

Generación automática de una malla irregular y su aplicación con elemento finito *

Jorge López López [†]
Justino Alavez Ramírez [‡]
Juan C. González Aguirre [§]

Universidad Juárez Autónoma de Tabasco, DACB

Este trabajo considera los aspectos computacionales de la aplicación del método de elemento finito para la solución de un problema elíptico bidimensional con condiciones de frontera mixtas. En particular se trata la generación automática de una malla irregular sobre un dominio no trivial. Lo aquí expuesto tiene aplicación directa en la solución numérica de problemas tipo Stefan en 2D.

The matter of this work are some computational aspects of the application of finite element method to the numerical solution of a 2D elliptic problem with elementary mixed boundary conditions. In particular, an automatic generation of an irregular mesh in an irregular domain like an L inverted is presented. This material could be applied to numerically solve some Stefan type problems in 2D.

Palabras clave: Elípticos, Elemento Finito, Matrices Definidas Positivas, Cholesky, Problemas de Stefan, Mallas no estructuradas.

Keywords: Elliptic, Finite Element, Positive Definite Matrix, Cholesky, Stefan Problems, Non structured meshes.

1. Introducción

El problema elíptico considerado en este trabajo es de la forma:

$$-\nu \Delta u + \alpha u = f, \text{ en } \omega \subset \mathbb{R}^2, \quad (1)$$

$$u = g_0 \text{ sobre } \gamma_0, \quad (2)$$

$$\frac{\partial u}{\partial n} = g_1 \text{ sobre } \gamma_1, \quad (3)$$

donde $\nu > 0$, $\alpha \geq 0$, ω un dominio en forma de L invertida cuya frontera $\gamma = \partial\omega$ es la unión disjunta de $\gamma_0 = \gamma_{01} \cup \gamma_{02}$ y $\gamma_1 = \gamma_{11} \cup \gamma_{12} \cup \gamma_{13} \cup \gamma_{14}$. Ver figura 1. Estos problemas y dominios resultan al aproximar numéricamente la solución de problemas tipo Stefan en un dominio rectangular Ω en el cual el subdominio en forma de L contiene una sustancia cuya concentración se difunde de acuerdo al modelo elíptico planteado y que en la interface γ_0 tienen lugar reacciones que hacen que esta se mueva hacia el punto A . Con esto, se espera que los fenómenos más interesantes se den alrededor de la interface, por lo cual es necesario usar una malla que esté de

*Recibido el 12 de marzo de 2009 y aceptado el 6 de mayo de 2009

[†]**Dirección postal:** Carr. Cunduacán-Jalpa Km 1, Cunduacán Tabasco, México. A.P. 24 C.P. 86690. Tel. (+52)914 336-0928. **Correo electrónico:** jorge.lopez@dacb.ujat.mx

[‡]**Correo electrónico:** justino.alavez@dacb.ujat.mx

[§]**Correo electrónico:** juanc@hotmail.com

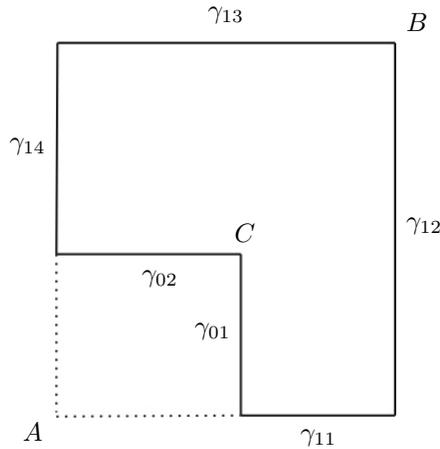


Figura 1. Un dominio ω en forma de L invertida.

