
METODO DE PANELES BIDIMENSIONAL

*Daniel Fortin**

*Paulo Porta**

Se presenta el Método de Paneles como alternativa a los esquemas clásicos en diferencias finitas para el tratamiento de flujos de fluidos ideales. Se muestra además la aptitud y economía computacional del método y se aplica como muestra al cálculo del flujo alrededor de un perfil bidimensional. Y se discuten posibles extensiones para tratar modelos de flujo más reales.

Es sabido que en los problemas de Mecánica de Fluidos es prácticamente imposible hallar soluciones cerradas a problemas reales. Esta imposibilidad radica en la naturaleza no lineal de los problemas que aparecen en las aplicaciones. Sin embargo se pueden realizar linealizaciones o hipótesis que simplifiquen el problema permitiendo el tratamiento analítico. Cuando la geometría del problema es compleja, la aproximación lineal que se haga de las ecuaciones de gobierno fracasa, de modo que la

* Universidad Tecnológica Nacional, Facultad Regional Haedo, Grupo CB.DI. Colaboradores: Héctor Mondelo y Roberto Gava.

única opción que resta es la de recurrir a algún método numérico. Entre los problemas de interés creciente se destaca el de determinar las fuerzas y parámetros aerodinámicos de flujos alrededor de obstáculos de geometría arbitraria. Las técnicas numéricas usuales resuelven el flujo en todo el dominio. Sin embargo, cuando se quiere especificar el valor de las variables de campo sólo sobre el obstáculo aparece como alternativa el Método de Paneles. Este método está basado sobre el principio de superposición lineal y fue aplicado a flujos lineales, como los gobernados por la ecuación de Laplace. Esta familia de flujos es de gran importancia, tanto teórica como práctica: a partir de ella se puede construir toda la teoría de la sustentación (uno de los problemas fundamentales de la aerodinámica), se pueden estudiar flujos no estacionarios impulsivos a grandes números de Reynolds (esto es de importancia a la hora de calcular performances de automóviles de alta velocidad) y se pueden determinar configuraciones de flujo, líneas de corriente, etcétera.

El método

Sea un contorno S , como el de la figura, inmerso en un dominio, que puede ser infinito en el cual se satisface una relación del tipo:

$$\nabla^2 \phi^* = 0 \quad (1)$$

donde ϕ^* es el potencial total que sale de sumar el potencial en el infinito ϕ_∞ con el potencial que introduce el obstáculo como perturbación ϕ .

La solución general de [1] puede construirse como suma de una fuente de intensidad σ y un doblete de intensidad μ sobre el contorno S , la que pondremos en términos de la identidad de Green como:

$$\phi^*(\mathbf{r}) = -\frac{1}{4\pi} \int_{S_b+W} [\sigma 1/r - \mu \mathbf{n} \cdot \nabla (1/r)] dS + \phi_\infty \quad (2)$$

el versor \mathbf{n} orienta S en la dirección del salto de potencial, mientras que ϕ_∞ puede calcularse como:

$$\phi_\infty = V_\infty \cdot \mathbf{r} \quad (3)$$

y puede pensarse como una constante que se introduce para ajustar la condición de borde lejos del obstáculo.

La formulación anterior admite una solución no única que satisface las condiciones de contorno. Es decir que hay muchas combinaciones de fuentes y dobletes que satisfacen la [2]. Si se desea simular el espesor, bastaría con usar fuentes, mientras que para modelar los fenómenos de sustentación sería necesario utilizar singularidades asimétricas como torbellinos o

dobletes.

Una vez elegido el tipo de discontinuidad a utilizar, se debe imponer la condición de contorno sobre S_b , a saber, que la velocidad normal al obstáculo debe ser nula. Esto puede escribirse como:

$$\nabla (\phi + \phi_\infty) \cdot \mathbf{n} = 0 \quad (4)$$

sobre S_b , mientras que en el infinito se debe verificar que ϕ decazca con r de modo que:

$$\ell_{r \rightarrow \infty} \nabla \phi = 0 \quad (5)$$

vale aclarar que esta condición es cumplida por las singularidades que usualmente se emplean. Tomando el gradiente de [2]:

$$\nabla \phi^*(\mathbf{r}) = \frac{1}{4\pi} \int_{S_{b+W}} \left[\mu \nabla \left(\frac{\partial(1/r)}{\partial n} \right) \right] dS - \frac{1}{4\pi} \int_{S_b} \left[\sigma \nabla \left(\frac{1}{r} \right) \right] dS + \nabla \phi_\infty \quad (6)$$

Y de acuerdo a [4] multiplicando escalarmente por \mathbf{n} y reordenando obtenemos:

$$\left\{ \frac{1}{4\pi} \int_{S_{b+W}} \left[\mu \nabla \left(\frac{\partial(1/r)}{\partial n} \right) \right] dS - \frac{1}{4\pi} \int_{S_b} \left[\sigma \nabla \left(\frac{1}{r} \right) \right] dS \right\} \cdot \mathbf{n} = \nabla \phi_\infty \cdot \mathbf{n} \quad (7)$$

Esta última expresión es la base del método que delinearemos a continuación y se cumple para todo punto del obstáculo S_b . Siendo S_b un contorno cerrado y con esta condición, el potencial interior es constante y puede ponerse como $\phi_{int} = cte$.

