

SIMULACION NUMERICA DE LA AERODINAMICA DE UN VEHICULO

Gerardo Franck*, Julián Fredes§, Diego Italiani§, Darío Matijacevich§,
Norberto Nigro†, Jorge D'Elía† y Mario Storti†

*Becario Proyecto FOMEC 1055. Universidad Tecnológica Nacional (UTN)
Facultad Regional Santa Fe (FRSF). Lavaise 610, 3000-Santa Fe, Argentina

§Escuela de Ingeniería Mecánica. Universidad Nacional de Rosario (UNR)
Pelegri 250, 2000-Rosario, Argentina

†Centro Internacional de Métodos Computacionales en Ingeniería (CIMEC)
INTEC; Universidad Nacional del Litoral - CONICET
Güemes 3450, 3000-Santa Fe, Argentina
e-mail: cimec@ceride.gov.ar, web: <http://venus.ceride.gov.ar/cimec>

Key Words: aerodinámica de vehículos, elementos finitos, elementos de borde, cálculo paralelo, cluster Beowulf, mecánica de fluidos

Abstract. *En este trabajo se consideran dos simulaciones numéricas independientes para el flujo del viento alrededor de un vehículo, en dos niveles crecientes de dificultad, primero un cálculo potencial incompresible y, a continuación, otro viscoso incompresible. Como una primera estimación consideramos flujo potencial incompresible, resuelto numéricamente mediante un método de elementos de borde (BEM), de bajo orden, en régimen estacionario y sin desprendimientos, con un barrido paramétrico en la dirección del viento. Luego, pasamos a una descripción más detallada, en donde consideramos flujo viscoso e incompresible, con números de Reynolds relativamente elevados y con posible separación, resuelto numéricamente con un método de elementos finitos (FEM), de igual orden para la presión y la velocidad, con la estabilización propuesta por Tezduyar et al.. Para las simulaciones se utiliza procesamiento distribuido en un “cluster Beowulf”, i.e. una máquina paralela construida con computadoras personales bajo Linux, y han sido escritos en: C++/MPI para el código PETSC-FEM¹ y en F90/HPF/MPI para el código BEM.*

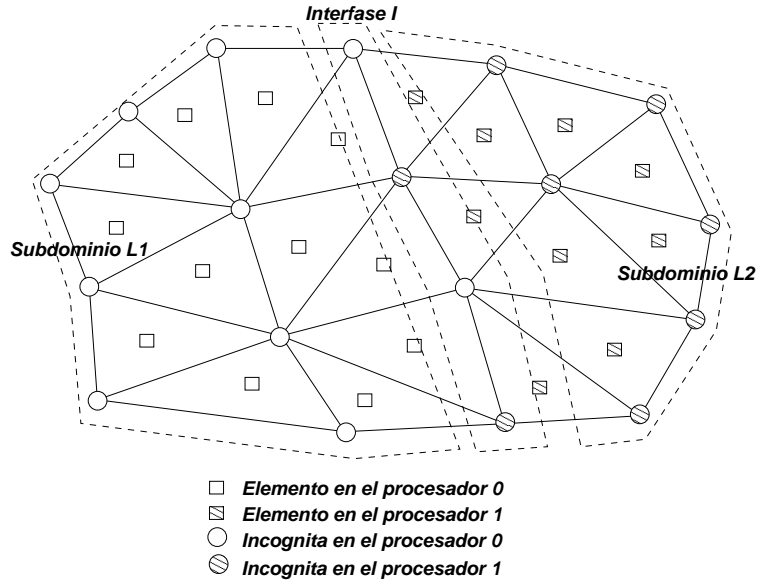


Figura 1: Descomposición de un dominio en subdominios.

1 INTRODUCCION

En el estudio aerodinámico de un vehículo, de calle o de competición, la industria automotriz emplea extensivamente los ensayos en túnel de viento y los ensayos en ruta. Estos procedimientos tienen la ventaja que estamos frente a una observación experimental directa, pero tienen como inconvenientes el hecho de insumir un gran tiempo de desarrollo, un esfuerzo laboral importante y un costo relativo considerable.

Debido a estos inconvenientes y gracias al advenimiento de computadoras digitales cada vez más poderosas, ha crecido el uso de los métodos computacionales para simular, con diferentes niveles de aproximación, lo que sucede en los ensayos. Así, mediante las técnicas de la dinámica de fluidos computacional (CFD), se pretende estimar, por un lado, los valores de fuerzas y momentos que son, usualmente, la información que se obtiene en un túnel de viento y, de esta forma, tener una idea de las fuerzas de resistencia al avance y de sustentación sobre el vehículo. También es posible estimar la posición del eje central de presión el cual incide en su estabilidad direccional, debido a su dinámica lateral (giros en curvas). Además, ofrece la ventaja de entregar otras variables del problema que son difíciles de medir o registrar en ensayos, por ejemplo, líneas de corriente, desprendimiento de vórtices o campos de presiones.

El flujo de aire relativo generado por el vehículo crea un campo de presión y su consecuente campo de velocidades, presentando zonas donde el flujo se acelera y otras donde se frena. Este campo de presión y el gradiente de velocidad resultante dan lugar a fuerzas aerodinámicas experimentadas por el vehículo las cuales dependen, fundamentalmente, de su geometría y de su velocidad de avance relativa al viento. Para todo propósito práctico, el flujo alrededor del vehículo es totalmente turbulento, i.e. $Re > 10^6$. Los efectos de

compresibilidad son insignificantes hasta una velocidad aproximada de 350 Km/h, así que para vehículos de calle el flujo puede ser considerado como prácticamente incompresible. Por otra parte, dada la gran complejidad de la geometría del flujo involucrado, en este trabajo la hemos simplificado omitiendo la presencia de accesorios externos, canalizadores de ventilación y refrigeración, tomas de aire o ruedas. Lo que suele importar a la hora de introducir modificaciones en el diseño, en primera instancia, es el comportamiento relativo antes y después de las modificaciones introducidas.

2 FLUJO VISCOSO INCOMPRESIBLE

2.1 Formulación diferencial

Las ecuaciones diferenciales para el balance de la cantidad de movimiento (ecuaciones de Navier-Stokes) y el balance de masa, para un flujo de un fluido viscoso e incompresible, son

$$\left. \begin{aligned} \rho(\partial_t \mathbf{u} + \mathbf{u} \cdot \nabla \mathbf{u}) - \nabla \cdot \boldsymbol{\sigma} &= \mathbf{0} \\ \nabla \cdot \mathbf{u} &= 0 \end{aligned} \right\} \quad (1)$$

en $\Omega \times (0, T)$, donde Ω es una región tridimensional de flujo, y $(0, T)$ es un cierto intervalo finito de tiempo, $\mathbf{u} = (u_x, u_y, u_z)$ y ρ son la velocidad y la densidad del fluido, respectivamente, mientras que $\boldsymbol{\sigma}$ es el tensor de tensiones $\boldsymbol{\sigma} = -p\mathbf{I} + 2\mu_e\boldsymbol{\epsilon}$, donde p es la presión. Además, $\mu_e = \rho\nu_e$, donde μ_e y ν_e son las viscosidades dinámica y cinemáticas efectivas, respectivamente (que en el caso laminar coincide con la molecular), \mathbf{I} es la matriz identidad y $\boldsymbol{\epsilon} = (\nabla \mathbf{u} + \nabla \mathbf{u}^T)/2$ es el tensor velocidad de deformación. Esta formulación del continuo se completa con adecuadas condiciones iniciales y de contorno.