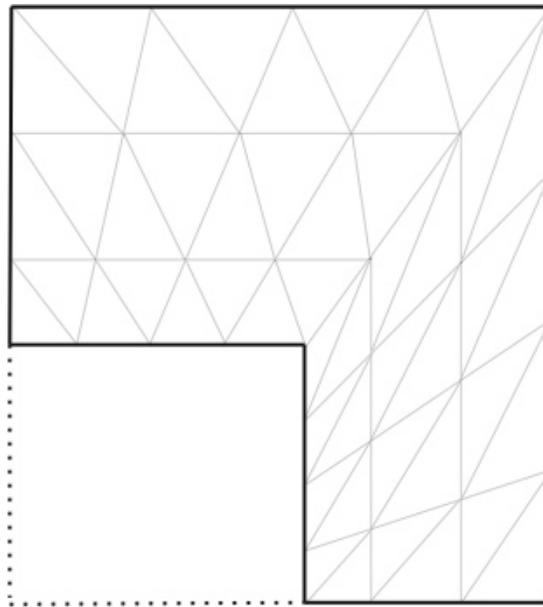


Figura 2. Malla para el dominio ω irregular.

acuerdo con estos requerimientos.

Como la formulación teórica de elemento finito para el tipo de problemas elípticos (1)-(3) se describió con suficiente detalle en [5], aquí nos concentraremos en los aspectos computacionales asociados con la generación automática de la malla propuesta, la cual es como se muestra en la figura 2, y en las consecuencias que esta modificación tiene en los cálculos posteriores implementados en el programa ELIG0G1SKY, que se implementó para resolver por elemento finito el problema (1)-(3) en un dominio rectangular ω y para el cual γ_0 y γ_1 son como en la figura 3.



Figura 3. Un dominio ω rectangular con malla regular.

2. Definición y generación automática de la malla

La malla usada es como se muestra en la figura 2 donde la característica principal es que está formada por rayos por un lado y por el otro por líneas horizontales/verticales mas cercanas unas de otras cuanto mas cercanas están de la interface.

Una vez visualizada la forma de la malla, para su uso computacional, debe generarse automáticamente, es decir, se debe tener un algoritmo para dados los puntos A , B y C , los valores NTB , NTA y una regla de distribución de las líneas verticales y horizontales, se conozcan los nodos, etiquetándolos con un número $1, 2, \dots, N$, y sus respectivas coordenadas, se conozcan los triángulos que forman la malla, etiquetándolos con un número $1, 2, \dots, T$, y estableciendo la relación de los vértices de cada triángulo con los nodos globales, lo cual es importante para llevar a cabo eficientemente los cálculos de las integrales asociadas con el método de elemento finito.

La numeración de nodos que se usa para esta malla se muestra en la figura 4 y en la cual se debe resaltar que primero se numeran los nodos interiores (incluidos los de la frontera γ_1 pues también son nodos incógnita) y se dejan para el final los nodos de la frontera γ_0 que son nodos en donde se conoce la solución.

La numeración de triángulos se ilustra en la figura 5 y lo que hay que notar en ella es que se numeraron primero los triángulos que no tienen vértices en la frontera γ_0 . Esto se hace así solamente porque permite establecer la relación entre nodos locales y nodos globales con un algoritmo similar al usado para establecer esta relación como en el caso del dominio y malla usados en [1], ya que el dominio de L invertida y la malla mencionada se pueden transformar de manera afín a un rectángulo y una malla uniforme como la de la figura 3.

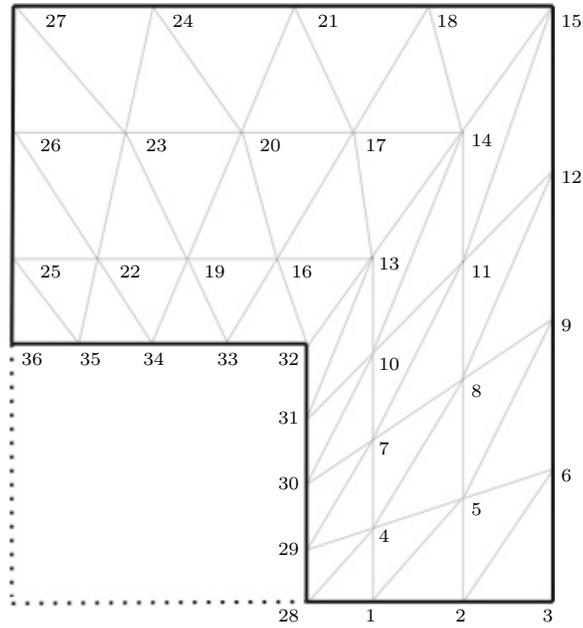


Figura 4. Numeración de nodos para ω y malla irregulares.