Lo que sigue ahora es la implementación de lo antes descrito; el procedimiento típico a seguir es el que esquematizamos a continuación:

1. Discretización geométrica y selección del elemento de singularidad

Se discretiza S_b usando m paneles, cada uno de los cuales tiene asociada una singularidad. Esta singularidad se caracteriza con dos parámetros, i) el tipo de singularidad, es decir si se trata de una fuente, un torbellino o un doblete y ii) por cómo está distribuida dicha singularidad en el elemento, esto es si está concentrada (orden cero), distribuida en forma uniforme (orden uno), etcétera.

Construcción de la matriz de influencia

La matriz de influencia es la matriz que contiene información acerca de la influencia de cada uno de los paneles sobre los puntos de colocación. Esta influencia se calcula para una intensidad unitaria de la singularidad. La forma general de calcular la matriz es resolver el primer miembro de la [7] para intensidades σ y μ unitarios. Es decir que podemos definir el coeficiente de influencia a_{ij} como la velocidad normal en el punto de colocación i debido a la contribución de una singularidad unitaria en el panel j .

Cálculo del RHS

El RHS (Right Hand Side) es el segundo miembro de la ecuación [7]. Este se construye a partir de la condición de borde que impone que el gradiente del potencial $\phi_{,n}$ sea tangente al obstáculo. Evidentemente, esto debe cumplirse sobre toda la frontera del obstáculo; sin embargo el método sólo verificará la condición en los puntos de control. De este modo, el sistema de ecuaciones diferenciales se reduce a un sistema de ecuaciones algebraico, con tantas incógnitas como puntos de control se desee tener.

Solución del sistema de ecuaciones algebraico

Para la resolución del sistema algebraico se puede utilizar cualquiera de las técnicas disponibles en los textos de Análisis Numérico (para más detalles, Cfr. referencia [3]). La solución de este sistema será el vector con los valores de σ_j que corresponden a cada panel.

Cálculos secundarios

Una vez establecidas las incógnitas de nuestro problema podemos a partir de ellas calcular otras variables derivadas como velocidades, presiones, fuerzas o coeficientes, etcétera.

Una aplicación bidimensional

Haremos una aplicación de los seis pasos descriptos al flujo alrededor de un perfil teórico con ángulo de ataque nulo. El perfil sobre el que trabajaremos es un perfil de Van de Vooren. Estos perfiles teóricos tienen solución exacta, de modo que probaremos el método al máximo, contrastando nuestros resultados numéricos contra éstos. Recorreremos los seis pasos tal como fueron presentados:

1. Discretización geométrica y selección del elemento de singularidad

Comenzaremos por dividir el perfil en 99 paneles, definidos por 100 nodos. Para nuestra simulación utilizaremos dobletes lineales de intensidad constante, es decir que el tipo de singularidad en el elemento será un doblete y éste será equivalente a dos vórtices concentrados en los extre-

mos del panel. Según la clasificación que hiciéramos más arriba se trata de fuentes de orden cero. Estos elementos inducen velocidades de la forma:

$$u_p = \frac{\mu}{2\pi} \left[\frac{z}{(x - x_1)^2 + z^2} - \frac{z}{(x - x_2)^2 + z^2} \right] \quad (8)$$

$$w_p = \frac{-\mu}{2\pi} \left[\frac{x - x_1}{(x - x_1)^2 + z^2} - \frac{x - x_2}{(x - x_2)^2 + z^2} \right] \quad (9)$$

donde las variables están definidas como en la figura.

Las coordenadas involucradas en estas definiciones son coordenadas locales del panel, de modo que para referir estas velocidades inducidas por la singularidad del panel a un sistema global, habrá que introducir una transformación de coordenadas (usualmente se trata de una traslación y una rotación). Resumiremos esto diciendo que las velocidades inducidas referidas al sistema global, por la singularidad del j -ésimo panel de extremos $(x_j, z_j, x_{j+1}, z_{j+1})$ en las coordenadas globales en el punto de coordenadas globales (x, z) se obtienen al aplicar un operador \mathfrak{S} a las variables antes mencionadas:

$$(u, w) = \mathfrak{S}(\mu_j, x, z, x_j, z_j, x_{j+1}, z_{j+1}) \quad (10)$$

A los efectos de la implementación, este operador será un módulo (subprograma o subrutina) de preprocesamiento de los datos geométricos disponibles luego de la discretización geométrica.