2.2 Resolución numérica

La resolución numérica de las ecuaciones de Navier-Stokes incompresibles dado por la Ec. 1, la hacemos mediante elementos finitos, con un esquema combinado de derivadas en contracorriente, en la variante SUPG+PSPG (Streamline Upwind Petrov Galerkin + Pressure Stabilized Petrov Galerkin) propuesto por Tezduyar *et. al.*². Este esquema combinado se debe a que las ecuaciones de Navier-Stokes para flujos incompresibles presentan dos dificultades distintivas para su resolución por elementos finitos. En primer lugar, tales ecuaciones se tornan cada vez más dominadas por la advección a medida que el número de Reynolds Re se incrementa. Por otra parte, la condición de incompresibilidad no representa una ecuación de evolución sino, más bien, una restricción sobre las de Navier-Stokes. Esto tiene el inconveniente que solo podemos usar algunas combinaciones de los espacios de interpolación para la velocidad y la presión, a saber, aquellas que satisfacen la denominada condición de Brezzi-Babuska. En la formulación de Tezduyar *et. al.*, la ecuación de advección es estabilizada agregando un cierto término “SUPG” y, similarmente, se incluye otro término “PSPG” para estabilizar la ecuación de incompresibilidad. Una vez que estas ecuaciones “modificadas” son discretizadas en el espacio mediante elementos

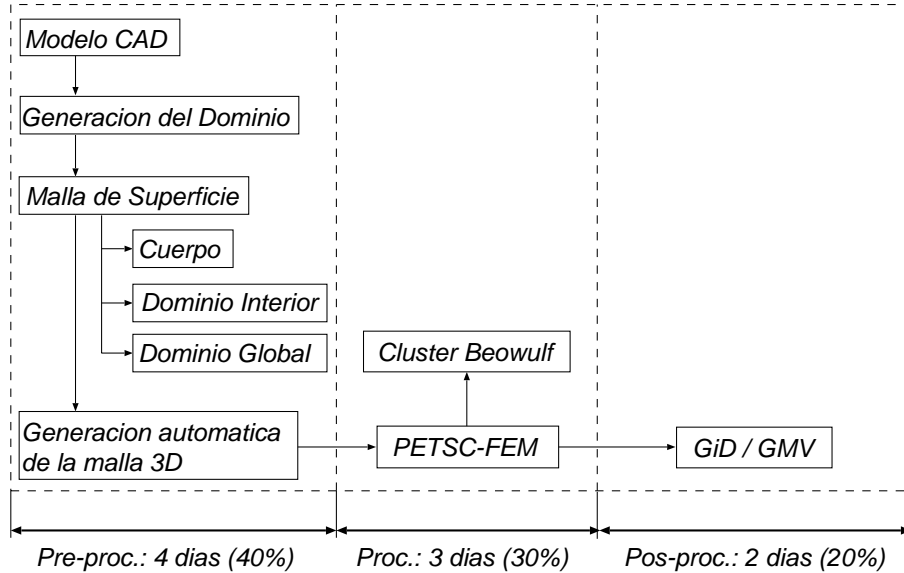


Figura 2: Etapas en el proceso de simulación numérica del flujo viscoso.

finitos, resulta un sistema de Ecuaciones Diferenciales Ordinarias (ODE) en el tiempo t , cuya solución numérica es hallada mediante un esquema en diferencias finitas retro-Euler. A su vez, en cada paso de tiempo t_k , el sistema de ecuaciones no lineales es resuelto iterativamente mediante residuos generalizados (GMRES³), con preconditionamiento a la derecha de Jacobi.

Todo este esquema es implementado en el programa de elementos finitos PETSc-FEM^{1,4}, orientado principalmente a la resolución numérica de problemas en mecánica de fluidos con varios campos acoplados (o multi-física), y está basado en la librería de paso de mensajes MPI⁵ (Message Passing Interface) y en la librería de cálculo paralelo científico PETSC⁶ (Portable Extensible Toolkit for Scientific computations). Esta última nos provee un entorno para programación distribuida, permitiendo diferentes niveles de inserción del programador con el paralelismo, ya sea desarrollando tareas paralelas (e.g. asignación de datos, resolución de sistemas) o bien programando explícitamente las mismas. El código PETSC-FEM es ventajoso en su flexibilidad y su capacidad de paralelizar en forma eficiente todo el proceso de resolución, permitiendo además utilizar diferentes estrategias para la misma. En la Fig. 2 resumimos las etapas involucradas en todo el proceso de simulación del caso viscoso.

2.3 El método de descomposición de dominios

Una vez discretizado espacial y temporalmente el operador del continuo obtenemos un sistema de ecuaciones algebraicas no lineales que debemos resolver en cada paso de tiempo. En general este tipo de problemas tiene la particularidad de requerir una discretización de la malla muy refinada, del orden del millón de elementos o más, lo cual no da posibilidad

de recurrir a métodos directos. Por otro lado, si bien los métodos iterativos han ido evolucionando en los últimos años de una forma importante, es bien conocida la dificultad que existe al resolver sistemas que son no simétricos e indefinidos en signo, tal como suele aparecer en estos problemas de mecánica de fluidos.

Uno de los métodos más referenciados recientemente es el de residuos generalizados (GMRES), pero es conocido que para lograr buena convergencia se necesita incrementar la dimensión del subespacio de Krylov a medida que la malla se va refinando. En los problemas más desafiantes de la actualidad las mallas superan el millón de elementos con lo cual, por lo anterior, coloca al usuario frente a la necesidad de consumir mucha memoria RAM con una tasa de convergencia no muy buena, lo cual redundaría en un mayor tiempo de cómputo. En los últimos años, ha habido un gran avance en técnicas que se basan en la descomposición del dominio en subdominios, con lo cual uno puede confeccionar un método *híbrido* (iterativo y directo), en donde parte del problema se resuelve en forma directa y la parte restante se hace en forma iterativa, con lo cual se espera lograr mejoras en cuanto a la relación memoria/tasa de convergencia.

Imaginemos una malla como, por ejemplo, la de la Fig. (1), particionada en forma tal que un cierto número de elementos y nodos pertenezcan al procesador 0 y los otros al procesador 1. Asumamos que una incógnita está asociada con cada nodo y que no se imponen condiciones de contorno Dirichlet, de forma tal que a cada nodo le corresponde una incógnita.

Dividamos las incógnitas nodales en tres subconjuntos disjuntos L_1 , L_2 e I , tal que los nodos en L_1 no están conectados a aquellos en L_2 (es decir, que no comparten un elemento) y, entonces, los elementos de la matriz de elementos finitos $A_{i,j}$ y $A_{j,i}$, con $i \in L_1$ y $j \in L_2$ son nulos por lo que la matriz la podemos dividir en los bloques

$$\mathbf{A} = \begin{bmatrix} \mathbf{A}_{LL} & \mathbf{A}_{LI} \\ \mathbf{A}_{IL} & \mathbf{A}_{II} \end{bmatrix} \quad \text{donde} \quad \mathbf{A}_{LL} = \begin{bmatrix} \mathbf{A}_{11} & \mathbf{0} \\ \mathbf{0} & \mathbf{A}_{22} \end{bmatrix}; \quad (2)$$

y

$$\mathbf{A}_{IL} = \begin{bmatrix} \mathbf{A}_{I1} & \mathbf{A}_{I2} \end{bmatrix} \quad ; \quad \mathbf{A}_{LI} = \begin{bmatrix} \mathbf{A}_{1I} \\ \mathbf{A}_{2I} \end{bmatrix}. \quad (3)$$

Ahora consideramos el sistema de ecuaciones $\mathbf{Ax} = \mathbf{b}$, el cual lo dividimos como

$$\left. \begin{aligned} \mathbf{A}_{LL}\mathbf{x}_L + \mathbf{A}_{LI}\mathbf{x}_I &= \mathbf{b}_L \\ \mathbf{A}_{IL}\mathbf{x}_L + \mathbf{A}_{II}\mathbf{x}_I &= \mathbf{b}_I \end{aligned} \right\} \quad (4)$$

Eliminemos \mathbf{x}_L a partir de la primera ecuación y reemplazemos en la segunda, de forma de obtener una ecuación para \mathbf{x}_I , es decir,

$$(\mathbf{A}_{II} - \mathbf{A}_{IL}\mathbf{A}_{LL}^{-1}\mathbf{A}_{LI}) \mathbf{x}_I = \mathbf{b}_I - \mathbf{A}_{IL}\mathbf{A}_{LL}^{-1}\mathbf{b}_L; \quad (5)$$

introduciendo la matriz y el vector modificados

$$\begin{aligned}\tilde{\mathbf{A}} &= \mathbf{A}_{II} - \mathbf{A}_{IL}\mathbf{A}_{LL}^{-1}\mathbf{A}_{LI} ; \\ \tilde{\mathbf{b}}_I &= \mathbf{b}_I - \mathbf{A}_{IL}\mathbf{A}_{LL}^{-1}\mathbf{b}_L ;\end{aligned}\tag{6}$$

obtenemos el sistema reducido $\tilde{\mathbf{A}}\mathbf{x}_I = \tilde{\mathbf{b}}_I$. Luego, resolvemos este sistema de ecuaciones reducido mediante un método iterativo tal como, por ejemplo, residuos generalizados GMRES. Para tal método iterativo tenemos que especificar cómo computar el vector miembro derecho modificado $\tilde{\mathbf{b}}$ y, también, cómo hacer el producto matriz-vector $\mathbf{y} = \tilde{\mathbf{A}}\mathbf{x}$. El cálculo del producto matricial involucra los siguientes pasos:

1. Calcular $\mathbf{y} = \mathbf{A}_{II}\mathbf{x}$
2. Calcular $\mathbf{w} = \mathbf{A}_{LI}\mathbf{x}$
3. Resolver $\mathbf{A}_{LL}\mathbf{z} = \mathbf{w}$ para \mathbf{z}
4. Calcular $\mathbf{v} = \mathbf{A}_{IL}\mathbf{z}$
5. Sumar $\mathbf{y} = \mathbf{y} - \mathbf{v}$

involucrando tres productos matriciales con las matrices \mathbf{A}_{II} , \mathbf{A}_{IL} y \mathbf{A}_{LI} y resolver el sistema con \mathbf{A}_{LL} . Como la matriz \mathbf{A}_{LL} no tiene elementos conectando incógnitas que pertenecen a diferentes procesadores, entonces, la solución del sistema puede calcularse en forma paralela muy eficientemente. Esta estrategia de descomposición en subdominios es implementada en el código PETSc-FEM, en donde resolvemos la matriz complemento de Schur en forma iterativa, para los nodos de interfase entre subdominios, y una resolución directa para los nodos interiores de cada subdominio, e.g. ver Lucquin y Pironneau⁷.

