$A^{(0)} = Q \Lambda^{(0)} Q^T, \quad T^{(0)} = Q \Omega^{(0)} Q^T$$

donde

$$\Lambda^{(0)} = \Lambda \quad y \quad \Omega^{(0)} = \Omega$$

Para $r \geq 0$ definimos

$$(4.3) \quad \Lambda^{(r)} = Q^T A^{(r)} Q \quad y \quad \Omega^{(r)} = Q^T T^{(r)} Q$$

De (3.1) obtenemos, para $r \geq 1$,

$$(4.4) \quad \begin{cases} \Lambda^{(r)} = 2(\Omega^{(r-1)})^2 - (\Lambda^{(r-1)})^2 \\ \Omega^{(r)} = (\Omega^{(r-1)})^2. \end{cases}$$

Fácilmente se puede deducir de (4.4) que las matrices $\Lambda^{(r)}$ y $\Omega^{(r)}$ son diagonales. Usando ahora (4.3) vemos que los elementos en las diagonales de estas matrices son los autovalores de $A^{(r)}$ y $T^{(r)}$, siendo las columnas de Q un conjunto común de autovectores. Por lo tanto, los autovalores de $A^{(r)}$ y $T^{(r)}$ están dados por las fórmulas

$$\lambda_i^{(r)} = 2(\omega_i^{(r-1)})^2 - (\lambda_i^{(r-1)})^2,$$

$$\omega_i^{(r)} = (\omega_i^{(r-1)})^2, \quad \text{para } i = 1, 2, \dots, p;$$

$$\text{donde } \lambda_i^{(0)} = \lambda_i, \quad \omega_i^{(0)} = \omega_i \quad y$$

$$\Lambda = \text{diag}\{\lambda_1, \lambda_2, \dots, \lambda_p\}, \quad \Omega = \text{diag}\{\omega_1, \omega_2, \dots, \omega_p\}$$

La determinación de estos autovalores permite la aplicación del método de descomposición matricial en la resolución de (4.1).

Hockney denominó a este método como el algoritmo FACR (ℓ), donde ℓ es el número de ciclos realizados.

5. EL METODO DE BUNEMAN.

Como señaláramos previamente, varias investigaciones, tanto teóricas como prácticas, han demostrado que los algoritmos COR y CORF son numéricamente inestables. Buneman desarrolló una estabilización basada en una reformulación en el cómputo de los vectores $b^{(r)}$. Suponemos $T = I$. Sea

$$p_j^{(0)} = 0, \quad q_j^{(0)} = b_j, \quad j = 1, 2, 3, \dots, 2^{k+1} - 1$$

Para $r = 1, 2, 3, \dots, k$ definimos

$$(5.1) \quad \begin{cases} p_j^{(r)} = p_j^{(r-1)} - (A^{(r-1)})^{-1} \left[p_{j-2^{r-1}}^{(r-1)} + p_{j+2^{r-1}}^{(r-1)} - q_j^{(r-1)} \right] \\ q_j^{(r)} = q_{j-2^{r-1}}^{(r-1)} + q_{j+2^{r-1}}^{(r-1)} - 2p_j^{(r)} \end{cases}$$

$$\text{con } j = 2^r, 2 \cdot 2^r, 3 \cdot 2^r, \dots, 2^{k+1} - 2^r.$$

Las secuencias $p_j^{(r)}$, $q_j^{(r)}$ y $b_j^{(r)}$ satisfacen la relación siguiente:

$$(5.2) \quad b_j^{(r)} = A^{(r)} p_j^{(r)} + q_j^{(r)}$$

$$\text{para } r = 0, 1, \dots, k \text{ y } j = 2^r, 2 \cdot 2^r, 3 \cdot 2^r, \dots, 2^{k+1} - 2^r$$