2. Construcción de la matriz de influencia

En esta fase se impone la condición de contorno (velocidad normal nula) a cada punto de colocación. Esto se implementa como un doble lazo anidado, ya que se requiere que la velocidad normal inducida por cada panel se anule en todos. Esta condición se impone a través de la relación:

$$\alpha_{i,j} = (u, w)_{i,j}^1 \cdot \mathbf{n} = \mathfrak{S}(1, x_i, z_i, x_j, z_j, x_{j+1}, z_{j+1}) \cdot \mathbf{n} \quad (11)$$

donde el punto genérico de coordenadas (x,y) se transforma en el i -ésimo punto de colocación de coordenadas (x_i,z_i) y se establece para intensidad unitaria, es decir $\mu = 1$.

Para el tipo particular de elemento de singularidad que hemos elegido, la velocidad que induce un panel sobre su punto de colocación es

$$\left(0, \frac{\mu}{\pi} \frac{2}{(x_2 - x_1)}\right)$$

de modo que el coeficiente de influencia correspondiente es:

$$a_{ii} = \frac{-2}{\pi \cdot \Delta c_i}$$

3. Cálculo del RHS

Especificando la condición de borde, en el punto de control 1, resulta la siguiente ecuación algebraica:

$$\frac{-2}{\pi \cdot \Delta c_i} \cdot \mu_1 + a_{12} \cdot \mu_2 + \dots + a_{1n} + (U_\infty, W_\infty) \cdot n_1 = 0$$

donde por supuesto $W_\infty = 0$

La velocidad normal de la corriente libre es transferida hacia el lado derecho de la ecuación y el vector RHS_1 es armado como:

1. Solución del sistema de ecuaciones algebraico

Especificando las condiciones de borde para cada uno de los puntos de colocación resulta en un sistema de ecuaciones resultante de la forma:

$$\begin{bmatrix} a_{11} & a_{12} & \dots & \dots & a_{1w} & \mu_1 & RHS_1 \\ a_{21} & a_{22} & \dots & \dots & a_{2w} & \mu_2 & RHS_2 \\ a_{31} & a_{32} & a_{33} & \dots & a_{3w} & \mu_3 & RHS_3 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & a_{nn} & a_{nw} & \mu_n & RHS_n \\ 1 & 0 & \dots & -1 & 1 & \mu_w & 0 \end{bmatrix} = \quad (12)$$

Sin embargo, la condición de Kutta para la circulación en el borde de fuga es cero requiere de un panel de estela (wake) adicional que será cancelado con el siguiente vértice:

$$(\mu_1 - \mu_n) + \mu_w = 0$$

una combinación de esta última con las N condiciones de borde resulta en un sistema lineal de N+1 ecuaciones ya visto, de modo que es posible en virtud de su estructura bien comportada, usar un método standard para su resolución.

2. Cálculos de presiones y cargas

Una vez conocido los dobletes μ_p , la perturbación de velocidad tangencial en cada punto de colocación puede ser calculada por suma de las velocidades inducidas de cada panel, esta velocidad tangencial es:

$$q_i = \sum_{j=1}^{N+1} (u, w)_{ij} \cdot t_i$$

donde $(u, w)_i$ es el resultado de la ecuación [9] y t_i es la tangente local y (N+1) por la componente agregada debido a la estela, si evaluamos la velocidad tangencial debida al panel por sí mismo:

$$q_i = \frac{1}{2} \cdot \frac{\partial \mu(i)}{\partial l}$$

donde l representa la distancia a lo largo de la línea de superficie.

Determinar el *coeficiente de presiones* C_p en función de las velocidades que hemos determinado. Para obtener, C_p en cada punto de colocación, aplicamos la relación:

$$C_p = 1 - \frac{(Q_{i\infty} + q_i)^2}{Q_\infty^2}$$

donde

$$(Q_{i\infty})_j = t_j \cdot Q_\infty$$

De la fórmula de Kutta Joukowski calculamos la sustentación:

$$\Delta L_i = \rho Q_\infty \Gamma_j = -\rho Q_\infty (\mu_{j+1} - \mu_j)$$

La sustentación total es entonces

$$L = \sum_{j=1}^N \Delta L_j$$

Resultados

1. El código que presentaremos a continuación fue realizado VisualBasic 3.0, está desarrollado pensando en una computadora que tenga un monitor del tipo EGA o VGA, debido a las rutinas gráficas necesarias.