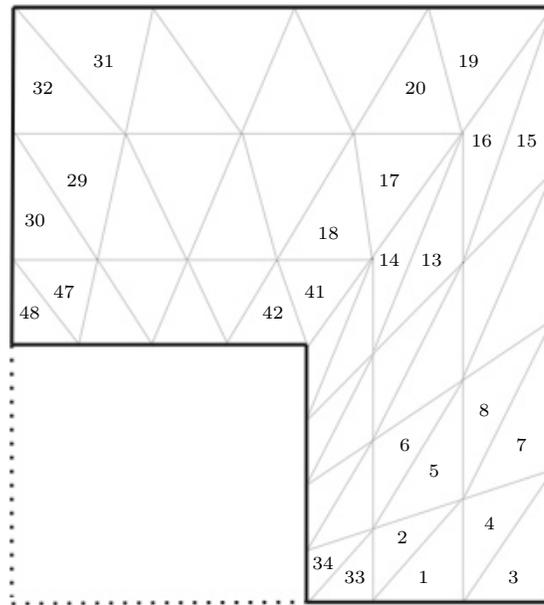


Figura 5. Numeración de triángulos para ω y malla irregulares.

Para establecer las líneas verticales, las cuales deben estar menos espaciadas cuanto más cerca estén de la interface, se mapean los puntos

$$\theta_i = i * \frac{\pi}{2NTB}, \quad i = 0, 1, \dots, NTB,$$

del intervalo $[0, 1]$ a los puntos x_i sobre el intervalo $[xc, xb]$ vía la transformación

$$(xb - xc)\theta^2 + xc.$$

La intersección de las rectas verticales definidas así cortan al segmento CB en puntos cuya abscisa define las rectas horizontales. Los $NTB/2 - 1$ puntos en cada segmento de la frontera que definen los rayos están igualmente espaciados, así que es fácil localizarlos y encontrar la intersección de cada uno de estos con las rectas verticales y con las rectas horizontales. El algoritmo que se usó para hacer todo esto se muestra en el apéndice.

Siendo τ_h la colección de triángulos K que forman la partición de ω al usar la malla mencionada, se define como en [1], para $K \in \tau_h$:

$$P_1(K) = \{p : K \rightarrow \mathbb{R} \mid p(x, y) = a + bx + cy\},$$

$$H_h(\omega) = \{v \in C^0(\omega) : v|_K \in P_1(K), \forall K \in \tau_h\},$$

$$V_h(\omega) = \{v \in H_h(\omega) : u_h = g_0 \text{ sobre } \gamma_0\},$$

$$W_h(\omega) = \{v \in H_h(\omega) : v(n_j) = 0, \forall n_j \in \gamma_0\}.$$

3. El sistema lineal a resolver

Como se vió en [1], si se define

$$V = \{v \in H^1(\omega) : v = g_0 \text{ sobre } \gamma_0\}$$

y

$$W = \{v \in H^1(\omega) : v = 0 \text{ sobre } \gamma_0\},$$

entonces el problema variacional asociado con nuestro problema elíptico es: Encontrar $u \in V$ tal que

$$\nu \int_{\omega} \nabla u \cdot \nabla v dx dy + \alpha \int_{\omega} u v dx dy = \int_{\omega} f v dx dy + \int_{\gamma_1} g_1 v d\gamma, \quad \forall v \in W, \quad (4)$$

y para usar elemento finito una opción es usar los subespacios de dimensión finita V_h y W_h contenidos en V y W respectivamente y resolver el problema

$$\begin{cases} \text{Encontrar } u_h \in V_h \text{ tal que} \\ a(u_h, v_h) = \langle f, v_h \rangle, \quad \forall v_h \in W_h, \end{cases} \quad (5)$$

donde

$$a(u_h, v_h) = \nu \int_{\omega} \nabla u_h \cdot \nabla v_h + \alpha \int_{\omega} u_h v_h$$

y

$$\langle f, v_h \rangle = \int_{\omega} f v_h + \int_{\gamma_1} g_1 v_h d\gamma.$$