En efecto, para $r = 0$ tenemos $b_j^{(0)} = b_j$. Supongamos que (5.2) se cumple para r . Entonces, usando (3.1) con $T = I$ y (5.1),

$$\begin{aligned} A^{(r+1)} p_j^{(r+1)} + q_j^{(r+1)} &= 2 p_j^{(r+1)} - (A^{(r)})^{-1} 2 p_j^{(r+1)} + q_j^{(r+1)} = \\ &= 2 p_j^{(r+1)} - (A^{(r)})^{-1} 2 p_j^{(r+1)} + q_{j-2^r}^{(r)} + q_{j+2^r}^{(r)} - 2 p_j^{(r+1)} = \end{aligned}$$

$$\begin{aligned}
&= - (A^{(r)})^2 p_j^{(r)} + A^{(r)} \left[p_{j-2}^{(r)} + p_{j+2}^{(r)} - q_j^{(r)} \right] + q_{j-2}^{(r)} + q_{j+2}^{(r)} = \\
&= A^{(r)} p_{j-2}^{(r)} + q_{j-2}^{(r)} + A^{(r)} p_{j+2}^{(r)} + q_{j+2}^{(r)} - A^{(r)} \left[A^{(r)} p_j^{(r)} + q_j^{(r)} \right] = \\
&= b_{j-2}^{(r)} + b_{j+2}^{(r)} - A^{(r)} b_j^{(r)} = \\
&= b_j^{(r+1)}
\end{aligned}$$

Con lo cual (5.2) queda demostrado. De (5.1) tenemos

$$A^{(r-1)} (p_j^{(r-1)} - p_j^{(r)}) = p_{j-2}^{(r-1)} + p_{j+2}^{(r-1)} - q_j^{(r-1)}$$

Con este resultado el algoritmo de Buneman puede ser resumido como si que:

a) Fase de reducción.

Computar las sucesiones $p_j^{(r)}$ y $q_j^{(r)}$:

$$p_j^{(0)} = 0, \quad q_j^{(0)} = b_j, \quad j = 1, 2, 3, \dots, 2^{k+1} - 1.$$

$$A^{(r-1)} (p_j^{(r-1)} - p_j^{(r)}) = p_{j-2}^{(r-1)} + p_{j+2}^{(r-1)} - q_j^{(r-1)}$$

$$p_j^{(r)} = p_j^{(r-1)} - (p_{j-2}^{(r-1)} + p_{j+2}^{(r-1)} - q_j^{(r-1)})$$

$$q_j^{(r)} = q_j^{(r-1)} + p_{j-2}^{(r-1)} + p_{j+2}^{(r-1)} - 2p_j^{(r)}$$

para $r = 1, 2, 3, \dots, k$ y $j = 2^r, 2 \cdot 2^r, 3 \cdot 2^r, \dots, 2^{k+1} - 2^r$.

b) Fase de solución.

Resolver los sistemas de ecuaciones (3.2) usando la relación (5.2):

$$A^{(r)} (x_j - p_j^{(r)}) = q_j^{(r)} - x_{j+2^r} - x_{j-2^r}$$

$$x_j = p_j^{(r)} + (x_j - p_j^{(r)})$$

para $r = k, k-1, \dots, 0$, $j = 2^r, 3 \cdot 2^r, 5 \cdot 2^r, \dots, 2^{k+1} - 2^r$.

En ambas fases deben resolverse sistemas de ecuaciones cuya matriz de coeficientes es $A^{(r)}$. Estos sistemas son resueltos usando la factorización

$$A^{(r)} = - \prod_{j=1}^{2^r} (A - 2 \cos \theta_j^{(r)} I), \text{ con}$$

$$\theta_j = (2j-1) \frac{\pi}{2^{r+1}}, \quad r = 1, 2, 3, \dots, k$$

$$A^{(0)} = A.$$

Queremos ahora, antes de aplicar el método de Buneman a la ecuación de Poisson con las condiciones de Dirichlet, mostrar todos los pasos que contiene este método para el caso en el cual $q = 7$, es decir, en la resolución del sistema

$$\underline{j = 6}, \quad A^{(0)}(p_6^{(0)} - p_6^{(1)}) = p_5^{(0)} + p_7^{(0)} - q_6^{(0)}$$

$$p_6^{(0)} = p_6^{(0)} - (p_6^{(0)} - p_6^{(1)})$$

$$q_6^{(1)} = q_5^{(0)} + q_7^{(0)} - 2p_6^{(1)}$$

Paso 3: $r = 2$.