2.4 Mallado con un despeje muy reducido del piso

Es bien sabido que la calidad de la malla afecta notablemente a los resultados obtenidos en CFD. Así, en la aerodinámica de vehículos, es necesario generarla con elementos de tamaños acordes a los detalles fluido-dinámicos a observar. Esto conlleva incluir el diseño geométrico con la mayor completitud posible, al menos la que tenga la influencia más relevante sobre la aerodinámica en estudio.

Por otra parte, en un vehículo existen ciertos elementos que hacen al funcionamiento, al confort y a la seguridad que alteran el flujo en forma notable. Su inclusión, o no, en la simulación depende de la potencia de cómputo disponible. Por ejemplo, las ruedas son un problema no muy fácil de resolver, ya que tienen un movimiento relativo respecto al resto de la carrocería y, además, alteran el flujo en todo el vehículo. Otro detalle son las canalizaciones, por las cuales se induce al aire a pasar a través del vano del motor, del radiador o por alguna toma de aire. Alerones, spoilers y otros accesorios también influyen

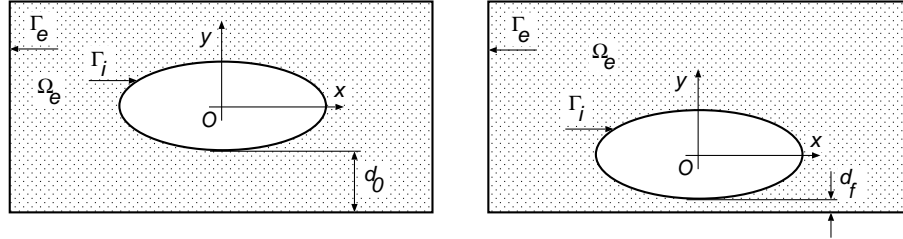


Figura 3: Mallado con un despeje muy reducido del piso. Generación de la malla en la zona del despeje: malla inicial (izq.) y malla deformada (der.).

sobre la aerodinámica del vehículo. Todos estos detalles introducen una dificultad no trivial a la hora de plantear y resolver el problema en forma computacional.

En este trabajo no se pretendió entrar en todos esos posibles detalles, así que el auto fue simulado sin ninguna canalización, sin accesorios ni ruedas. Salvo este último detalle es así como se los ensaya, en general, en los túneles de viento. El vehículo se lo separó del piso por una distancia, o despeje, equivalente al que tendría en el caso de incluir los neumáticos.

El mallado tridimensional es mucho más difícil en cuanto a una generación completamente automática y aceptable, ya que la cantidad de elementos crece notablemente para poder resolver en forma más o menos fidedigna, el mapa de flujo. A este problema se le agregan, por una parte, el tema del refinamiento localizado (sobre la superficie del vehículo y en su estela) y, por otra parte, el inconveniente relacionado con la dificultad de ubicar una cantidad razonable de elementos en la zona del despeje del vehículo, distancia entre el piso del automóvil y el piso del túnel de viento.

Como hemos mencionado, la necesidad de ubicar una gran cantidad de elementos cerca de la superficie del vehículo y, luego, desrefinar hacia el exterior en pos de no excedernos excesivamente en la cantidad total de elementos, trajo aparejado un problema en la generación de la grilla. La cantidad total de elementos no podía exceder los 200.000 tetraedros, por razones de factibilidad de cálculo y, en la zona del despeje, se necesita incluir al menos una decena de elementos.

La estrategia de mallado consistió en ubicar el vehículo en el centro del túnel, generar una malla inicial (o de referencia) y, luego, mover el piso del túnel de tal modo de llevarlo a la distancia de despeje prescripta. La malla inicial la confeccionamos de modo tal que, luego de moverla, resulte una malla deformada con una densidad de elementos aceptable.

El algoritmo de movimiento de la malla se basa en minimizar la máxima distorsión de los elementos de la malla. Como todo problema de minimización debemos definir un funcional apropiado, en este caso, elegimos la distorsión de cada elemento. Para este problema, las incógnitas son los desplazamientos de los nodos de la malla, a excepción de aquellos nodos que pertenecen a los bordes fijos, con $\delta \mathbf{x} = \mathbf{0}$, o los que están sobre un borde con desplazamiento prescripto.

En la Fig. 3 denotamos con \mathbf{d}_0 un despeje inicial, con \mathbf{d}_f el final, en donde el borde

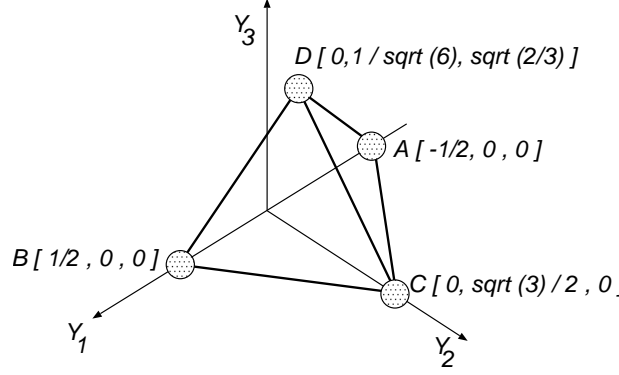


Figura 4: Mallado con un despeje muy reducido del piso: definición del tetraedro master.

exterior Γ_e es desplazado una distancia $\delta \mathbf{x} = \mathbf{d}_0 - \mathbf{d}_f$ hacia arriba. Este desplazamiento del borde exterior modifica la posición de los nodos interiores, pudiendo llegarse a situaciones donde el elemento se “da vuelta” o se deforma excesivamente. La primera patología es inadmisibles, mientras que la segunda da lugar luego a soluciones poco precisas, si no se controla esa distorsión. Nosotros hemos priorizado la distorsión del elemento, buscando mover los nodos de forma tal que la máxima distorsión sea minimizada.

Este problema de optimización consiste en definir un funcional $F(\mathbf{x})$ y hallar un vector incógnita \mathbf{x} que haga mínimo dicho funcional. Proponemos una combinación lineal de dos efectos: un cambio de volumen del elemento respecto a un valor de referencia y otro que nos de una idea de la distorsión del elemento, medida a través de los autovalores del tensor métrico \mathbf{G} , el cual surge del producto $\mathbf{G} = \mathbf{J}^T \mathbf{J}$, donde \mathbf{J} es el jacobiano de la transformación que nos mapea el elemento dado a otro llamado “master”, o ideal, el cual será un triángulo equilátero en 2D o un tetraedro de aristas unitarias en 3D, y cuyos nodos estarán ubicados según se muestra en la Fig. 4.

