

TECNICAS OPERATIVAS PARA MATRICES RALAS

Miguel Benavente

Guillermo A. Lombera

Juan C. Belmonte

INTEMA (Instituto de Tecnología de Materiales)

Juan B. Justo 4302 - Mar del Plata - Argentina

Haroldo Hack

Facultad de Ciencias Exactas y Naturales - UNMDP

Funes y Peña - Mar del Plata - Argentina

Enrique Pardo

Dept. of Mech. Eng. University of Waterloo

Waterloo - Ontario - Canada

RESUMEN

En el presente trabajo se describen técnicas operativas para matrices almacenadas en formato ralo que permiten confeccionar algoritmos computacionales muy eficientes en lo que hace a memoria y tiempo de cálculo invertidos, por lo que pueden ser exitosamente aplicados a diversos problemas de la mecánica.

Se presenta en primer lugar una rutina para realizar el ensamblaje simbólico de una matriz de rigidez para nodos con más de un grado de libertad. A continuación se detallan programas en lenguaje FORTRAN que resuelven sistemas de ecuaciones cuya matriz de coeficientes posee simetría estructural. Esto incluye la descomposición de dicha matriz y una optimización del proceso de triangulación para matrices banda.

ABSTRACT

Algorithms operating on sparse matrices are described in this work. These algorithms are very efficient, regarding both computer memory and time, and are convenient for different problems in mechanics.

A routine to make the symbolic assenbleage of stiffness matrix for more than one degree of freedom is presented.

FORTRAN programs to solve systems of equations which matrices of coefficients have structural symmetry, are detailed. These include descomposition of such matrices, and optimization of the triangulation process for band matrices.

INTRODUCCION

Este trabajo es una extensión para matrices con estructura simétrica [2] del que realizara S.Pissanetzki [1]. En todos los casos se supone un conocimientos de este último, y no se ha profundizado en detalles ampliamente tratados por este. Así, el primer algoritmo (R1. rutina de ensamblaje simbólico para nodos con más de un grado de libertad) es en esencia similar a la de [1], por lo que si bien se ha considerado conveniente incluirla, se encuentra precedida sólo de una breve descripción.

Un algoritmo muy simple como el producto de una matriz simétrica por un vector lleno en formato ralo es presentado pues fundamenta una forma de representación para matrices con simetria estructural, para la cual es inmediata la extensión del producto citado. Estas operaciones son necesarias para resolver un sistema una vez triangulada su matriz de coeficientes sea que se descomponga o no, y tampoco se volvera sobre mayores detalles.

ABREVIATURAS

MS: Matriz Simétrica.
MES: Matriz con Estructura Simétrica.
NEL: Cantidad de elementos.
N: Cantidad de nodos.
TI Triángulo inferior.
TS: Triángulo superior.
A: Matriz de rigidez
U: Matriz A triangulada

REPRESENTACION DE MATRICES EN FORMATO RALO

En principio se muestra con un ejemplo la representación de una MS:

	5	0	2	0	0	IA=(1,2,4,5,6,6)
	0	6	3	1	0	JA=(3,3,4,4,5)
A=	2	3	4	1	0	AN=(2,3,4,5,6)
	0	1	1	3	3	AD=(5,6,4,3,3)
	0	0	0	3	5	

A continuación se transcribe directamente el algoritmo que efectúa el producto de una MS por un vector lleno.

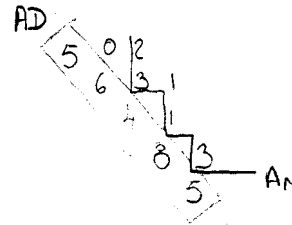
A.1

DATOS: IA,JA,AN,AD
Vector B

N dimensión de la MS A y del vector B
 RESULTADO: Vector C

```

1      DO 10 I=1,n
2          IAA=IA(I)
3          IAB=IA(I+1)-1
4          U=AD(I)*B(I)
5      DO 20 K=IAA,IAB
6          J=JAGO
7          U=U+AN(K)*B(J)
8      20  C(J)=C(J)+AN(K)*B(I)
9      10  C(I)=C(I)+U
    
```



Este algoritmo se encuentra simplificado para rescatar su esencia. Seguidamente se representa una MES en formato ralo en una forma muy conveniente a los efectos del presente trabajo.