$$\underline{j = 4}, \quad A^{(1)}(p_4^{(1)} - p_4^{(2)}) = p_2^{(1)} + p_6^{(1)} - q_4^{(1)}$$

$$p_4^{(2)} = p_4^{(1)} - (p_4^{(1)} - p_4^{(2)})$$

$$q_4^{(2)} = q_2^{(1)} + q_6^{(1)} - 2p_4^{(1)}$$

Fase de Solución:

Paso 1: $r = 2$.

$$\underline{j = 4}, \quad A^{(2)}(x_4 - p_4^{(2)}) = q_4^{(2)} - x_8 - x_0 = q_4^{(2)}$$

$$x_4 = p_4^{(2)} + (x_4 - p_4^{(2)})$$

Paso 2 : $r = 1$.

$$\underline{j = 2}, \quad A^{(1)}(x_2 - p_2^{(1)}) = q_2^{(1)} - x_4 - x_0 = q_2^{(1)} - x_4$$

$$x_2 = p_2^{(1)} + (x_2 - p_2^{(1)})$$

$$\underline{j = 6}, \quad A^{(1)}(x_6 - p_6^{(1)}) = q_6^{(1)} - x_8 - x_4 = q_6^{(1)} - x_4$$

$$x_6 = p_6^{(1)} + (x_6 - p_6^{(1)})$$

Paso 3: $r = 0$.

$$\underline{j = 1}, \quad A^{(0)}(x_1 - p_1^{(0)}) = q_1^{(0)} - x_2 - x_0 = q_1^{(0)} - x_2$$

$$x_1 = p_1^{(0)} + (x_1 - p_1^{(0)})$$

$$\underline{j = 3}, \quad A^{(0)}(x_3 - p_3^{(0)}) = q_3^{(0)} - x_4 - x_2$$

$$x_3 = p_3^{(0)} + (x_3 - p_3^{(0)})$$

$$\underline{j = 5}, \quad A^{(0)}(x_5 - p_5^{(0)}) = q_5^{(0)} - x_6 - x_4$$

$$x_5 = p_5^{(0)} + (x_5 - p_5^{(0)})$$

$$\underline{j = 7}, \quad A^{(0)}(x_7 - p_7^{(0)}) = q_7^{(0)} - x_8 - x_6 = q_7^{(0)} - x_6$$

$$x_7 = p_7^{(0)} + (x_7 - p_7^{(0)}).$$

Debido al especial orden en los cálculos, los vectores $p_j^{(r)}$ y $q_j^{(r)}$ pueden ocupar los lugares que ocupaban los vectores $p_j^{(0)}$ y $q_j^{(0)}$ en razón de que estos no vuelven a ser ocupados. Similarmente, en la fase de solución, los vectores x_j pueden ir ocupando las posiciones que ocupaban los vectores $q_j^{(r)}$. El siguiente cuadro ilustra el caso $q = 7$.

Fase de Reducción. Fase de Solución.

6. APLICACION DEL METODO DE BUNEMAN A LA ECUACION DE POISSON EN UN RECTANGULO CON LAS CONDICIONES DE DIRICHLET.

Consideremos la ecuación

$$\Delta u(x,y) = f(x,y) , \quad (x,y) \in R$$

$$u(x,y) = g(x,y) , \quad (x,y) \in \partial R$$

donde R es un rectángulo de lados paralelos a los ejes coordenados. Una partición de este rectángulo en cuadrados de lados h y el uso de la fórmula de cinco puntos para aproximar al laplaciano conduce a la ecuación de diferencias

$$-4v_{ij} + v_{(i-1)j} + v_{(i+1)j} + v_{i(j-1)} + v_{i(j+1)} = f_{ij} h^2,$$

con $1 \leq i \leq n-1$, $1 \leq j \leq m-1$ y donde v_{ij} es una aproximación para $u(x_i, y_j)$ y

$$v_{0j} = g(x_0, y_j) , \quad v_{nj} = g(x_n, y_j)$$

$$v_{i0} = g(x_i, y_0) , \quad v_{im} = g(x_i, y_m)$$

$$f_{ij} = f(x_i, y_j).$$

De este modo obtenemos el sistema de ecuaciones

$$(6.1) \quad \begin{bmatrix} A & I & & & & & \\ I & A & I & & & & \\ & & \cdot & \cdot & \cdot & & \\ & & & \cdot & \cdot & \cdot & \\ & \bigcirc & & & \cdot & \cdot & I \\ & & & & & I & A \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \\ \cdot \\ \cdot \\ \cdot \\ v_{m-1} \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ \cdot \\ \cdot \\ \cdot \\ b_{m-1} \end{bmatrix},$$