Al resolver el problema de autovalores para el tensor métrico obtendremos la descomposición $\mathbf{G} = \mathbf{S} \mathbf{\Lambda} \mathbf{S}^{-1}$, donde $\mathbf{\Lambda}$ y \mathbf{S} son las matrices de autovalores (diagonal) y de autovectores, respectivamente, y, a continuación, obtenemos las direcciones principales. El volumen del elemento lo estimamos mediante la traza $V = \text{tr}(\mathbf{G}) = \sum_i \lambda_i$ (suma de los autovalores λ_i), mientras que para su distorsión usamos

$$\Delta = (\lambda_1 - \lambda_2)^2 + (\lambda_2 - \lambda_3)^2 + (\lambda_3 - \lambda_1)^2 . \quad (7)$$

Finalmente, el funcional lo construimos mediante una combinación lineal de ambos, esto es,

$$F = C_V (V - V_r)^2 + C_\Delta \left[\frac{\Delta}{V^{2/3}} \right]^\alpha ; \quad (8)$$

donde C_V , C_Δ , V_r , α son parámetros pre-definidos por el usuario. Los dos primeros miden el peso que le daremos al cambio de volumen y a la distorsión en el funcional, V_r

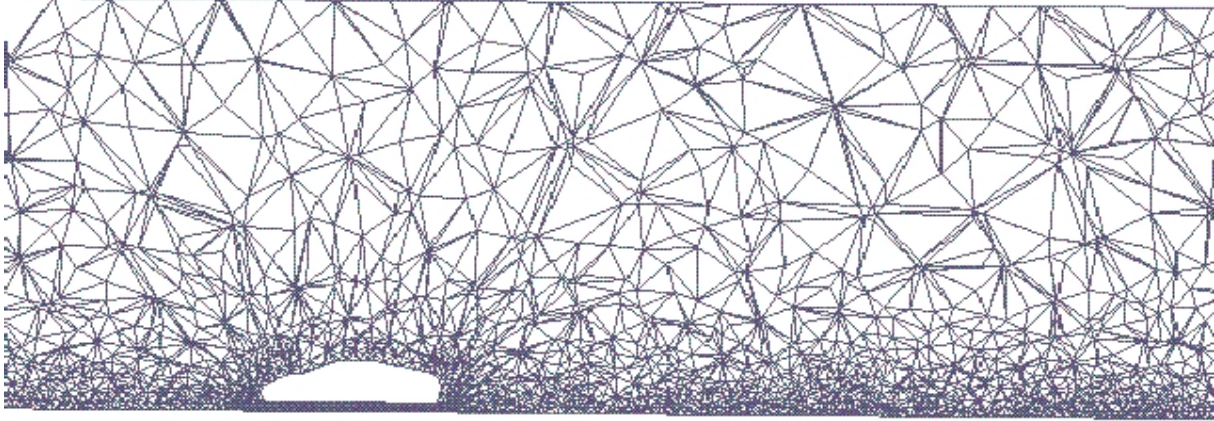


Figura 5: Mallado con un despeje muy reducido del piso: corte según plano simetría vertical.

es un volumen de referencia contra el cual comparar y, finalmente, α es un exponente que representa el tipo de norma usada para medir el funcional. Elegimos $\alpha = 2$ (la usual norma euclídea) y $\alpha \rightarrow \infty$ (norma del supremo) pero, por razones de regularidad, se suele adoptar un valor finito no muy alto. Nosotros hemos tomado $\alpha = 4$. De esta forma obtenemos un sistema de ecuaciones no-lineales que se resuelve mediante una estrategia de Newton-Raphson, iterando sobre el residuo $R(\mathbf{x})$. Si bien uno podría mover el borde exterior de una única vez, es bien sabido que las no-linealidades podrían jugaros una mala pasada, por lo que preferimos avanzar en el tiempo de a poco, usando una decena de pasos para cubrir todo el desplazamiento necesario para el despeje impuesto. En las Figs. 5 y 6, mostramos diferentes vistas de una de las mallas obtenidas con esta estrategia, generando unos 170.000 elementos y 34.000 nodos.

2.5 Mallado con dos subdominios

Los generadores de mallas se valen de algún tipo de ordenamiento de los datos en las fronteras del dominio, que deben ser suministrados por el usuario. En nuestro caso, usamos un sistema CAD para definir un modelo geométrico, algo simplificado, del vehículo. Luego adoptamos dimensiones relativas del túnel de viento “virtual”, esto es, su largo, ancho y altura, resultando entonces un cierto dominio *global* (finito) pre-definido. La primera tarea es una generación de la malla superficial, con cuadriláteros, sobre la superficie del vehículo (e.g. con GiD) y sobre las fronteras del dominio global (e.g. con un sistema CAD).

Por una parte, necesitamos un buen refinamiento *cerca* de la superficie del vehículo y en *toda* una estela hipotética, en donde suceden fenómenos que definen buena parte de los objetivos de la simulación. Pero, por otro lado, no queremos que el número total de elementos generados en forma automática “se dispare”, debido a restricciones de la memoria RAM y/o tiempo de CPU, aún con las bondades del “cluster Beowulf”.

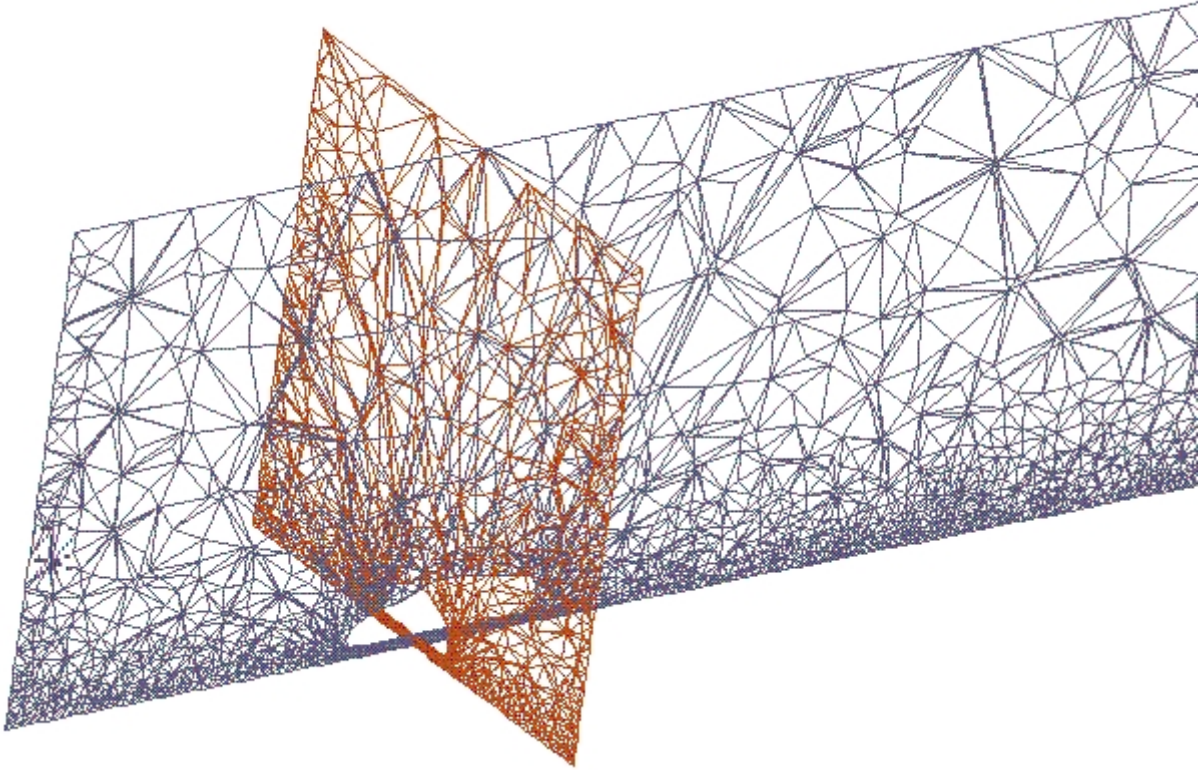


Figura 6: Mallado con un despeje muy reducido del piso: corte según planos de simetrías vertical y transversal.

Entonces, elegimos definir un dominio *interior* auxiliar, ver Fig. 7 (izq.), el cual comprende una zona inmediatamente próxima al vehículo y toda una estela hipotética, el cual nos servirá para disponer de un mejor control en el grado de refinamiento y desrefinamiento. Las mallas de superficie de todas las fronteras son generadas inicialmente con cuadriláteros, con un gran refinamiento sobre el dominio interior. A continuación, éstas son convertidas a triángulos. En la Fig. 7 (der.), vemos una vista de los cuadrángulos obtenidos sobre la superficie del vehículo. Por último, generamos una malla de tetraedros mediante el método de Delaunay-Voronoi junto con un control semi-automático en el grado de refinamiento y desrefinamiento. La malla de elementos finitos resultante tiene un costo computacional de orden n , donde n es el número de puntos que define los dominios interior y global, e.g. ver Calvo e Idelsohn⁸ y, además, cumple mucho mejor con las especificaciones del usuario en cuanto a las zonas refinadas y desrefinadas.