2. Para la validación del código usaremos un perfil teórico, que nos brinda un método de cálculo exacto de la distribución de presión. Este perfil es llamado *Perfil de Van de Vooren*:

Sea un circunferencia de radio a se efectuará una transformación:

$$Y = \frac{(f-a)^k}{(f-ae)^{k-1}} + l$$

El centro de la circunferencia es en el origen de coordenadas en el plano f y el radio es a , ϵ es el espesor del perfil, k controla el ángulo del borde de fuga y l determina el largo de la cuerda. Para el círculo:

$$f = ae^{i\theta}$$

La transformación quedará:

$$Y = \frac{[a(\cos \theta - 1) + ia \sin \theta]^k}{[a(\cos \theta - \epsilon) + ia \sin \theta]^{k-1}} + l$$

Note que si $\theta=0$ corresponde $Y=l$, el borde de fuga. Para $\theta=\pi$, el borde de ataque es dado por:

$$Y = \frac{-a2^k}{(1+\epsilon)^{k-1}} + l$$

Para el perfil de Van de Vooren, el largo de la cuerda está dado por:

$$2l = \frac{a2^k}{(1+\epsilon)^{k-1}}$$

Las coordenadas $\{x,z\}$ están dadas por:

$$x = \frac{r_1^k}{r_2^{k-1}} \cdot [\cos k\theta_1 \cdot \cos(k-1)\theta_2 + \sin k\theta_1 \cdot \sin(k-1)\theta_2]$$

$$z = \frac{r_1^k}{r_2^{k-1}} \cdot [\sin k\theta_1 \cdot \cos(k-1)\theta_2 - \cos k\theta_1 \cdot \sin(k-1)\theta_2]$$

$$r_1 = \sqrt{(a \cdot \cos \theta - a)^2 + a^2 \sin^2 \theta}$$

$$r_2 = \sqrt{(a \cdot \cos \theta - \epsilon a)^2 + a^2 \sin^2 \theta}$$

$$\theta_1 = \arctan \frac{a \sin \theta}{a \cos \theta - \epsilon a} + \pi$$

$$\theta_2 = \arctan \frac{a \sin \theta}{a \cos \theta - \epsilon a} + n_1 \pi$$



Donde n_1 depende del cuadrante donde θ_2 está siendo evaluado (0 en el primer cuadrante, 1 en el segundo y tercero y 2 en el cuarto).

La distribución de velocidad está dada como solución en el círculo en el plano más la transformación:

$$u = 2Q_\infty \frac{r_2^k}{r_1^{k-1}} \frac{\sin \alpha - \sin(\alpha - \theta)}{D_1^2 + D_2^2} \cdot (D_1 \sin \theta + D_2 \cdot \cos \theta)$$

$$w = -2Q_\infty \frac{r_2^k}{r_1^{k-1}} \frac{\sin \alpha - \sin(\alpha - \theta)}{D_1^2 + D_2^2} \cdot (D_1 \cos \theta - D_2 \cdot \sin \theta)$$

$$A = \cos(k-1)\theta_1 \cos k\theta_2 + \sin(k-1)\theta_1 \sin k\theta_2$$

$$B = \sin(k-1)\theta_1 \cos k\theta_2 - \cos(k-1)\theta_1 \sin k\theta_2$$

$$D_0 = a(1 - k + k\epsilon)$$

$$D_1 = A(a \cos \theta - D_0) - Ba \sin \theta$$

$$D_2 = A(a \sin \theta) + B(a \cos \theta - D_0)$$

Por lo tanto podemos calcular la distribución exacta de presión:

$$C_p = 1 - \frac{u^2 + w^2}{Q_\infty^2}$$

En el Anexo presentaremos el código generado para el cálculo y los resultados de los cálculos.

Perspectivas

Hemos presentado el Método de Paneles como una alternativa económica para el cálculo de flujos alrededor de obstáculos. Sin embargo nos hemos restringido al tratamiento de problemas bidimensionales potenciales. En esta sección discutiremos algunas modificaciones sencillas para exten-

der el método a otros regímenes de flujo.

Flujos tridimensionales

Esta es la primera extensión natural del método y la más sencilla. El único cuidado que hay que tener es en imponer correctamente las condiciones de contorno sobre la estela detrás del obstáculo, de manera de verificar la condición de Kutta. Para esto será necesario hacer un modelo de la estela en el que se deberán especificar

1. La intensidad de la estela en el borde de fuga
2. La forma y ubicación de ella

Esto no es una tarea trivial ya que los parámetros de la estela no se conocen *a priori*. En general se adopta un criterio aproximado y basado en la experiencia. (Para más detalles Cfr. [2]).

Flujos compresibles

Para tener en cuenta los efectos de compresibilidad un camino viable y sencillo es el de incorporar la fórmula de Prandtl-Glauert, definiendo el factor de compresibilidad β :

$$\beta = \sqrt{1 - M^2}$$

Para simplicidad de la discusión, supongamos que el flujo es paralelo al eje x , de modo que es posible definir un potencial compresible ϕ_M como función del potencial incompresible ϕ . Este efecto se manifestará como un incremento en las fuerzas aerodinámicas.