$$A = \begin{bmatrix} -4 & 1 & & & \bigcirc \\ 1 & -4 & 1 & & \\ & 1 & -4 & 1 & \\ & & \cdot & \cdot & \cdot \\ \bigcirc & & & \cdot & 1 \\ & & & 1 & -4 \end{bmatrix}, \quad v_j = \begin{bmatrix} v_{1j} \\ v_{2j} \\ \cdot \\ \cdot \\ \cdot \\ v_{(n-1)j} \end{bmatrix}, \quad 1 \leq j \leq m-1$$

A es una matriz de orden $(n-1) \times (n-1)$. Los vectores b_j , $1 \leq j \leq m-1$, contienen a $h^2 f(x,y)$ como también los valores de $g(x,y)$ en ∂R .

Si tomamos $m = 2^{k+1}$ para cierto $k \in \mathbb{N}$ podemos aplicar el método de Buneman para resolver el sistema de ecuaciones (6.1). En este caso,

$$(6.2) \quad A^{(r)} = - \prod_{j=1}^{2^r} G_j^{(r)}, \quad A^{(0)} = A,$$

$$G_j^{(r)} = \begin{bmatrix} -4+c_j^{(r)} & & 1 & & \bigcirc \\ & 1 & & -4+c_j^{(r)} & \\ & & \ddots & & 1 \\ \bigcirc & & & \ddots & -4+c_j^{(r)} \end{bmatrix},$$

$$c_j^{(r)} = 2 \cos(2j-1) \frac{\pi}{2^{r+1}}.$$

Vemos que para cada r y cada j , tanto en la fase de reducción como de solución, el uso de la factorización (6.2) para $A^{(r)}$ implica la resolución de 2^r sistemas tridiagonales cuya matriz de coeficientes es simétrica y estrictamente diagonal dominante pues

$$2 < 4 - c_j^{(r)} < 6.$$

7. PROGRAMA COMPUTACIONAL.

El algoritmo de Buneman para resolver el problema

$$(7.1) \quad \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = f(x,y), \quad (x,y) \in \text{int } R$$

$$(7.2) \quad u(x,y) = g(x,y), \quad (x,y) \in \partial R$$

donde $R = [a,b] \times [a,b]$, ha sido implementado computacionalmente mediante una subrutina denominada BUNEMA. Esta subrutina tiene por parámetros los valores de a , b , el exponente k de 2^{k+1} .

BUNEMA llama a las siguientes subrutinas: SEG, IMPRIM, RESOLV, TRIDIA.

La subrutina SEG genera el segundo miembro del sistema de ecuaciones (7.1). Los parámetros de esta subrutina son: la matriz B , en la cual se almacenan por columnas las componentes de dicho segundo miembro, el valor $N = 2^{k+1} - 1$, el parámetro H que corresponde a la distancia entre los puntos de la partición del intervalo $[a,b]$, los valores $WA = WC = a$ y $WB = WD = b$. Previamente deben definirse las funciones $F(x,y)$ y $G(x,y)$ correspondientes a $f(x,y)$ y $g(x,y)$ respectivamente.

iii) Para $i = n-2, n-3, \dots, 1$; computar

$$x_i = q_i x_{i+1} + u_i$$

La subrutina RESOLV, después de llamar sucesivamente a la subrutina TRIDIA, nos da la solución de los sistemas de ecuaciones que aparecen en la fase de reducción y en la fase de solución, usando para ello la factorización ya explicada para $A^{(r)}$. Los parámetros de esta subrutina son: el vector B que corresponde al segundo miembro de tales sistemas de ecuaciones, el valor R que representa al índice r, y el valor N correspondiente al tamaño del sistema. Se hace uso de la simetría de los valores

$$\lambda_j^{(r)} = -4 + 2 \cos \frac{(2j-1) \pi}{2^{r+1}}, \quad j = 1, 2, \dots, 2^r$$

respecto de -4.