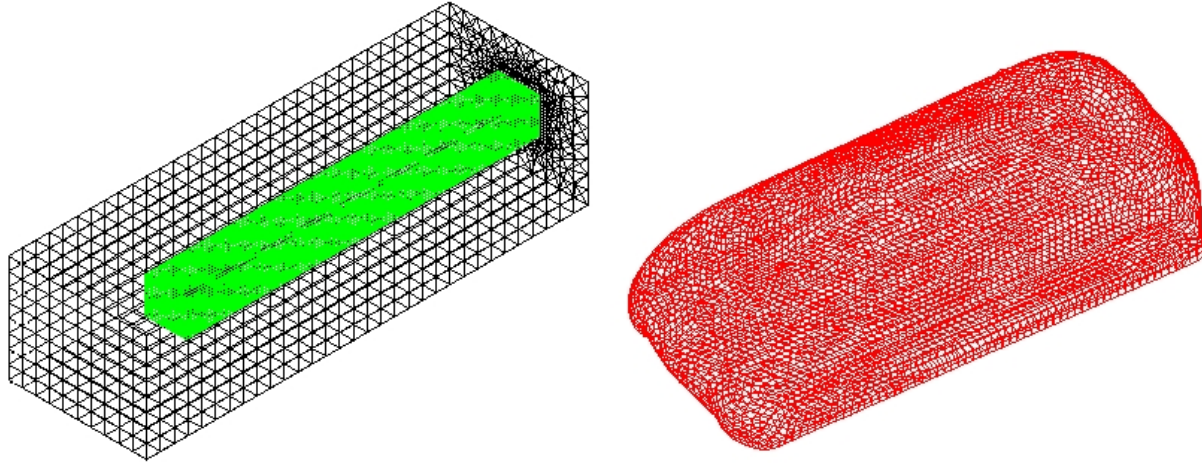


Figura 7: Mallado con dos subdominios: Izq.: malla de cuadrángulos en las fronteras del dominio global e interior, ésta última comprende una zona cercana al vehículo y una estela hipotética. Der.: vista de los triángulos resultantes sobre la superficie del vehículo.

3 FLUJO POTENCIAL INCOMPRESIBLE

3.1 Descripción del código de paneles en paralelo

El código de paneles para el cálculo potencial ha sido re-escrito en Fortran 90/95 (F90) el cual, al momento presente, incluye diversos módulos (e.g. wave-drag y seakeeping para hidrodinámica potencial de barcos o flujo reptante para sistemas coloidales), en donde se implementan técnicas casi Orientadas a Objetos (OO), esto es, dentro de las restricciones actuales del F90 para tal paradigma (e.g. ver Norton,⁹ Decyk,^{10,11} Meissner¹², Singh¹³), con High Performance Fortran (HPF) para su opcional procesamiento distribuido, e.g. en un “cluster Beowulf”, es decir, una máquina paralela construída con computadoras personales (PC) bajo Linux¹⁴. Entre otras facetas de este código, todavía en fase de desarrollo y depuración, mencionamos:

1. Clases: implementadas mediante la construcción *module*, con atributos de acceso *private/public*. F90 permite la creación de Tipos Abstractos de Datos (TAD) definidos por el usuario, para encapsular tanto datos (con estructura o no) como procedimientos (*contains*), muy relacionados entre sí, en única unidad funcional;
2. Polimorfismo: mediante el uso de *interface*, permite una programación genérica, sobrecarga de operadores y construcción de interfases explícitas;
3. Herencia: es la capacidad de re-usar y/o compartir datos/procedimientos de clases pre-existentes, para así construir una jerarquía de clases, y suele distinguirse:
 - (a) Herencia por acceso permitido (granting access): es el método más simple, en donde los objetos de una clase acceden a objetos de otra clase base mediante el uso de la instrucción *use* y el atributo selectivo *public*;

- (b) Herencia por composición (“has-a”): usa otras clases para construir clases mayores y más elaboradas. Inmediato de implementar en F90;
 - (c) Herencia por sub-tipo (“is-a”): innata en C++ y en Java pero en F90 tiene que ser *emulada* mediante composición. En parte, por esto, se dice que F90 es *parcialmente* OO;
4. Sintaxis y librería matricial: muy similar al estilo usado en Octave/Matlab:
 5. Paralelismo: lo hacemos con la norma High Performance Fortran¹⁵ (HPF), una extensión del F90 apto para su uso en “cluster Beowulf”. En cambio descartamos, por ahora, el uso de la norma *Open Machine Paralell*¹⁶ (OMP) la cual, si bien permite hacer paralelismo dentro, por ejemplo, de una Pentium dual, obliga usar librerías de pasos de mensajes para la comunicación entre nodos, si estos estuvieran contruídos con Pentium duales, librerías tales como la Parallel Virtual Machine¹⁷ (PVM) o la Message Passing Interface⁵ (MPI);
 6. Ingreso de datos: se emplea extensamente el *namelist*, estándar en F90, por su comodidad de uso y seguridad de ingreso.

A modo de observaciones diremos que:

1. La ventaja primordial de un esquema tipo OO está en un encapsulamiento de datos y procedimientos. La información queda descentralizada y se reduce drásticamente los típicos efectos colaterales;
2. En una clase (*module*) agrupamos los datos (atributos) y los procedimientos (métodos) como una única unidad funcional, no siendo necesario pasar argumentos dentro de la clase. La herencia por composición (“has-a”) es la más fácil de implementar en F90, mientras que la herencia por sub-tipo (“is-a”) tiene que se emulada.
3. El agrupamiento de los datos y procedimientos en unidades funcionales (clases), da mayor independencia relativa en un programa multipropósito, facilitando tanto la depuración de errores como su actualización;
4. La próxima norma fortran 2002¹⁸ prevee que deberá ser completamente orientado a objetos incluyendo, por ejemplo, polimorfismo dinámico, punteros a procedimientos, selección de tipo dinámica (la ejecución dependerá del tipo actual que tenga el objeto polimórfico), con lo cual, entre otros aspectos, permitirá una implementación no-emulada de la herencia por sub-tipo.

3.2 Ecuación integral de borde

El problema de flujo potencial incompresible y estacionario lo formulamos con la ecuación integral de borde, e.g. ver Morino¹⁹,

$$\frac{1}{2}\phi(\mathbf{x}) + \frac{1}{4\pi} \int_{\Gamma} d\Gamma G_{,n}(\mathbf{x}, \boldsymbol{\xi})\phi(\boldsymbol{\xi}) = Q(\mathbf{x}) ; \quad (9)$$

para $\mathbf{x} \in \Gamma$, donde Γ es el borde mojado del cuerpo, $\mathbf{x}, \boldsymbol{\xi}$ son los puntos de observación y fuente, respectivamente,

$$Q(\mathbf{x}) = \frac{1}{4\pi} \int_{\Gamma} d\Gamma G(\mathbf{x}, \boldsymbol{\xi})\sigma(\boldsymbol{\xi}) ; \quad (10)$$

es el término fuente, $\phi(\mathbf{x})$, es el potencial de perturbación, mientras que σ es el flujo de transpiración dados por

$$\sigma = (\mathbf{U}, \mathbf{n}) ; \quad (11)$$

donde $\mathbf{n} = (n_x, n_y, n_z)$ es el versor normal en la superficie mojada del cuerpo, mientras que la normal generalizada es

$$\begin{aligned} (n_1, n_2, n_3) &= \mathbf{n} \\ (n_4, n_5, n_6) &= (\mathbf{r} - \mathbf{r}_0) \times \mathbf{n} \end{aligned} \quad (12)$$

con $\mathbf{r}(x, y, z)$ y \mathbf{r}_0 es algún punto conveniente de reducción.

Para resolver numéricamente la Ec. 9 se emplea un método de elementos de borde de bajo orden, en el cual se la impone mediante una técnica por colocación en los centroides de los elementos de borde (o paneles). Se obtiene así el sistema de ecuaciones algebraicas lineales $\mathbf{A}\boldsymbol{\phi} = \mathbf{C}\boldsymbol{\sigma} = \mathbf{b}$, con n incógnitas, donde la matriz del sistema \mathbf{A} es cuadrada y no singular, con

$$A_{ij} = \frac{1}{4\pi} \int_{\Gamma_j} d\Gamma G_{,n}(\mathbf{x}_i, \boldsymbol{\xi}) ; \quad (13)$$

y

$$C_{ij} = \frac{1}{4\pi} \int_{\Gamma_j} d\Gamma G(\mathbf{x}_i, \boldsymbol{\xi}) ; \quad (14)$$

donde \mathbf{A}, \mathbf{C} son las matrices de influencia dipolar y monopolar, respectivamente, $\boldsymbol{\phi}$ es el vector potencial de perturbación, y $\boldsymbol{\sigma}$ es el vector de flujos. La matriz monopolar \mathbf{C} incluye la función de Green G , mientras que la matriz dipolar \mathbf{A} contiene su derivada normal $G_{,n}$. La matriz del sistema \mathbf{A} es la misma para los N estadios de velocidad, y lo único que cambia el vector fuente \mathbf{b} , lo cual permite, eventualmente, una eficiente solución iterativa por métodos tipo gradientes conjugados, e.g. ver Nachtigal *et.al*²⁰, Vavasis²¹. La función de Green elegida es mediante fuentes de Rankine, la cual la calculamos mediante una integración analítica aunque, si se desea, se prevee una integración numérica.