	2 0 2 1	IA=(1,3,5,5,5)
A =	0 7 9 0	JA=(3,4,4,3)
	3 8 3 0	AN=(2,1,6,9)
	7 5 0 1	ANT=(3,7,5,9)
		AD=(2,7,3,1)

Debería notarse que el triángulo inferior ha sido almacenado por columna de manera que se corresponda con los punteros IA y JA, hecho en el que radica la potencia de esta representación. Asimismo ésta no necesita estar ordenada siendo esto conveniente cuando se trata de una matriz de conectividad de una red de elementos finitos.

Es interesante ver ahora como se altera A.1 para producir el producto de una MES por un vector lleno. Solo hay que ingresar el vector ANT con los datos del triángulo inferior y modificar la línea 6 de la siguiente manera:

```

6      C(J)=C(J)+ANT(K)*B(I)
    
```

ENSAMBLAJE SIMBOLICO DE UNA MATRIZ DE RIGIDEZ

Sea E la matriz de conectividad de una red de elementos finitos de dimensión $NEL \times N$, la que da por filas los nodos pertenecientes a cada elemento. Por consiguiente su traspuesta E^T da los nodos asociados con cada elemento. Se pretende explicar como funciona un algoritmo de ensamblaje.

Sea I un índice que recorre los nodos de la red. En un principio chequea en E^I los elementos asociados con el nodo I, luego para cada uno de aquellos se determina a través de E cuales son los nodos que le corresponden, los que son en definitiva los que estarán relacionados con el nodo corriente definiendo las posiciones no nulas de la matriz de rigidez, o lo que es lo mismo, llenando su representación en formato raio.

Si por ejemplo llamamos U al vector de las incógnitas y U_x y U_y a sus dos grados de libertad el sistema resultante tendrá dimensión doble y es necesario ingresar esta información en la matriz de rigidez. Así las variables correspondientes al nodo I se ubican de la siguiente manera:

U_x ----- $U(2*I-1)$
 U_y ----- $U(2*N)$

Una vez ensamblada la fila $2*I-1$ ya están a mano los datos necesarios para hacer lo propio con la fila $2*I$. Así si las posiciones no nulas de la fila $2*I-1$ son:

las de la fila $2*I$ serán:

$J_1 \dots J_k$
 $J_1 \dots J_k$

...donde debería notarse que el elemento $A(2*I-1,2*N)$ resulta siempre distinto de cero, y además que si en los arreglos anteriores se encuentra el índice $2*N-1$, también debe estar el índice $2*N$.

Con la rutina de ensamblaje para nodos con un grado de libertad [1], y las aclaraciones precedentes no debería haber dificultad en interpretar la correspondiente a dos grados de libertad (R1), que aparece en el apéndice.

TRIANGULACION DE UNA MES

Notaciones.

-Con (f_i, f_j) se indicará la operación de restar a la fila I la fila J multiplicada por el coeficiente $A(i, j)/A(j, j)$. Al elemento $a(i, j)$ se le dirá multiplicador y a $A(j, j)$ pivot.

-La cadena asociada a la fila I es una lista de números que indica las filas que deben ser combinadas con ella en el proceso de triangulación. Se notará $CF(I)$, y en las rutinas estará almacenada en el vector auxiliar IP.

....-Repetidamente se hace referencia a la primer posición no nula de la fila I de la matriz triangulada U, o a la siguiente en el caso de que una (fk,fi) haya sido efectuada. En todo caso siempre se trata del siguiente pivot de esa fila. Se simbolizará con PNNCI).

En el algoritmo para la triangulación de una MS se opera exclusivamente con el TS pues con cada operación queda implícita la simétrica. Además los multiplicadores correspondientes a una fila cualquiera se encuentran en la columna homóloga. En el correspondiente para una MES las operaciones en el TI deben explicitarse y parece conveniente hacerlo simultáneamente con las simétricas.

En principio el TI de A está almacenado por columna en el vector ANT y en correspondencia simétrica con AN. Debe definirse también un vector UNT de la misma dimensión que UN para almacenar los multiplicadores del TI.

A continuación se aclaran algunos detalles sobre el programa adjuntado a este trabajo R2.

El ciclo do 170 recorre tanto las filas de la matriz A como las de U. En un principio se cargan los vectores auxiliares DI y DIT con los elementos no nulos de AN y ANT.