Asumiendo que $\{v_1^h, \dots, v_n^h, v_{n+1}^h, \dots, v_N^h\}$ es la base para H_h y $\{v_1^h, \dots, v_n^h\}$ es la base para W_h , se tiene que si $u = (u_1, \dots, u_N)$ es el vector de aproximación para la solución u en cada nodo:

$$u_j = u_h(n_j), \quad j = 1, \dots, N,$$

para n_j el nodo j de la discretización de ω , $u_j = g_0(n_j)$ para $j = n + 1, \dots, N$,

$$u_h = \sum_{j=1}^N u_j v_j^h,$$

y que las u_j deben satisfacer

$$\sum_{j=1}^n u_j a(v_i^h, v_j^h) = \langle f, v_i^h \rangle - \sum_{j=n+1}^N u_j a(v_i^h, v_j^h), \quad i = 1, \dots, n, \quad (6)$$

o equivalentemente

$$Au = b, \quad (7)$$

donde $a_{ij} = a(v_i^h, v_j^h)$ y $b_i = \langle f, v_i^h \rangle - \sum_{j=n+1}^N u_j a(v_i^h, v_j^h)$. La matriz A es simétrica y definida positiva, por lo cual el vector solución u existe y es único, así que la aproximación u_h a la solución u existe y es única. Para el cálculo de los elementos de la matriz A se hace uso de la propiedad de aditividad de la integral y de la forma en que se discretizó el dominio ω :

$$a_{ij} = \sum_{K \in \tau_h} \int_K (\nu \nabla v_i^h \cdot \nabla v_j^h + \alpha v_i^h v_j^h). \quad (8)$$

$$= \sum_{K \in \tau_h} Area(K) \left[\nu (\nabla v_i^h \cdot \nabla v_j^h) + \alpha \begin{cases} 1/6 & \text{si } i = j \\ 1/12 & \text{si } i \neq j \end{cases} \right]. \quad (9)$$

Como en la mayoría de los triángulos las funciones v_i^h y v_j^h tendrán valor cero, no es eficiente fijar una i y una j y moverse sobre todos los triángulos con el fin de calcular a_{ij} . Lo que se hace es moverse sobre cada triángulo K , calcular:

$$Area(K) [\nu (\nabla v_l^K \cdot \nabla v_p^K) + \alpha \begin{cases} 1/6 & \text{si } l = p \\ 1/12 & \text{si } l \neq p \end{cases}]$$

para cada $l = 1, 2, 3$ y cada $p = 1, 2, 3$, siendo v_l^K las funciones base locales del triángulo K , y acumular este valor a a_{ij} si el nodo local l corresponde al nodo global i y el nodo local p corresponde al nodo global j . Debe tenerse en cuenta, además, que si i y j son menores que n , la participación es para A , mientras que si $i \leq n$ y $j > n$ la participación es para b_i y finalmente, si $i > n$ no hay participación ni para A ni para

b. Todas estas ideas se usaron ya en el algoritmo para el cálculo de A en [1], así que se usa el mismo.

Dado que

$$b_i = \int_{\omega} f v_i^h + \int_{\gamma_1} g_1 v_i^h d\gamma - \sum_{j=n+1}^N u_j a(v_i^h, v_j^h)$$

y en [1] se mostró cómo se pueden calcular las integrales involucradas al moverse triángulo por triángulo y cómo se deben de ensamblar al respectivo b_i , y los cálculos se hacen solo en términos de $Area(K)$ y de las coordenadas de los nodos, aquí tampoco afecta usar una malla no uniforme, con tal de que se tengan las coordenadas de los nodos y la relación de nodos locales con globales. En resumen, a partir del programa ELIG0G1SKY se modificó solo el algoritmo con el que se establecen las coordenadas de los nodos y la relación entre nodos locales con globales y con esto se tiene el programa ELIELE que resuelve el problema elíptico (1)-(3) en un dominio ω como el de la figura 2, en donde también se ilustra qué parte de la frontera es γ_0 y γ_1 .