3.3 Influencia del piso por simetría especular

Para evitar generar una malla BEM para el piso, lo cual significaría un mayor costo de cómputo, usamos el método de imágenes, es decir, consideramos la imagen especular del cuerpo con respecto al plano de reflexión $z = 0$ (en donde está el piso) y planteamos la interacción entre el cuerpo real y el virtual, dando lugar a la descomposición matricial

$$\begin{bmatrix} \mathbf{A}_{11} & \mathbf{A}_{12} \\ \mathbf{A}_{21} & \mathbf{A}_{22} \end{bmatrix} \begin{bmatrix} \boldsymbol{\phi}_1 \\ \boldsymbol{\phi}_2 \end{bmatrix} = \begin{bmatrix} \mathbf{C}_{11} & \mathbf{C}_{12} \\ \mathbf{C}_{21} & \mathbf{C}_{22} \end{bmatrix} \begin{bmatrix} \boldsymbol{\sigma}_1 \\ \boldsymbol{\sigma}_2 \end{bmatrix} ; \quad (15)$$

donde \mathbf{X}_{11} son las matrices de auto-influencia en los paneles sobre el cuerpo real (dipolar: $\mathbf{X} \equiv \mathbf{A}$ y monopolar: $\mathbf{X} \equiv \mathbf{C}$), mientras que \mathbf{X}_{12} y \mathbf{X}_{21} son las matrices de interacción entre el cuerpo real y el virtual. Además, por simetría, los potenciales y flujos sobre el cuerpo real y virtual son iguales entre sí, esto es, $\boldsymbol{\phi}_1 = \boldsymbol{\phi}_2$ y $\boldsymbol{\sigma}_1 = \boldsymbol{\sigma}_2$, por lo que es suficiente plantear y resolver la primera fila de la Ec. 15, esto es, el sistema reducido $\tilde{\mathbf{A}}\boldsymbol{\phi}_1 = \tilde{\mathbf{C}}\boldsymbol{\sigma}_1$, donde $\tilde{\mathbf{A}} = \mathbf{A}_{11} + \mathbf{A}_{12}$ y $\tilde{\mathbf{C}} = \mathbf{C}_{11} + \mathbf{C}_{12}$.

4 EJEMPLOS NUMERICOS

4.1 Flujo viscoso incompresible

La simulación tridimensional tuvo como objetivo parcial, evaluar la factibilidad de resolver problemas de gran tamaño, construyendo además herramientas, tales como: (i) un buen pre-procesamiento de datos a partir de un diseño geométrico del vehículo, e.g. ver Calvo e Idelsohn⁸, (ii) alguna estrategia para poder obtener soluciones en el menor tiempo posible y, (iii) una buena interfase de visualización de los resultados detallados (e.g. mapa de presiones).

Una de las mallas empleadas en las simulaciones tuvo unos 170.000 elementos en el dominio de flujo Ω , pero solo del orden de 1.600 sobre la superficie del vehículo, lo cual es un tanto insuficiente, ya que deberíamos tener al menos unos 5.000 elementos, como para poder resolver mejor los detalles más importantes sobre el vehículo. Incluso es deseable tener una capa de varios elementos de espesor en la zona próxima a la superficie superior del auto, para así resolver mejor la zona de desprendimiento.

En la Fig. 7 se muestran vistas de una de las mallas empleada sobre la superficie del vehículo donde, en colores, se pinta la distribución de presión en un instante de tiempo dado.

En la Fig. 8 se muestra las isóbaras de presión sobre la superficie del vehículo. Se observa una sobrepresión en la trompa del vehículo, una depresión en la zona del capot y una recuperación en la zona trasera del mismo. En la Fig. 9 se muestra la distribución de presiones en la estela, la cual es asimétrica debido a la propia inestabilidad lateral del flujo.

En la Fig. (10) se muestra líneas de corriente alrededor del vehículo.

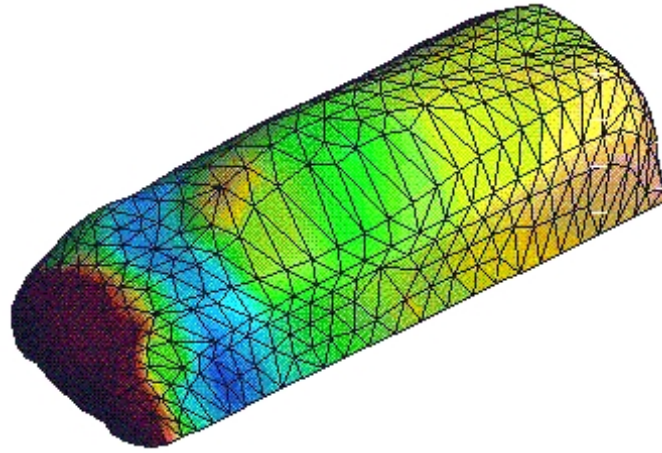


Figura 8: Mapa de presiones sobre la superficie del vehículo.

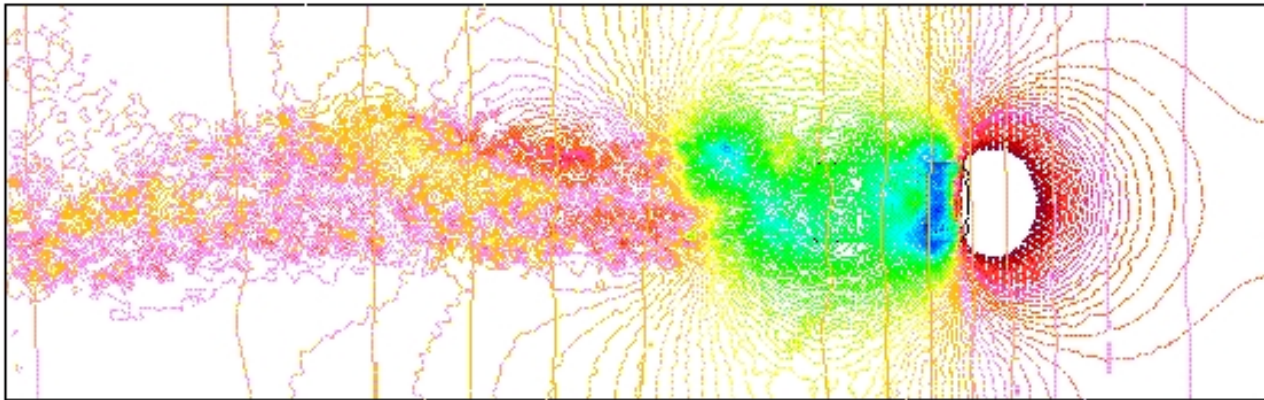


Figura 9: Presión en la estela.

4.2 Flujo potencial incompresible

Elegimos un sistema cartesiano rectangular xyz , donde el plano del suelo está en $z = 0$, con el eje y paralelo al eje axi-simétrico del vehículo (apuntando hacia atrás), el eje x frontal al mismo y el eje z positivo hacia arriba (terna derecha), ver Fig. 11. Como flujo estacionario imponemos una viento uniforme de velocidad no perturbada corriente arriba U unitaria.