El programa computacional con sus diferentes subrutinas es presentado en el apéndice.

8. EJEMPLOS.

Deseamos presentar ahora resultados numéricos a través de dos ejemplos referidos ellos al cuadrado $[0,1] \times [0,1]$ y ejecutados usando precisión doble.

$$8.1. \quad f(x,y) = 0$$

$$g(x,y) = x+y$$

La solución exacta es claramente $u(x,y) = x+y$. Para este ejemplo el error al discretizar la ecuación (7.1) es cero. Por lo tanto, salvo errores de redondeo, la solución obtenida al emplear cualquier método directo debe ser igual a la solución exacta.

Trabajando con $n = m = 8$ obtuvimos los siguientes resultados:

0.249999946	0.374999883	0.499999881	0.624999858	0.749999855	0.874999821	0.999999919
0.374999900	0.499999789	0.624999784	0.749999751	0.874999739	0.999999691	1.124999856
0.499999867	0.624999719	0.749999713	0.874999677	0.999999659	1.124999605	1.249999813
0.624999847	0.749999676	0.874999674	0.999999633	1.124999617	1.249999557	1.374999790
0.749999845	0.874999667	0.999999671	1.124999626	1.249999617	1.374999552	1.499999791
0.874999867	0.999999705	1.124999719	1.249999669	1.374999674	1.499999607	1.624999823
0.999999920	1.124999811	1.249999829	1.374999787	1.499999803	1.624999749	1.749999893

El valor presentado en la i -ésima fila y j -ésima columna es, en general, la aproximación para $u(i/n, j/n)$. Observamos que efectivamente, excepto errores debido a redondeo, los valores obtenidos corresponden a los valores exactos de $u(x,y)$.

$$8.2. \quad f(x,y) = -\sin x - \sin y$$

$$g(x,y) = \sin x + \sin y$$

Podemos verificar que $u(x,y) = \sin x + \sin y$ es la solución exacta para el problema (7.1)-(7.2). Trabajando con $n = m = 8$ obtuvimos las siguientes aproximaciones para $u(x,y)$:

0.249362135	0.372101485	0.490977579	0.604135095	0.709807711	0.806345274	0.892239448
0.372101502	0.494848620	0.613730154	0.726890539	0.832563015	0.929096694	1.014982388
0.490977565	0.613730092	0.732615183	0.845777237	0.951449284	1.047980043	1.133859913
0.604135085	0.726890469	0.845777235	0.958939960	1.064611688	1.161140944	1.247017830
0.709807702	0.832562946	0.951449294	1.064611695	1.170283394	1.266812897	1.352690050
0.806345316	0.929096705	1.047980147	1.161141046	1.266813008	1.363344780	1.449226202
0.892239448	1.014982348	1.133859928	1.247017828	1.352690061	1.449226137	1.535116762

En este caso el error al discretizar la ecuación (7.1) no es cero y naturalmente no de bemos esperar un resultado igual al valor exacto.

Para el mismo ejemplo, trabajando finalmente con $n = m = 16$ obtuvimos las aproximaciones siguientes:

0.124919462	0.187135621	0.248864877	0.309866139	0.369901215	0.428735643	0.486139745	0.541889270
0.595766548	0.647561128	0.697070845	0.744102221	0.788471611	0.830005736	0.868542147	
0.187135629	0.249352459	0.311082331	0.372084128	0.432119651	0.490954436	0.548358841	0.604108564
0.657985926	0.709780459	0.759290037	0.806321101	0.850689978	0.892223252	0.930758805	
0.248864868	0.311082298	0.372812713	0.433814977	0.493850884	0.552685973	0.610090622	0.665840500
0.719717909	0.771512378	0.821021799	0.868052564	0.912420966	0.953953588	0.992488337	
0.309866131	0.372084085	0.433814972	0.494817635	0.554853866	0.613689206	0.671094052	0.726844047
0.780721480	0.832515878	0.882025144	0.929055638	0.973423625	1.014955695	1.053489781	
0.369901207	0.432119610	0.493850894	0.554853889	0.614890385	0.673725927	0.731130927	0.786881007
0.840758449	0.892552778	0.942061905	0.989092166	1.033459807	1.074991425	1.113524970	
0.428735658	0.490954430	0.552686035	0.613689293	0.673725995	0.732561689	0.789966804	0.845716944
0.899594387	0.951388654	1.000897668	1.047927741	1.092295109	1.133826370	1.172359491	
0.486139733	0.548358787	0.610090633	0.671094082	0.731130929	0.789966726	0.847371919	0.903122096
0.956999535	1.008793753	1.058302686	1.105332624	1.149699795	1.191230795	1.229763603	
0.541889263	0.604108505	0.665840506	0.726844068	0.786880995	0.845716844	0.903122078	0.958872272
1.012749705	1.064543892	1.114052779	1.161082639	1.205449695	1.246980535	1.285513144	
0.595766536	0.657985863	0.719717922	0.780721514	0.840758453	0.899594299	0.956999535	1.012749725
1.066627151	1.118421326	1.167930210	1.214960055	1.259327083	1.300857871	1.339390406	
0.647561149	0.709780443	0.771512454	0.832515982	0.892552858	0.951388643	1.008793840	1.064544002
1.118421423	1.170215608	1.219724530	1.266754430	1.311121528	1.352652383	1.391184982	
0.697070826	0.759289953	0.821021794	0.882025155	0.942061884	1.000897545	1.058302657	1.114052767
1.167930180	1.219724397	1.269233405	1.316263431	1.360630707	1.402161768	1.440694590	
0.744102207	0.806321011	0.868052544	0.929055619	0.989092115	1.047927583	1.105332558	1.161082583
1.214959985	1.266754254	1.316263393	1.363293622	1.407661197	1.449192621	1.487725856	
0.788471596	0.850689895	0.912420962	0.973423624	1.033459790	1.092294992	1.149699772	1.205449674
1.259327058	1.311121398	1.360630705	1.407661211	1.452029215	1.493561185	1.532095064	
0.830005674	0.892223254	0.953953678	1.014955789	1.074991526	1.133826390	1.191230907	1.246980642
1.300857992	1.352652413	1.402161912	1.449192763	1.493561324	1.535094048	1.573628850	
0.868542146	0.930758764	0.992488350	1.053489784	1.113524986	1.172359448	1.229763621	1.285513146
1.339390425	1.391184934	1.440694616	1.487725865	1.532095084	1.573628780	1.612164831	


```

WRITE (IS,500)
500 FORMAT('1',/40X,"SEGUNDO MIEMBRO",/40X,15('-'),//)
CALL IMPRIM(Q,IS,NSMALL-1,2,NSMALL)

```

C
C
C
C

ASIGNACION DE CEROS A P Y A PARTES DE Q

```

DO 10 J=1,NSMALL-1
  Q(J,1)=0.0
  Q(J,NSMALL+1)=0.0
  DO 10 I=1,NSMALL-1
    P(I,J)=0.0

```

C
C
C

* FASE DE REDUCCION *

```

DO 30 R=1,K
  J1=2**R
  J2=NSMALL-J1
  DO 30 J=J1,J2,J1
    J3=J-J1/2
    J4=J+J1/2
    DO 26 I=1,NSMALL-1
      RH(I)=P(I,J3)+P(I,J4)-Q(I,J+1)
      CALL RESOLV(RH,R-1,NSMALL)

```

C
C
C
C

RESOLV ES UNA RUTINA ENCARGADA DE RESOLVER
LOS 2**R SISTEMAS TRIDIAGONALES

```

DO 27 I=1,NSMALL-1
  P(I,J)=P(I,J)-RH(I)
  Q(I,J+1)=Q(I,J3+1)+Q(I,J4+1)-2*P(I,J)
27 CONTINUE
30

```

C
C
C

* FASE DE SOLUCION *

```

R=K
40 IF(R.LT.0) GO TO 60
  J1=2**R
  J2=NSMALL-J1
  J3=2*J1
  DO 55 J=J1,J2,J3
    J4=J+1+J1
    J5=J+1-J1
    DO 45 I=1,NSMALL-1
      RH(I)=Q(I,J+1)-X(I,J4)-X(I,J5)
      CALL RESOLV(RH,R,NSMALL)
    DO 48 I=1,NSMALL-1
      X(I,J+1)=P(I,J)+RH(I)
48 CONTINUE
55