Posteriormente se carga en LAST (16) el primer elemento de CFCI) si no está vacía. En (19) se define L, la primera fila que será combinada con la fila I. En (21) y (23) se calcula el coeficiente de (fi,fi) a la vez que se normaliza $UNT(IUP(L))$ a los efectos de descomponer la matriz. En (24) se efectúa separadamente la operación sobre el elemento diagonal. En (25) (29) y (30) se completan las operaciones con el resto de la fila L, y en (39) la operación se registra sobre el vector de términos independientes.

Entre la (19) y la (40) se recorre CFCI). Entre la (31) y la (38) L se registra en el vector IP. En la (44) y (45) son transferidos a UN y UNT los valores obtenidos en DI y DIT. Entre la (46) y la (53) I es registrada en el vector IP, y por último en la (54) IUPCI) señala a PNNCI).

Conclusiones

Este programa es muy eficiente en tiempo de cálculo y en aprovechamiento de memoria si el objetivo es descomponer la matriz (notar que en este caso habría que eliminar la 39). Pero si el problema es resolver un sistema donde matriz y vector de términos independientes varían simultáneamente lo ideal es ir insertando las modificaciones en este y luego hacer sustitución hacia atrás. En este caso el vector UNT provoca un considerable desperdicio de memoria ya que este suele ser grande comparado con el vector ANT. Seguidamente se presenta una solución a este problema para cierta clase de matrices.

RALEZA Y ANCHO DE BANDA

Si la matriz del sistema de ecuaciones además de ser rala presenta banda, es posible mejorar la performance de los algoritmos. El proceso de triangulación se realiza en dos etapas, una simbólica y otra numérica entre las que debe intercalarse la doble trasposición como rutina ordenadora [1]. En el siguiente ítem se presenta como alternativa una rutina que determina una representación simbólica ordenada de la matriz triangulada R3, haciendo uso del concepto de ancho de banda (IBAN).

Triangulación simbólica de una MS de banda. R3.

La diferencia principal con la rutina simbólica presentada en [1] es que en esta a medida que se producían las posiciones no nulas se cargaban en JU y se indicaban en IU para impedir que algunas de estas fueran contabilizadas reiteradamente, de manera que la cola de IU actuaba realmente como una llave; en la nueva rutina las posiciones que se van generando son tildadas en IU (12) y (22), hasta el final del procesamiento de cada fila (23) en que son transferidas en forma ordenada al vector JU. Esto se puede hacer sin mayor gasto puesto que en general el arreglo IU se carga en forma densa en las primeras IBAN-1 posiciones, con respecto a la diagonal. Por el contrario, si la matriz no es de banda este método supone una búsqueda onerosa en la cola de la llave IU. Otra ventaja nada despreciable es que al transferirlos ordenadamente el contador JP (27) almacenado por filas en JPI (5) da directamente PNN(I) (31) sin necesidad de buscar el mínimo (JUKJPI)), en cuya cadena será añadida I solo si tiene otros elementos no nulos (29) y (30).

Triangulación numérica de una MES de banda

Sea el caso triangular una MES en la cual $AC(1,N)$ es no nulo. Cuando se genere $UK(1,N)$ y se cargue en UN sucederá lo propio con $UK(N,1)$ y UNT . De modo que una posición cargada en la primer iteración recién será liberada en la última. Este hecho desafortunado sugiere la siguiente observación: la dispersión de los elementos útiles en el vector UNT será tanto menor cuanto menor sea $IBAN$.

Supongase entonces triangular una MES de banda. El siguiente esquema presenta una porción de esta matriz donde por simplicidad los índices representan a los elementos.

11	12	13	14					
21	22	23	24	25				
31	32	33	34	35	36			
41	42	43	44	45	46	47		
	52	53	54	55	56	57	58	
		63	64	65				
			74	75				
				85				

Supóngase que el proceso se encuentra en la fila 5. Cuando se operaron las filas 2, 3 y 4, se generaron por simetría las columnas homólogas. Porciones de estas han sido destacadas en la figura, conformando un triángulo que encierra precisamente los elementos que se invocarán para generar la fila 5 y sus simétricos. En principio es claro que 52, 53 y 54 son los multiplicadores de la iteración corriente. Además, cuando se modifique 56 se utilizarán 36 y 46, y por simetría se generará 65 invocando a 63 y 64. Y así siguiendo hasta cubrir el triángulo.