Efectuamos un cálculo paramétrico, con un barrido en la dirección del viento α , con respecto al eje axi-simétrico del vehículo y , abarcando desde $-\pi$ hasta $+\pi$ radianes, con $N = 40$ estadios, o puntos, de cálculo. La marcha de cálculo comprende:

1. calculamos las matrices de influencia dipolar \mathbf{A} y monopolar \mathbf{C} para el sistema constituido por el auto y su imagen especular con respecto del suelo (ubicado este

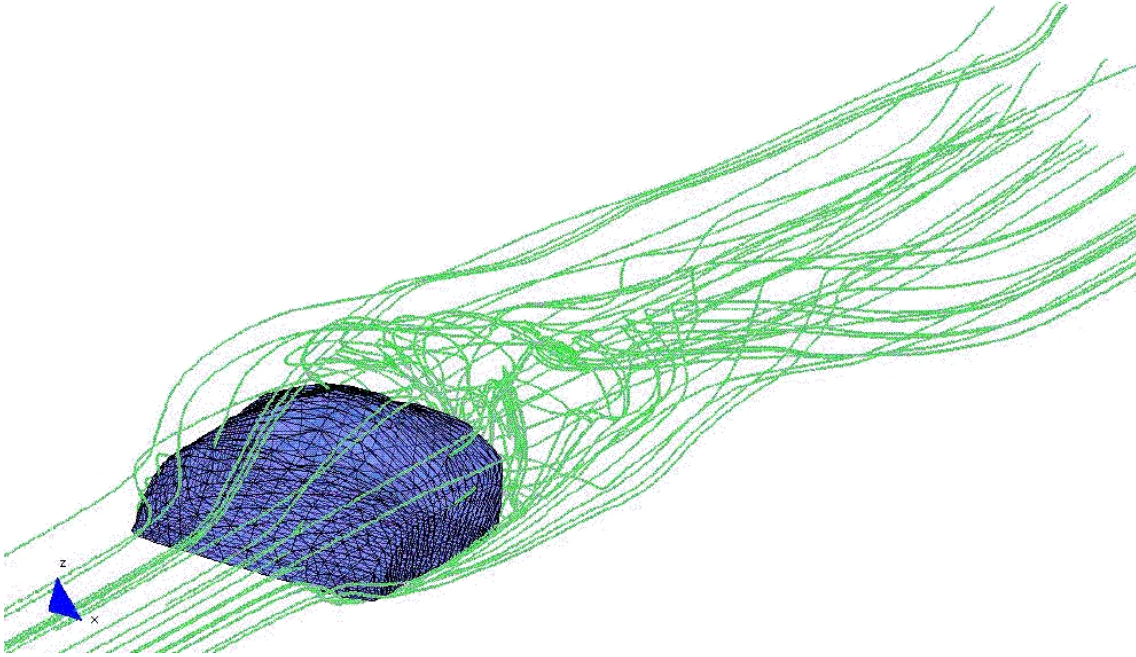


Figura 10: Líneas de corriente alrededor del vehículo.

último en $z = 0$);

2. efectuamos un barrido en la dirección horizontal del viento α desde $\alpha_{min} = -\pi/2$ hasta $\alpha_{max} = +\pi/2$ para $N = 40$ estadios;
3. para cada dirección del viento α , calculamos el flujo σ incidente sobre cada panel mediante el producto escalar $\sigma = (\mathbf{U}, \mathbf{n})$, donde \mathbf{U} es la velocidad del viento y \mathbf{n} es la normal (exterior) del panel;
4. determinamos el potencial de perturbación ϕ , en los centroides de los paneles, mediante una resolución iterativa por gradientes conjugados simultáneos para los N vectores fuente (e.g. ver D'Elía *et. al*²²);
5. luego se calculan las fuerzas \mathbf{F} de presión de cada panel y su momento \mathbf{T} con respecto al origen de coordenadas $O(x, y, z)$.

Una de las mallas de elementos de borde utilizadas en las simulaciones numéricas, se muestra en la Fig. 11, la cual tiene unos 11.000 paneles (incógnitas) y unos 5.100 nodos, lo cual involucra unos 850 MBytes de memoria RAM, ya que la matriz del sistema de ecuaciones a resolver es densa y no simétrica.

En la Fig. 12 (izq.) se muestran las curvas de la fuerza \mathbf{F}_p y del torque \mathbf{T}_p de presión en función de la dirección del viento α con respecto al eje axi-simétrico y del vehículo, mientras que a la derecha se muestra la posición del centro de presión en el plano horizontal

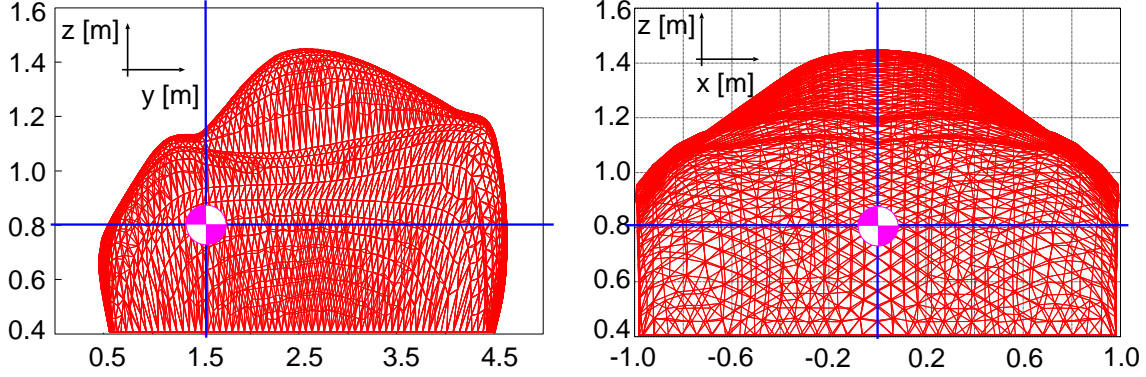


Figura 11: Posición del centro de presión para viento de frente $\alpha = 0$: vista en el plano lateral yz (izq.) ; vista en el plano frontal xz (der.).

xy (paralelo al piso del vehículo) y en el plano transversal zx . Estas curvas fueron obtenidas para $N = 40$ estadios en la dirección del viento en el plano horizontal.

Respecto de los resultados obtenidos, se observa que la posición del centro de presión para viento de frente está, estimativamente, en $y = 1.538$ [m] y $z = 0.811$ [m], desplazándose cuando la dirección del viento cambia, ver Fig. 12 (derecha, arriba y abajo).

Para el cálculo del eje central de las fuerzas de presión en el vehículo, las cuales constituyen un sistema tridimensional de fuerzas aplicadas, primero determinamos un punto de paso \mathbf{O}^* con el “cociente vectorial”

$$(\mathbf{O}^* - \mathbf{O}) = \frac{\mathbf{M}_\mathbf{O}}{\mathbf{R}} \equiv \frac{\mathbf{R} \times \mathbf{M}_\mathbf{O}}{R^2}; \quad (16)$$

donde \mathbf{O} es un punto de reducción de fuerzas y momentos, $\mathbf{O}^* = (x^*, y^*, z^*)$ es un punto de paso del eje central, $\mathbf{M}_\mathbf{O}$ es el momento de reducción, mientras que \mathbf{R} es la resultante de las fuerzas aplicadas (invariante vectorial). Como es conocido, el eje central es una recta en el espacio cuyos puntos dan centros de reducción, tales que el momento del sistema resulta paralelo a la resultante, lo cual significa, a su vez, que sea el mínimo posible. Su ecuación segmentaria estará dada por

$$\frac{x - x^*}{R_x} = \frac{y - y^*}{R_y} = \frac{z - z^*}{R_z}. \quad (17)$$

En base a la Ec. 16 se trazaron las curvas de la Fig. 12. Como el eje central de un sistema de vectores es paralelo a la resultante y ésta, en la suposición de flujo potencial, es una fuerza perpendicular al plano de simetría, se concluye que el eje central de las fuerzas de presión en un vehículo en ruta es una recta perpendicular al piso, pasando por el punto mostrado en la Fig. 12 (izq.).