Efectos de la capa límite

Cuando se estudian flujos a grandes números de Reynolds, los efectos viscosos quedan confinados a una región delgada de fluido vecina al sólido. Cuando se hacen las aproximaciones necesarias para que el flujo sea potencial, estos efectos se pierden y en algunas aplicaciones puede resultar perjudicial. Para salvar este inconveniente es posible resolver el problema en la forma descrita y usar esta solución como entrada para algún modelo simplificado de capa límite, de modo de poder estimar factores de fricción, coeficientes de arrastre locales, etc. De la calidad del modelo de capa límite dependerá la validez de la estimación que se haga de estos parámetros.

Flujos no estacionarios

Tener en cuenta los efectos del flujo no estacionario no es tan sencillo. Se debe contar con los datos del movimiento del obstáculo o al menos con la ley de variación del flujo en el tiempo. Pero, sin embargo, la mayor dificultad radica en establecer la condición de Kutta no estacionaria: esta tarea implica hacer un modelo de la estela no estacionaria pero *a priori*. Las condiciones que usualmente se pueden implementar restringen los problemas a los casos de velocidades verticales del obstáculo y ángulos de ataque pequeños.

Chorros (Jets)

Muchos problemas de la dinámica de fluidos involucran entre sus componentes chorros con gran contenido de energía cinética, en los cuales el valor de la presión de estancamiento y de la velocidad son muy distintos de los del flujo en la corriente libre. Si bien el flujo dentro del chorro es compresible y el proceso de mezcla que sufre es altamente viscoso, hay varios modelos que permiten simular los efectos del chorro sobre el campo; éstos se basan en simular el chorro con paneles cuya singularidad sea un doblete que varía linealmente, distribuido en la superficie del chorro. Así la superficie del chorro se convierte en una capa de contacto, sobre la que se cumplirán las condiciones de borde.

Entradas de aire (Inlets)

Cuando se desea simular una entrada de aire, una técnica sencilla y efectiva es la de permitir en los paneles de dicha entrada cierta permeabilidad, caracterizada por una velocidad V_n que se impone como condición de borde. Es decir, que en lugar de imponer la condición de flujo normal nulo, se impone un valor de flujo prescripto.

BIBLIOGRAFIA

FLETCHER J., Numerical Methods in Fluid Dynamics, Spriger Series in Computational Physics, Springer Verlag.

KATZ & PLOTKIN, Low-speed Aerodynamics. From Wing Theory to Panel Methods, McGraw-Hill Series in Aerospace Sciences, McGraw-Hill.

DALHQUIST & BJORK, Numerical Methods, Academic Press.

Anexo

```
Global alfa1, a(), r(), gridxz(), nropaneles, u(), l(), x()
Sub Form_Load ()
'Qinf = 1
```

```

label2(0).Caption = "C1"
label2(1).Caption = "L"
label2(2).Caption = "c1"
label3(0).Caption = "0"
label3(2).Caption = "0"
label3(1).Caption = "0"
Combo1.Text = "Coeficientes"
form1.Show
Const pi = 3.14159265
nperfil = 0
metodo = 0
AbrirArchivo u(), l(), x(), nropaneles, gridxz(), nperfil, metodo
'AumentaNroPaneles nropaneles,gridxz()

'dimensionando las coordenadas del perfil, ptos de ctrl y grilla

ReDim ctrlz(nropaneles), ctrlx(nropaneles), tem(nropaneles)

'dimensionando las coordenadas del perfil, ptos de ctrl y grilla

'dimensionando vector de angulos de perfil,matriz de coeficientes de
'influencia "a(i,j)" izq y b(i,j) der y r(i) salida de resultados
ReDim Alfa(nropaneles + 1), a(nropaneles + 1, nropaneles + 2),
b(nropaneles + 1, nropaneles + 2), r(nropaneles + 2)

'angulo de ataque

alfa1 = 0 / 180 * pi
'calculo de los ptos de control superiores
For i = 1 To nropaneles
    ctrlz(i) = (gridxz(i + 1, 2) + gridxz(i, 2)) / 2
    ctrlx(i) = (gridxz(i + 1, 1) + gridxz(i, 1)) / 2
    DoEvents
Next i

'graficacion del perfil y ptos de control

max_x = gridxz(1, 1)
max_z = gridxz(1, 2)
For i = 1 To nropaneles + 1
    If Abs(max_x) < Abs(gridxz(i, 1)) Then max_x = Abs(gridxz(i, 1))