Debería notarse que lo que está sobre el triángulo ya puede ser deshechado, mientras lo que está por debajo aún no fue cargado. Además en este paso será liberada la primer fila, generándose a cambio una columna que en principio fue colocada debajo del elemento diagonal 55.

En la rutina R4 que se presenta la técnica que se utiliza es la siguiente: la columna generada se almacena sobre la fila que se descarta, requiriéndose entonces un vector que sólo tiene la dimensión del triángulo. Esto origina una mayor complejidad en la búsqueda de los multiplicadores, como se manifiesta en las fórmulas (29), (30), (32) y demás de R4, cuyos detalles fueron considerados poco esenciales y demasiado técnicos para ser incluidos en este trabajo.

Descripción de R4

En la (26) se define L_1 , variable auxiliar para localizar en VV a los simétricos de la fila L . Por eso los elementos del TI se obtienen via VL_1 como en (30) donde se calcula el coeficiente de (f_1, f_1) o en (45) donde se invocan los elementos del triángulo para efectuar las operaciones simétricas. Ambas previo pase por las llaves (29).

En (27) $L_2 = I - N$ es la distancia entre las filas, pero también es la de UCL, I a la diagonal, por lo tanto es la componente en VL_1 del multiplicador (30) y (32). Además L_2 es el número que se debe restar a los índices de L para homologarlos con los de la fila I (40) y (41), de manera que las operaciones se hacen en la posición J_3 de DI y DIT .

Entre (45) y (47) se determinan las posiciones de los elementos simétricos que serán requeridos del triángulo según lo expuesto.

Debería notarse que las operaciones en el TS tienen como elemento fijo al multiplicador VVC , el mismo que en el algoritmo anterior era $UNT(IUC)$. En el TI el elemento fijo es el simétrico de este que siempre es $UNC(IUC)$. El resto de esta rutina coincide en esencia con la presentada en [1]

CONSIDERACIONES FINALES

El presente trabajo proporciona un método eficiente para la resolución de sistemas de ecuaciones con matrices ralas de estructura simétrica de grandes dimensiones.

Tales problemas aparecen usualmente asociados a la resolución numérica de operadores diferenciales no autoadjuntos [4] [5] y [6]. Las matrices de estos sistemas de ecuaciones son generalmente irreducibles y débilmente diagonal dominantes, por lo que se hace innecesario cualquier forma de pivoteo [3].

APLICACIONES

Aplicación al problema de flujo térmico bidimensional en uniones soldadas [4], [5], [6] y [7].

Una importante aplicación de estas técnicas, que dieron origen a su estudio es en el problema de flujo térmico en soldadura. En dicho problema la ecuación de conducción de calor estacionaria es:

$$\frac{\partial}{\partial x} \left[K \frac{\partial T}{\partial x} \right] + \frac{\partial}{\partial \xi} \left[K \frac{\partial T}{\partial \xi} \right] + c \rho v \frac{\partial T}{\partial \xi} + \frac{2h}{s} (T - T_0) + q = 0$$

siendo:

- K = conductividad térmica [W/m °C]
- h = coeficiente de transferencia pelicular [W/m² °C]
- s = espesor de la chapa [m]
- q = fuente calórica [W/m³]
- c = calor específico [Joule/kg °C]
- ρ = densidad [kg/m³]
- t = tiempo [s]
- T = temperatura [°C]
- ξ, y = coordenadas espaciales (ξ = y - vt)
- v = velocidad de soldadura [m/s]

Esta ecuación diferencial es resuelta por el método de elementos finitos, con una formulación del tipo Galerkin y una aproximación lineal. La discretización fue hecha con elementos triangulares trinodales. La matriz de rigidez posee estructura simétrica, para lo cual se utilizaron los algoritmos explicados anteriormente con muy buen resultado. Los tiempos de cómputo comparados con la utilización del formato clásico, bajaron en un 60%, y la memoria utilizada en un 30%.

Modelo de elementos finitos [7] aplicado a la hidrología subterránea: el acuífero de Camet de la ciudad de Mar del Plata.

La ecuación que describe el flujo de aguas subterráneas es:

$$\frac{\partial}{\partial x} (T \frac{\partial h}{\partial x}) + \frac{\partial}{\partial y} (T \frac{\partial h}{\partial y}) = Q + (S + S_y) \frac{\partial h}{\partial t}$$

h : altura hidráulica

T : transmisividad

Q : describe los caudales extraídos, la infiltración y la evaporación (depende de la posición y el tiempo).