4. Ejemplo numérico

Se considera el dominio ω en forma de L invertida definido por los puntos $A = (-1/2, -1/2)$, $B = (1/2, 1/2)$ y $C = (0, 0)$, $\alpha = 1$, $\nu = 1$, $g_0 = 0$, $g_1 = 0$, y $f(x, y) = (\alpha + 2\nu\pi^2)\text{sen}(\pi x)\text{sen}(\pi y)$. Para estos datos la solución exacta es $u(x, y) = \text{sen}(\pi x)\text{sen}(\pi y)$.

Se ejecutó el programa ELIELE con $NTB = 16$ y $NTH = 32$, resultando una malla como la de la figura 6 y obteniéndose la solución aproximada que se muestra en la figura 7. En la figura 8 se muestra la solución exacta.

En las figuras 9 y 10 se muestran las curvas de nivel de la solución exacta y de la solución aproximada, respectivamente.

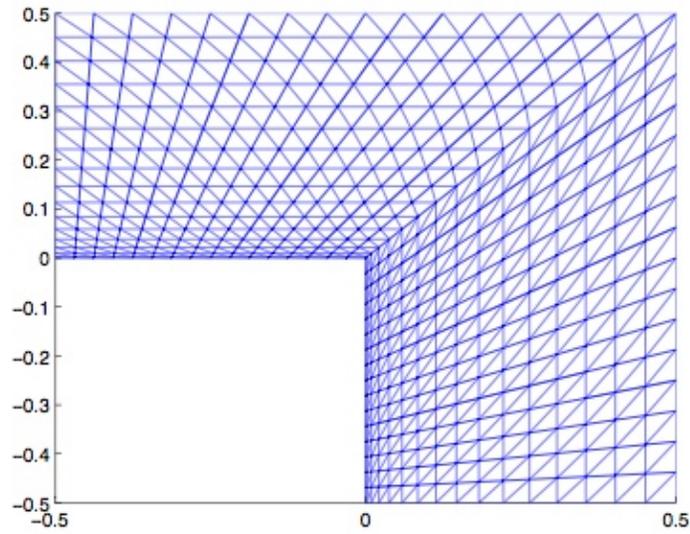


Figura 6. Malla irregular de 16×32 para el ejemplo numérico.

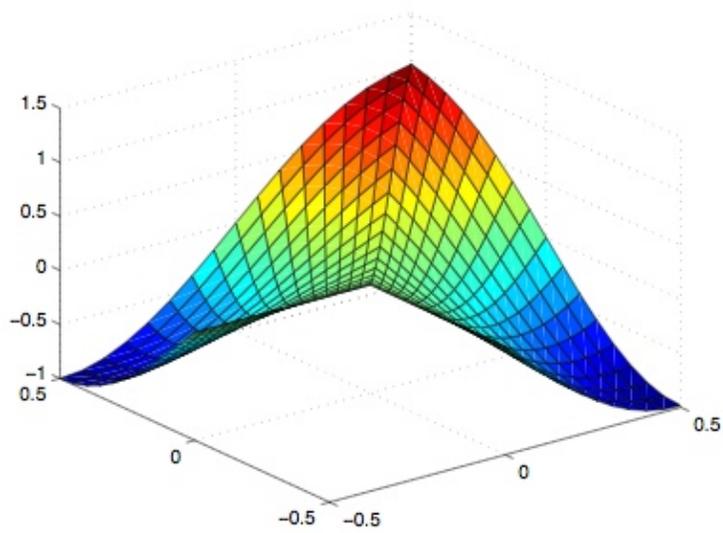


Figura 7. Solución aproximada con una malla de 16×32 .

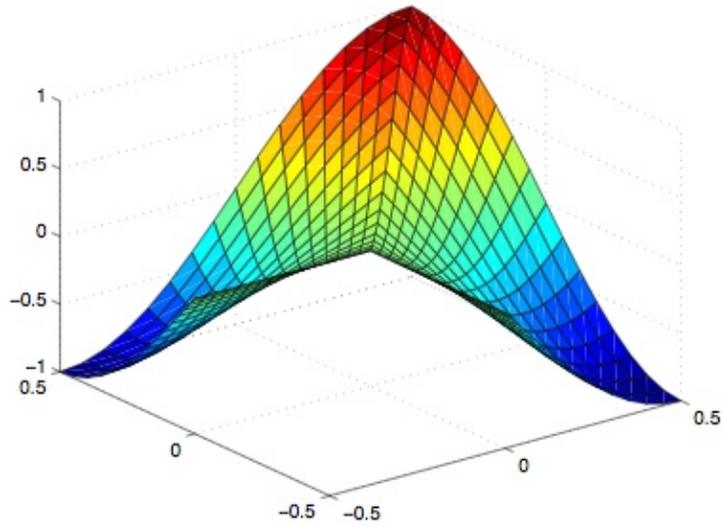


Figura 8. Solución exacta con una malla de 16×32 .