```

```

    If Abs(max_z) < Abs(gridxz(i, 2)) Then max_z = Abs(gridxz(i, 2))
Next i
If max_x > max_z Then
    max_z = (max_x * picture1.Height / picture1.Width) / 2
Else
    max_x = max_z * picture1.Height / picture1.Width
End If
picture1.ScaleWidth = (max_x)
picture1.ScaleHeight = (max_z * 2)
picture1.Scale (0, max_z)-(max_x), -max_z)
picture1.Line (0, 0)-(max_x, 0)
For i = 1 To nropaneles
    picture1.Line (gridxz(i, 1), gridxz(i, 2))-(gridxz(i + 1, 1), gridxz(i + 1, 2)), QBColor(6))
Next i
picture1.DrawWidth = 2
For i = 1 To nropaneles
    picture1.PSet (ctrlx(i), ctrlz(i)), RGB(255, 0, 0)
Next i

'transformacion de coordenadas del perfil en paneles
'en la direccion de las agujas del reloj

For i = 1 To nropaneles + 1
    picture1.PSet (gridxz(i, 1), gridxz(i, 2)), RGB(125, 125, 0)
Next i
picture1.DrawWidth = 1
'Encontrando los angulos de los paneles para las matrices de conversion
de
' coordenadas globales en locales
,
For i = 1 To nropaneles
    DeltaX = gridxz(i + 1, 1) - gridxz(i, 1)
    DeltaY = gridxz(i + 1, 2) - gridxz(i, 2)
    Alfa(i) = Abs(Atn(DeltaY / DeltaX))
Next i
'Armado de la matriz de coef de influencia

For i = 1 To nropaneles
    For j = 1 To nropaneles

```

'transformar los ptos de control a coord locales

```

xmx0 = ctrlx(i) - gridxz(j, 1)
zmx0 = ctrlz(i) - gridxz(j, 2)
difx = gridxz(j + 1, 1) - gridxz(j, 1)
difz = gridxz(j + 1, 2) - gridxz(j, 2)

```

'coordenadas del panel para el pto

```

xptocol = xmx0 * Cos(Alfa(j)) + zmx0 * Sin(Alfa(j))
z = -xmx0 * Sin(Alfa(j)) + zmx0 * Cos(Alfa(j))

```

'coord p vector r2 no se calcula z pues es = 0

```

xr2 = difx * Cos(Alfa(j)) + difz * Sin(Alfa(j))
'xr2 = Sqr(difx ^ 2 + difz ^ 2)

```

'vectores

```

r1 = Sqr(xptocol ^ 2 + z ^ 2)
r2 = Sqr((xptocol - xr2) ^ 2 + z ^ 2)

```

'seleccionar los elementos de la diagonal

'si no se usa mas r1 y r2 no elevar al cuadrado y sacar la raiz cuadrada

If i = j Then

```

upanel = 0
wpanel = -1 / (pi * xptocol)
'wpanel = -2 / (pi * xr2)

```

Else

```

upanel = 1 / (2 * pi) * (z / r1 ^ 2 - z / r2 ^ 2)
wpanel = -1 / (2 * pi) * (xptocol / r1 ^ 2 - (xptocol - xr2) / r2 ^ 2)

```

End If

'transformar a globales las veloc inducidas

```

ug = upanel * Cos(-Alfa(j)) + wpanel * Sin(-Alfa(j))
w = -upanel * Sin(-Alfa(j)) + wpanel * Cos(-Alfa(j))

```

'calculo de los coeficientes de influencia

```

a(i, j) = -ug * Sin(Alfa(i)) + w * Cos(Alfa(i))

```

```

b(i, j) = ug * Cos(Alfa(i)) + w * Sin(Alfa(i))
Next j

```

' influencia del panel de estela

```

r1 = ((ctrlx(i) - gridxz(nropaneles + 1, 1)) ^ 2 + (ctrlz(i) -
gridxz(nropaneles + 1, 2)) ^ 2)
'si no se usa mas r1 no elevar al cuadrado y sacar la raiz cuadrada
ug = 1 / (2 * pi) * (ctrlz(i) / r1)
w = -1 / (2 * pi) * (ctrlx(i) - gridxz(nropaneles + 1, 1)) / r1
DeltaAlfa = Alfa(i)
a(i, nropaneles + 1) = -ug * Sin(DeltaAlfa) + w * Cos(DeltaAlfa)
b(i, nropaneles + 1) = ug * Cos(DeltaAlfa) - w * Sin(DeltaAlfa)
a(i, nropaneles + 2) = Qinf * Cos(alfa1) * Sin(Alfa(i)) - Qinf * Sin(alfa1)
* Cos(Alfa(i))
Next i

```

' condicion de borde de kutta

```

For i = 1 To nropaneles + 2
  a(nropaneles + 1, i) = 0
Next i
a(nropaneles + 1, 1) = -1
a(nropaneles + 1, nropaneles) = 1
a(nropaneles + 1, nropaneles + 1) = -1