```

```

R=R-1
GO TO 40
60 CONTINUE
WRITE (IS,501)
501 FORMAT('0',40X,'SOLUCION APROXIMADA',/41X,19('-',)/)
CALL IMPRIM(X,IS,NSMALL-1,2,NSMALL)
RETURN
END
SUBROUTINE RESOLV(B,R,N)
DOUBLE PRECISION B(63),XLDA,PI,XAUX
INTEGER N,R
PI=3.14159265358979
IF(R.NE.0)GO TO 10
XLDA=-4
C
C R=0 ----> A(0) = A
CALL TRIDIA(XLDA,B,N)
RETURN
10 XAUX=PI/2**(R+1)
DO 20 J=1,2**(R-1)
    XANG=(2*DCOS((2*J-1)*XAUX))
    XLDA=XANG-4.0
    CALL TRIDIA(XLDA,B,N)
    XLDA=-XANG-4.0
20 CALL TRIDIA(XLDA,B,N)
DQ 30 I=1,N-1
30 B(I)=(-1)*B(I)
RETURN
END
SUBROUTINE TRIDIA(XLDA,B,N)
DOUBLE PRECISION XLDA,B(63),P(63),Q(63),U(63)
P(1)=XLDA
Q(1)=-1.0/P(1)
U(1)=B(1)/P(1)
DO 10 I=2,N-1
    P(I)=XLDA+Q(I-1)
    Q(I)=-1.0/P(I)
10 U(I)=(B(I)-U(I-1))/P(I)

B(N-1)=U(N-1)
I=N-2
20 IF(I.LT.1) GO TO 30
    B(I)=Q(I)*B(I+1)+U(I)
    I=I-1
    GO TO 20
30 RETURN
END
SUBROUTINE SEG(B,N,H,WA,WB,WC,WD)
DOUBLE PRECISION B(63,65),H,F,G,XA,XB,YC,YD
INTEGER N,WA,WB,WC,WD

```

```

XA=WA
XB=WB
YC=WC
YD=WD
DO 10 J=2,N
    DO 10 I=1,N-1
10      B(I,J)=F(XA+I*H,YC+((J-1)*H))*(H**2)
    DO 20 I=1,N-1
        B(I,2)=B(I,2)-G(XA+I*H,YC)
20      B(I,N)=B(I,N)-G(XA+I*H,YD)
    DO 30 J=2,N
        B(1,J)=B(1,J)-G(XA,YC+((J-1)*H))
30      B(N-1,J)=B(N-1,J)-G(XB,YC+((J-1)*H))
RETURN
END
SUBROUTINE IMPRIM(A,IS,N,IN,IF)
INTEGER IS,N,IN,IF
DOUBLE PRECISION A(63,65)
DO 10 I=1,N
10    WRITE(IS,100)(A(I,J),J=IN,IF)
100  FORMAT(' ',10F13.9)
    WRITE(IS,200)
200  FORMAT('0',30X,70('#'))
RETURN
END
DOUBLE PRECISION FUNCTION F(X,Y)
DOUBLE PRECISION X,Y
F=-DSIN(X)-DSIN(Y)
RETURN
END
DOUBLE PRECISION FUNCTION G(X,Y)
DOUBLE PRECISION X,Y
G=DSIN(X)+DSIN(Y)
RETURN
END

```

BIBLIOGRAFIA.

- [1] B.L. BUZBEE G.H. GOLUB, C.W. NIELSON, "On direct methods for solving Poisson's equations" Siam J. Numer. Anal. Vol. 7, N° 4, December 1970.

- [2] O. BUNEMAN "A compact non-iterative Poisson Solver"
Rep. 294 Standord University Institute for plasma
research. 1969.

- [3] THEODOR MEIS, ULRICH MARCOWITZ. "Numerical Solution of Partial
Differential Equations". Applied Mathematical Sciences 32. Springer - Verlag. 1981.

- [4] ROLAND A. SWEET. "A cyclic reduction algorithm for solving block
triadiagonal systems of arbitrary dimension".
Siam J. Numer. Anal. Vol. 14, N°4, September 1977.