S : coeficiente de almacenamiento.

S_y : coeficiente de rendimiento específico.

La matriz asociada al problema resulta simétrica. Se confeccionó una red de 864 nodos (822 incógnitas), y 414 elementos triangulares hexanodales (aproximación cuadrática). Con los siguientes resultados:

$$IBAN = 150 \quad \dim \langle AN \rangle = 4128 \quad \dim \langle UN \rangle = 32.106$$

Es notable el hecho de que la matriz de rigidez tiene menos de un 2% de elementos significativos, y triangulada menos del 40%. Además para fijar las condiciones iniciales se resolvió el problema estacionario para lo cual debió reenumerarse una parte de la red sacrificando la condición de banda de la matriz, resultando la matriz triangulada del mismo orden. Todo esto hizo posible la corrida del programa en una HP 1000 de sólo 900 K.

Distribución de tensiones térmicas en una chapa delgada (8).

Fue confeccionado para hallar la distribución de tensiones alrededor de una unión soldada. Opera absolutamente con las técnicas expuestas en este trabajo y se logró con el mismo una gran eficiencia.

APENDICE

```
R1
DATOS:IE,JE,IET,JET,N
RESULTADOS:IA,JA,IBAN
IA(1)=1
IDIF=1
IBAN=2
JP=1
NM=N-1
DO 51 I=1,NM
I2=2*I
JPI=JP
IDIF=IBAN+I
IF(IP(I).EQ.N)GO TO 56
JA(JP)=I2
JP=JP+1
IETA=IET(I)
IETB=IET(I+1)-1
DO 52 II=IETA,IETB
J=JET(II)
IEA=IE(J)
IEB=IE(J+1)-1
DO 53 KP=IEA,IEB
K=JE(KP)
IF(K.LE.DGO TO 58
IF(GP(K).GE.DGO TO 58
IP(K)=I
IF(K.GT.IDIF)IDIF=K
K=2*K-1
JA(JP)=K
JP=JP+1
58 CONTINUE
```

```
52 CONTINUE
   IDIF=IDIF-1
   IF(IDIF. GT. IBAN)IBAN=IDIF
   IH=JP-JPI-1
   IN2=2*IH
   IF(IH. EQ. 0)GO TO 54
   DO 20 L=1,IH
20  JA(JP+L-1)=JA(JP+L)+1
   JP=JP+IH
   DO 24 L=1,IN2
24  JA(JP+L-1)=JA(JP-L)
   JP=JP+IN2
54  IA(I2)=JPI+IN2+1
   IA(I2+1)=JPI
   GO TO 51
56  IA(I2)=JPI
   IA(I2+1)=JPI
51 CONTINUE
   N2=2*N
   JP1=JP+1
   JA(JP)=N2
   IA(N2)=JP1
   IA(N2+1)=JP1
   IBAN=IBAN*2+2
   END
```

R2.

DATOS:IA,JA,AN,ANT,AD,IU,JU,M
RESULTADOS:UN,UNT,AD(diag de U)

```
DO 170 J=1,M          3
  IUA=IU(I)           4
  IUB=IU(I+1)-1      5
  IF(IUB. LT. IUA)GO TO 40  6
  DO 20 J=IUA,IUB     7
    DI(JU(J))=0      8
20  DIT(JU(J))=0     9
    IAA=IA(I)        10
    IAB=IA(I+1)-1    11
    IF(IAB. LT. IAA)GO TO 40  12
    DO 30 J=IAA,IAB  13
      DI(JA(J))=AN(J)  14
30  DIT(JA(J))=ANT(J)  15
40  LAST=IP(I)       16
    IF(LAST. EQ. 0)GO TO 110  17
    LN=IP(LAST)     18
50  L=LN            19
```

```

LN=IPLJ          20
IUC=IUP(L)      21
IUD=IUP(L+1)-1  22
UM=UNT(IUC)/AD(L) 23
AD(D=AD(I)-UM*UN(IUC) 24
IF(IUC.EQ.IUD)GO TO 80 25
IUP(L)=IUC+1    26
DO GO J=IUP(L),IUD 27
JI=JU(J)        28
DI(J)=DI(JI)-UM*UN(J) 29
GO DIT(JI)=DIT(JI)-UN(IUC)*UNT(J)/AD(L)
J=JU(IUP(L))    31
JJ=IP(J)        32
IF(JJ.EQ.0)GO TO 70 33
IP(L)=IP(JJ)    34
IP(JJ)=L        35
GO TO 80        36
70 IP(J)=L       37
IP(L)=L         38
80 B(I)=B(I)-UM*B(L) 39
100 IF(L.NE.LAST)GO TO 50 40
110 IF(IUB.LT.IUA)GO TO 100 41
DO 120 J=IUA,IUB 42
JJ=JU(J)        43
UN(J)=DI(JJ)    44
120 UNT(J)=DIT(JJ) 45
J=JU(IUA)       46
JJ=IP(J)        47
IF(JJ.EQ.0)GO TO 140 48
IP(D=IP(JJ)     49
IP(JJ)=I        50
GO TO 100       51
140 IP(J)=I     52
IP(D=I          53
160 IUP(D=IUA   54
170 CONTINUE    55
END