```

```

crout a(), r()
Dl = 0
lift = 0

```

```

form1.Combo1.AddItem "Dif de CL"
j = 0
For i = 1 To nropaneles
  Dl = -.125 * Qinf * (r(i + 1) - r(i))
  lift = lift + Dl
  j = j + 1
  form1.Combo1.AddItem Dl
Next i
j = j + 1
form1.Combo1.AddItem "Coef. de Presión"
max_cp = 0

```

```

For i = 1 To nropaneles
    temp = 0
    For j = 1 To nropaneles + 1
        temp = temp + b(i, j) * r(j)
    Next j
    If i <> 1 And i <> nropaneles Then
        r1 = Sqr((ctrlx(i + 1) - ctrlx(i - 1)) ^ 2 + (ctrlz(i + 1) - ctrlz(i - 1)) ^ 2)
        vloc = (r(i + 1) - r(i - 1)) / r1
    ElseIf i = 1 Then
        r1 = Sqr((ctrlx(2) - ctrlx(1)) ^ 2 + (ctrlz(2) - ctrlz(1)) ^ 2)
        vloc = (r(2) - r(1)) / r1
    ElseIf i = nropaneles Then
        r1 = Sqr((ctrlx(nropaneles) - ctrlx(nropaneles - 1)) ^ 2 +
(ctrlz(nropaneles) - ctrlz(nropaneles - 1)) ^ 2)
        vloc = (r(nropaneles) - r(nropaneles - 1)) / r1
    End If
    veloc = Qinf * Cos(alfa1) * Cos(Alfa(i)) + Qinf * Sin(alfa1) * Sin(Alfa(i))
    + temp + vloc / 2
    cp = 1 - (veloc ^ 2) / Qinf ^ 2

    form1.Combo1.AddItem cp
    tem(i) = cp
    If max_cp < Abs(tem(i)) Then max_cp = Abs(tem(i))
Next i
d = 3
max_cp = 2
For i = 1 To nropaneles - 2
    If i >= nropaneles / 2 Then d = 2
    picture1.Line (gridxz(i, 1), -tem(i) * Abs(max_z / max_cp))-(gridxz(i +
1, 1), -tem(i + 1) * Abs(max_z / max_cp)), QBColor(d)
Next i

label3(0).Caption = r(nropaneles + 1)
label3(1).Caption = lift
label3(2).Caption = lift / 1175 * 9.8

End Sub

```


Sub AbrirArchivo (u(), l(), x(), nropaneles, gridxz(), nperfil, metodo)

form1.Show

metodo = 3

aum = 6

ReDim x(n), l(n), u(n), gridxz(n * 2 * (aum - 1), 2)

nropaneles = ((n - 1) * aum) * 2

Select Case metodo

Case 1

AumentaPaneles x(), l(), u(), aum

Case 2

AumentaNroPaneles x(), l(), u(), aum

Case 3

AumentaPan gridxz(), aum

nropaneles = aum

nperfil = (aum + 1) / 2

Exit Sub

Case Else

aum = 17

End Select

ReDim gridxz(aum * 2, 2)

n = aum

i = 0

nropaneles = aum * 2 - 2

For ii = n To 2 Step -1

i = i + 1

gridxz(i, 1) = x(ii)

gridxz(i, 2) = l(ii)

Next ii

For ii = 1 To n

i = i + 1

gridxz(i, 1) = x(ii)

gridxz(i, 2) = u(ii)

Next ii

nperfil = n

End Sub

Sub AumentaPan (gridxz(), aum)

Static a, x

e = .15'coefic. de espesor

ak = $-(5 * 3.14159 / 180) / 3.14159 + 2$ 'ángulo del borde de fuga

al = $0 * \pi / 180$ 'ángulo de ataque en radianes

```

m = 100' nro de paneles
itheta = 360 / m
tl = 1
a = 2 * tl * (e + 1) ^ (ak - 1) / 2 ^ ak

```

```

ReDim AntesX(m + 1), AntesU(m + 1)

```

```

j1 = 0

```

```

For i = 0 To 360 Step itheta

```

```

    j1 = j1 + 1

```

```

    If i = 0 Or i = 360 Then

```

```

        x = 1

```

```

        y = 0

```

```

        cp = 1

```

```

        AntesX(j1) = x

```

```

        AntesU(j1) = y

```

```

        If ak = 2 And i = 0 Then 100

```

```

        If ak = 2 And i = 360 Then 100

```

```

        GoTo 100

```

```

    Else

```

```

        th = i / 57.2958

```

```

        r1 = Sqr((a * (Cos(th) - 1)) ^ 2 + (a * Sin(th)) ^ 2)

```

```

        r2 = Sqr((a * (Cos(th) - e)) ^ 2 + (a * Sin(th)) ^ 2)