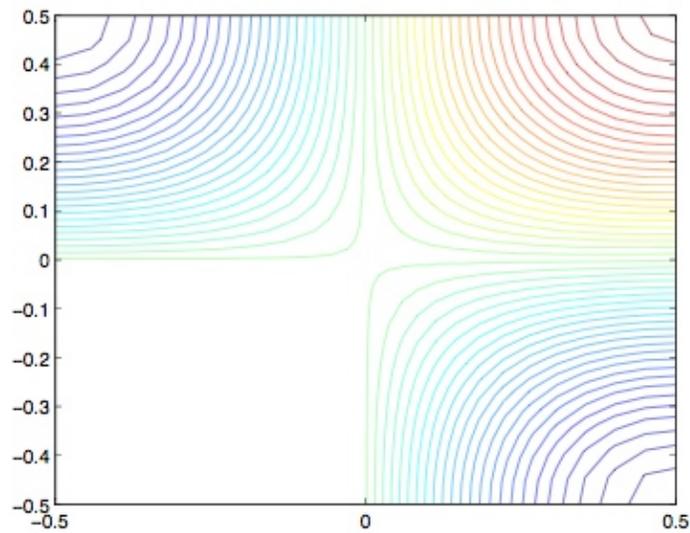


Figura 9. Curvas de nivel para la solución aproximada.

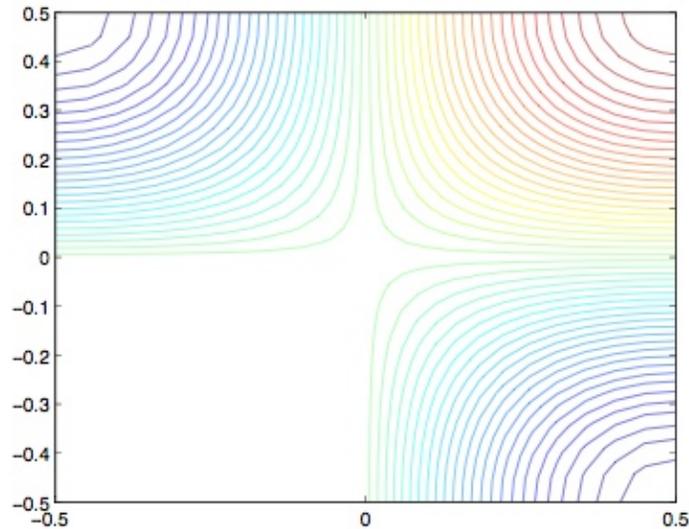


Figura 10. Curvas de nivel para la solución exacta.

Apéndice

A. Algoritmo

Listing 1. Seudocódigo en FORTRAN 90 para construir la malla radial en el dominio L invertida, coordenadas y relación de nodos locales con globales.

```
Datos entrada :
A=(a(1),a(2)),B=(b(1),b(2)),C=(c(1),c(2)),NTB,NTA.
Datos salida :
coor(i,j), Coordenadas j de los nodos i
            de acuerdo a la numeracion
            preestablecida.
nodo(i,j): nodo global al que corresponde el nodo
            local i del triangulo j.
```