```

R3.

DATOS:IA,JA,M,IBAN
RESULTADOS:IU,JU(arreglo ordenado)

```

JP=1            3
DO GO I=1,N-1  4
JPI=JP         5
IUC=JPI        6
IAA=IA(D)      7

```

```
IAB=(I+1)-1 8
IF(IAB.LT.IAA)GO TO 30 9
DO 20 J=IAA,IAB 10
JJ=JA(J) 11
20 IU(JJ)=I 12
30 LAST=IP(I) 13
IF(LAST.EQ.0)GO TO 60 14
L=LAST 15
40 L=IP(L) 16
IUA=IU(L)+1 17
LH=L+1 18
IUA=IU(LH)-1 19
DO 50 J=IUA,IUB 20
JJ=JU(J) 21
50 IU(JJ)=I 22
IF(L.NE.LAST)GO TO 40 23
DO 70 K=1,IBAN-1 24
IF(IU(I+K).NE.I)GO TO 70 25
JU(JP)=I+K 26
JP=JP+1 27
70 CONTINUE 28
J1=JP-JPI 29
IF(J1.LE.1)GO TO 90 30
MIN=JU(JPI) 31
M=IP(MIN) 32
IF(M.EQ.0)GO TO 80 33
IP(I)=IP(M) 34
IP(M)=I 35
GO TO 90 36
80 IP(MIN)=I 37
IP(I)=I 38
90 CONTINUE 39
IU(M)=JP 40
IU(M+1)=JP 41
END 42
```

R4.

DATOS:IA,JA,AN,ANT,AD,IU,JU,M,B,IBAN
RESULTADOS:UN,AD(dieg. de U)

```
DO 180 I=1,M 3
IUA=IU(I) 4
IUB=IU(I+1)-1 5
IF(IUB.LT.IUA)GO TO 40 6
DO 20 J=IUA,IUB 7
J1=JU(J) 8
J2=J1-1 9
```