```

```

        If th = 0 Then

```

```

            th1 = 1.5708

```

```

        Else

```

```

            th1 = (Atn((a * Sin(th)) / (a * (Cos(th) - 1)))) + 3.1415927

```

```

        End If

```

```

        If Cos(th) - e < 0 And Sin(th) > 0 Then

```

```

            th2 = (Atn((a * Sin(th)) / (a * (Cos(th) - e)))) + 3.1415927

```

```

        ElseIf Cos(th) - e < 0 And Sin(th) < 0 Then

```

```

            th2 = (Atn((a * Sin(th)) / (a * (Cos(th) - e)))) + 3.1415927

```

```

        ElseIf Cos(th) - e > 0 And Sin(th) < 0 Then

```

```

            th2 = (Atn((a * Sin(th)) / (a * (Cos(th) - e)))) + 2 * 3.1415927

```

```

        Else

```

```

            th2 = (Atn((a * Sin(th)) / (a * (Cos(th) - e))))

```

```

        End If

```

```

        com1 = ((r1 ^ ak) / (r2 ^ (ak - 1))) / ((Cos((ak - 1) * th2)) ^ 2 + (Sin((ak - 1) * th2)) ^ 2)

```

```

        x = com1 * (Cos(ak * th1) * Cos((ak - 1) * th2) + Sin(ak * th1) *

```

```

        Sin((ak - 1) * th2)) + tl

```

```

        y = com1 * (Sin(ak * th1) * Cos((ak - 1) * th2) - Cos(ak * th1) *

```

```

    Sin((ak - 1) * th2)
    AntesX(j1) = x
    AntesU(j1) = y
End If
100 Next i
ReDim gridxz(m, 2)
For i = 1 To m
    gridxz(i, 1) = AntesX(m - i + 1) + 1
    gridxz(i, 2) = AntesU(m - i + 1)
Next i
aum = m - 1

End Sub

Sub crout (a(), r())

'.....Algoritmo de Crout para descomposición L-U
Open "c:\dany\matriz.prn" For Output As 1
For i = 1 To UBound(a, 1)
    For j = 1 To UBound(a, 2) - 1
        Print #1, Int(a(i, j) * 100) / 100; " ";
    Next j
    Print #1, Int(a(i, j) * 100) / 100
Next i

Close #1

For j = 2 To UBound(a) + 1
    a(1, j) = a(1, j) / a(1, 1)
Next j

For i = 1 To UBound(a) - 1
    For l11 = i + 1 To UBound(a)
        For k = 1 To i
            sum = sum + a(l11, k) * a(k, i + 1)
        Next k
        a(l11, i + 1) = a(l11, i + 1) - sum
    Next l11
Next i

```

```
sum = 0
Next l11
For l2 = i + 2 To UBound(a) + 1
  For k = 1 To i
    sum = sum + a(i + 1, k) * a(k, l2)
  Next k
  a(i + 1, l2) = (a(i + 1, l2) - sum) / a(i + 1, i + 1)
  sum = 0
Next l2
Next i
```

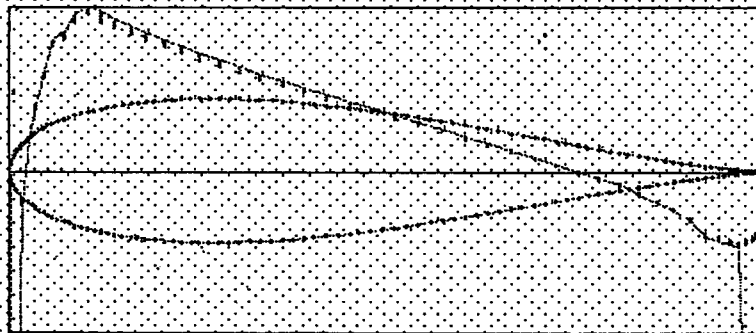
'.....Backward Sustitution

```
r(UBound(a)) = a(UBound(a), UBound(a) + 1)
form1.Print r(UBound(a))
For i = UBound(a) - 1 To 1 Step -1
  For j = UBound(a) To i + 1 Step -1
    sum = sum + a(i, j) * r(j)
  Next j
  r(i) = a(i, UBound(a) + 1) - sum
  sum = 0
Next i
```

8.36485072694341E-02

Coeficientes	
Dif de CL	
-1.93317103722359E-03	
-2.36611495493151E-03	
-2.49030046673714E-03	
-2.55098105082433E-03	
-2.57042779540553E-03	
-2.54790829849724E-03	
-2.48277676404891E-03	

Cl	8.36485072694341E-02
L	-1.41088714339714E-02
cd	-1.176739915394E-04



Cálculo
 — Solución analítica

alfa = 0°
 99 paneles