Inicio instrucciones:

```
incB=1/NTB;
NTAS2=NTA/2;
! Parte inferior de la L invertida :
incA1=(c(2)-a(2))/NTAS2;
incA2=(b(2)-a(2))/NTAS2;
NNI=(NTA+1)*NTB;
! calculo de las lineas verticales :
coor(1,NNI+1)=c(1);
coor(2,NNI+1)=a(2);
do j=1,NTB
```

```

t=incB*j;
w(j)=(b(1)-c(1))*t^2 + c(1);
! numeramos los nodos de la linea horizontal
coor(1,j)=w(j);
coor(2,j)=a(2);
end do
! se calculan las y para cada recta radial:
do i=1,NTAS2
! primer nodo de cada linea:
coor(1,NNI+1+i)=c(1);
y1=a(2)+incA1*i;
coor(2,NNI+1+i)=y1;
! nodos restantes de las lineas radiales
y2=a(2)+incA2*i;
do j=1,NTB
z=(w(j)-b(1))*(y2-y1)/(b(1)-c(1))+y2;
! numeramos los nodos de la linea diagonal
coor(1,i*NTB+j)=w(j);
coor(2,i*NTB+j)=z;
end do
end do
! Parte superior de la L invertida:
incx1=(c(1)-a(1))/NTAS2;
incx2=(b(1)-a(1))/NTAS2;
! primer nodo linea vertical
coor(1,(NTA+1)*(NTB+1))=a(1);
coor(2,(NTA+1)*(NTB+1))=c(2);
! los otros nodos
do j=1,NTB
! calculo de las lineas horizontales :
t=incB*j;
w(j)=(b(2)-c(2))*t^2 + c(2);
! numeramos los nodos de la linea vertical
coor(1,NTA*NTB+j)=a(1);
coor(2,NTA*NTB+j)=w(j);
end do
! se calculan las x para cada recta :
NTAS2M1=(NTAS2+1)*NTB;
do i=1,NTAS2-1 ! (para cada linea diagonal de derecha a izq)
x1=c(1)-incx1*i;
! primer nodo:
coor(1,NTB*(NTA+1)+NTAS2+i+1)=x1;
coor(2,NTB*(NTA+1)+NTAS2+i+1)=c(2);
! los otros nodos.
x2=b(1)-incx2*i;
NTAS2M1NTB=NTAS2M1+(i-1)*NTB;
do j=1,NTB
z=(w(j)-c(2))*(x2-x1)/(b(2)-c(2))+ x1;
! numeramos los nodos de la linea diagonal
coor(1,NTAS2M1NTB+j)=z;

```

```

        coor(2,NTAS2M1NTB+j)=w(j);
    end do
end do

! se establece la relacion entre nodos locales con nodos globales.
! triangulos interiores:

do i=0,NTA-1
    indt=2*(NTB-1)*i
    do j=1,NTB-1
        numt=indt + 2*j
        indn=numt/2 + i
        nodo(1,numt-1)=indn
        nodo(2,numt-1)=indn+1
        nodo(3,numt-1)=indn+NTB+1
        nodo(1,numt)=indn
        nodo(2,numt)=indn+NTB+1
        nodo(3,numt)=indn+NTB
    end do
end do

! triangulos frontera izquierda:

indt=2*(NTB-1)*NTA
indn=NTB*(NTA+1)
do i=1,NTA
    numt=indt + 2*i
    nodo(1,numt-1)=indn + i
    nodo(2,numt-1)=(i-1)*NTB + 1
    nodo(3,numt-1)=i*NTB + 1
    nodo(1,numt)=indn + i
    nodo(2,numt)=i*NTB + 1
    nodo(3,numt)=indn + i + 1
end do

fin del algoritmo.

```

Referencias

- [1] Bresiz, H. *Análisis funcional. Teoría y aplicaciones*, Alianza Editorial, 1978.
- [2] Ciarlet, P. G. *The finite element method for elliptic problems*, North Holland, 1978.
- [3] Girault, V., Raviart, P. *Finite element methods for Navier-Stokes equations*, Springer, Berlin, 1986.
- [4] Glowinski, R., Pan, T-W, Periaux, J. *A fictitious domain method for Dirichlet problem and applications*, Comput. Meth. Appl. Mech. Eng., 111, 283-303, 1994.
- [5] López, J., Chablé, D., Juárez, L. *Generación de campos de viento por métodos variacionales*, Revista de Ciencias Básicas, UJAT, Vol 6, Num 2, pp. 47-68, 2007.
- [6] Marco A. Núñez, Ciro F. Flores, L. Héctor Juárez, *Interpolation of hydrodynamic velocity with the continuity equation*, Journal of Computational Methods in Science and Engineering, Artículo aceptado para su publicación.