	DI(J2)=0.	10
20	DIT(J2)=0.	11
	I1=I-(I/IBAN)*IBAN+1	12
	IAA=IACD	13
	IAB=IA(I+1)-1	14
	IF(IAB.LT.IAA)GO TO 40	15
	DO 30 J=IAA,IAB	16
	J1=JA(J)	17
	J2=J1-I	18
	DI(J2)=AN(J)	19
30	DIT(J2)=ANT(J)	20
40	LAST=IP(I)	21
	IF(LAST.EQ.0)GO TO 110	22
	LN=IP(LAST)	23
	L=LN	24
	LN=IP(L)	25
	L1=L-(L/IBAN)*IBAN+1	26
	L2=L-L	27
	L3=L1+L2	28
	IF(L3.GT.IBAN)GO TO 55	29
	UM=VV((L1-1)*(2*IBAN-L1)/2+L2)/AD(L)	
	GO TO 60	31
55	UM=VV((L3-IBAN-1)*(3*IBAN-L3)/2+IBAN-L2)/AD(L)	
60	IUC=IUP(L)	33
	IUD=IUD+1)-1	34
	ADD=ADD-UM*UN(IUC)	35
	IF(IUC.EQ.IUD)GO TO 80	36
	IUP(L)=IUC+1	37
	DO 65 J=IUP(L),IUD	38
	J1=JU(J)	39
	J2=J1-L	40
	J3=J2-L2	41
	DI(J3)=DI(J3)-UM*UN(J)	42
	J4=L1+J2	43
	IF(J4.GT.IBAN)GO TO 61	44
	F=VV((L1-1)*(2*IBAN-L1)/2+J2)	
	GO TO 65	46
61	F=VV((J4-IBAN-1)*(3*IBAN-J4)/2+IBAN-J2)	
65	DIT(J3)=DIT(J3)-UN(IUC)*F/AD(L)	
	J=JU(IUP(L))	49
	JJ=IP(J)	50
	IF(JJ.EQ.0)GO TO 70	51
	IP(L)=IP(JJ)	52
	IP(JJ)=L	53
	GO TO 80	54
70	IP(J)=L	55
	IP(L)=L	56
80	B(I)=B(I)-UM*B(L)	57

```
100 IF(L. NE. LAST)GO TO 50          58
110 IF(IUB. LT. IUA)GO TO 170       59
    DO 140 J=IUA,IUB                 60
    J1=JU(J)                         61
    J2=J1-I                          62
    J3=J2+I1                         63
    UN(J)=DI(J2)                     64
    IF(J3. GT. IBAN)GO TO 130        65
    VV((I1-1)*(2*IBAN-I1)/2+IBAN-J2)=DIT(J2)
    GO TO 140                         67
130 VV((J3-IBAN-1)*(3*IBAN-J3)/2+IBAN-J2)=DIT(J2)
140 CONTINUE                         69
    J=JU(IUA)                        70
    JJ=IP(J)                         71
    IF(JJ. EQ. 0)GO TO 150          72
    IP(I)=IP(JJ)                    73
    IP(JJ)=I                         74
    GO TO 170                        75
150 IP(J)=I                         76
    IP(I)=I                         77
170 IUP(I)=IUA                      78
180 CONTINUE                         79
    END                              80
```

AGRADECIMIENTOS

Los autores agradecen la colaboración prestada por la Div. Soldadura del INTEMA y a su director Ing. Raul Conde. Asimismo a la Lic. Carola Suarez de la Universidad Nacional de Mar del Plata por los datos aportados en el modelo del acuífero.

Desean también hacer llegar este reconocimiento al Director de la Fundación Latinoamericana de Soldadura, Ing. Luis A de Vedia y al Lic. Gustavo Sanchez Sarmiento, por el apoyo que permanentemente brindan a esta división.

Finalmente, cabe mencionar que las tareas se realizan bajo el patrocinio del CONICET, la CIC, la SECYT y la UNMDP, para quienes se hace extensivo este agradecimiento.

REFERENCIAS

- [1] Pissanetzky, S. "Operaciones con Matrices Ralas". Argentina. 1980.
- [2] Benavente M., Pardo E., Hack H., "Técnicas Operativas para Matrices Ralas con Estructura Simétrica". Informe Técnico de INTEMA - 1987.
- [3] Wendroff, B. "Theoretical Numerical Analysis" Academic Press Inc., Nueva York - 1966.
- [4] Pardo, E. "Aplicación de métodos numéricos al problema de flujo térmico en uniones soldadas". Tesis Doctoral. Universidad Nacional de Mar del Plata - 1987
- [5] Pardo, E., Lombera, G., Benavente, M., Hack, H., Sanchez Sarmiento, G. "Análisis térmico en uniones soldadas mediante técnicas de elementos finitos" Jornadas metalúrgicas de la S.A.M. Bs. As. 1987
- [6] Pardo, E., Lombera, G., Benavente, M., Hack, H., Sanchez Sarmiento, G. "Distribución de temperaturas en Uniones Soldadas mediante el empleo de técnicas de elementos finitos" Conferencia Interamericana sobre Tecnología de Materiales, Santiago, Chile 1987.
- [7] Lombera, G., Sedrani, A., De vedía L. "Computational and experimental determination of temperature distribution in arc welding" 2nd. International Conference on Trends in Welding Research - Tennessee - 1989.
- [8] Hack, H. "Modelo en elementos finitos para la resolución de ecuaciones diferenciales en derivadas parciales". Informe interno Facultad de Ciencias Exactas y Naturales- 1988.
- [9] Benavente, M., Belmonte, J., Cassannelli, M., Lombera, G. "Programa TENSITER, para el cálculo de la distribución de tensiones térmicas bidimensionales en soldadura" Informe interno de INTEMA -1988.