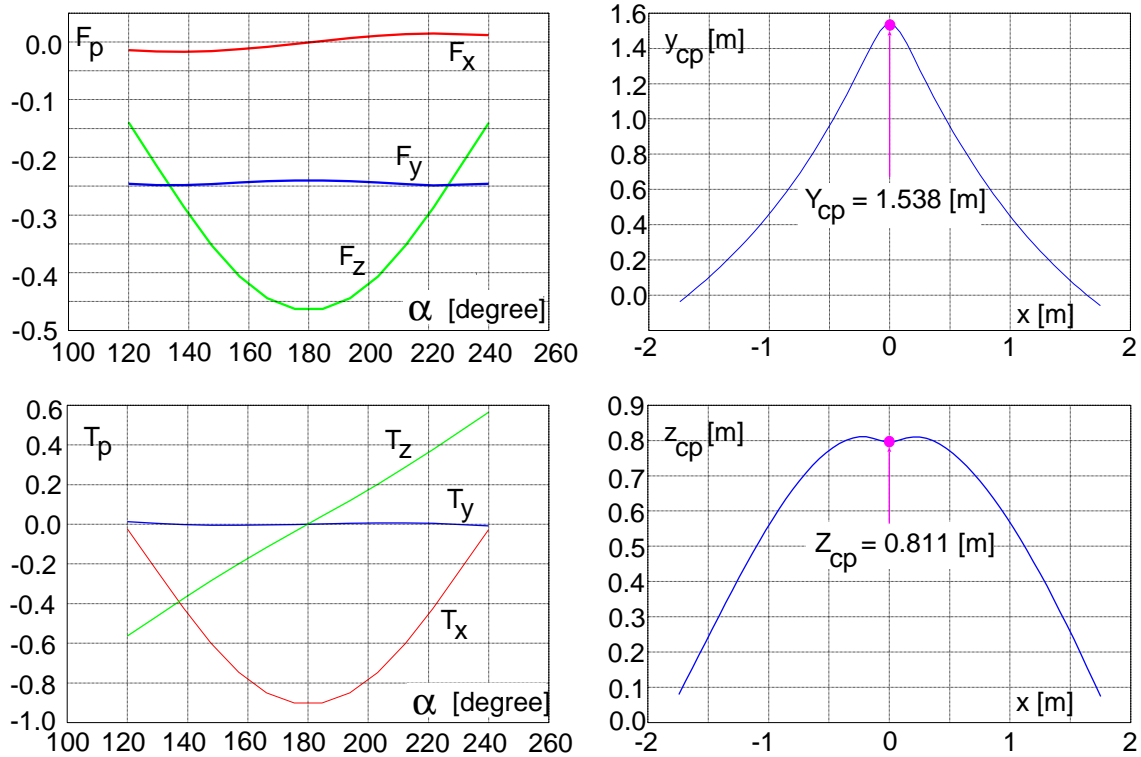


Figura 12: Fuerza (izq.-arriba) y cupla (izq.-abajo) de presión en función de la dirección del viento $\alpha = 0$. Posición del centro de presión en el plano yx (der.-arriba) y en el plano zx (der.-abajo).

5 CONCLUSIONES

Hemos considerado dos simulaciones numéricas independientes para el flujo del viento alrededor de un vehículo en ruta, en dos niveles crecientes de dificultad: primero un cálculo potencial incompresible y luego un cálculo viscoso turbulento, donde:

(i) el cálculo potencial es relativamente mucho más rápido de hacer, tanto en pre/pos y procesamiento propiamente dicho. Esto es muy útil para diseño preliminar y/o para cálculos paramétricos, donde un extensivo conjunto de casos de ensayo deben resolverse a efectos de estudios comparativos. La posición del eje central de presión predicha por el flujo potencial resulta correctamente estimada. Empero, el nivel de detalle del flujo, como es ampliamente conocido, es bastante menor ya que no percibe ni estela, ni fuerza de arrastre ni desprendimientos (a menos que se introduzcan modelos paliativos de simulación de la estela). Empero, una utilidad adicional del modelo potencial es que su campo de presión sería un buen dato inicial para un cálculo para las ecuaciones de Navier Stokes;

(ii) el cálculo viscoso, en contraposición, exige mayor esfuerzo de cálculo, tanto en memoria RAM como en tiempo de CPU, pero nos entrega un mayor nivel de detalle en la descripción de flujo, e.g. los vórtices desprendidos y la fuerza de arrastre, dos magnitudes que un cálculo potencial estándar no puede ofrecer.

Agradecimientos Este estudio fue producto del apoyo que el CIMEC ha ofrecido al proyecto “Proyección de un Automóvil Deportivo” para el grado de Ingeniero Mecánico, en la Escuela de Ingeniería Mecánica de la Universidad Nacional de Rosario. Este trabajo fue desarrollado gracias a subsidios de CONICET, ANPCyT y UNL a través de los proyectos CONICET-PIP-198/98 *Germen-CFD*, SECyT-FONCyT-PICT-6973 *PROA*, SECyT-PID-99/74 *FLAGS* y CAI+D-UNL-PIP-02552-2000. Se ha hecho uso intensivo de *Software Libre* como las librerías MPI, *SO GNU/Linux*, *Octave*, *Xfig*, *Tgif*, *GMV* y muchos otros, así como del procesador gráfico interactivo *GiD*²³. Néstor Calvo ha participado gentilmente en discusiones acerca de la generación de mallas tridimensionales con refinamiento y desrefinamiento espacial selectivo.

REFERENCIAS

- [1] PETSc-FEM: A general purpose, parallel, multi-physics FEM program. GNU General Public License (GPL), <http://minerva.arcrude.edu.ar/petscfem>.
- [2] T. Tezduyar, S. Mittal, S. Ray, and R. Shih. Incompressible flow computations with stabilized bilinear and linear equal order interpolation velocity-pressure elements. *Comp. Meth. Applied Mechanics and Engineering*, **95**(95), 221–242 (1992).
- [3] Y. Saad and M. Schultz. Generalized minimum residual method (gmres). *SIAM J. Sci. Statist. Comput.*, **7**, 856–869 (1986).
- [4] V.E. Sonzogni, A. Yommi, N.M. Nigro, and M. Storti. Cfd finite element parallel computations on a beowulf cluster. In *European Congress on Computational Methods in Applied Sciences and Engineering, ECCOMAS 2000*, (11-14 September 2000).
- [5] Message passing interface (MPI). <http://www.mpi-forum.org/docs/docs.html>.
- [6] S. Balay, W. Gropp, L.C. McInnes, and B. Smith. *Petsc 2.0 users manual*. Technical Report UC-405, Argonne National Laboratory. Mathematics and Computer Science, (1997).
- [7] B. Lucquin and O. Pironneau. *Introduction to Scientific Computing*. John Wiley and Sons, (1998).
- [8] N.A. Calvo and S.R. Idelsohn. Generación de mallas tridimensionales en tiempo lineal. In *MECOM 2002: 1st South-American Congress on Computational Mechanics, III Brazilian Congress on Computational Mechanics, VII Argentine Congress on Computational Mechanics*, Santa Fe-Paraná, Argentina, (October 28-31 2002).
- [9] C.D. Norton. *Object Oriented Programming Paradigms in Scientific Computing*. PhD thesis, Rensselaer Polytechnic Institute, (1996).
- [10] V.K. Decyk. How to express c++ concepts in fortran 90. *Scientific Programming*, **6**(4), 363–390 (1997). IOS Press.
- [11] V.K. Decyk, Norton C.D., and B.K. Szymanski. How to support inheritance and run-time polymorphism in fortran 90. *Computer Physics Communications*, **115**, 9–17 (1998).
- [12] L. P. Meissner. Fortran 90 and 95. array and pointer techniques. objects, data structures and algorithms. Technical report, Computer Science Department. Univesity of

San Francisco, (1998).

- [13] M. Singh and J.E. Akin. Object-oriented fortran 90 p-adaptive finite element method. In B.H.V. Topping, editor, *Developments in Engineering Computational Technology*, pages 141–149. Civil-Comp Press, Edinburgh, (2000).
- [14] The Linux Documentation Project, <http://sunsite.unc.edu/mdw/linux.html>.
- [15] C.H. Koebel, D.B. Loveman, G.L. Steele, and M.E. Zosel. *The High Performance Fortran Handbook*. The MIT Press, (1997).
- [16] OpenMP Architecture Review Board. Openmp fortran application program interface, v2.0. Technical report, OpenMP Architecture Review Board, (1997-2000).
- [17] Parallel virtual machine (PVM). <http://www.epm.ornl.gov/pvm/pvmhome.html>.
- [18] J3 Fortran 200x, (August 2002). <http://www.j3-fortran.org/>.
- [19] Morino L. *Computational Methods in Potential Aerodynamics*. Springer-Verlag, (1985).
- [20] N.M. Nachtigal, S.C. Reddy, and L.N. Trefethen. How fast are non-symmetric matrix iterations ? *SIAM J. Matrix Anal. Appl.*, **13**(3), 778–795 (1992).
- [21] S.A. Vavasis. Preconditioning for boundary integral equations. *SIAM J. Matrix Anal. Appl.*, **13**(3), 905–925 (1992).
- [22] D’Elía J., Storti M., and Idelsohn S. Iterative solution of panel discretizations for potential flows. the modal/multipolar preconditioning. *Int. J. Num. Meth. Fluids*, **32**(1), 1–27 (2000).
- [23] GiD. *International Center for Numerical Methods in Engineering (CIMNE, Barcelona)*. Universidad Politécnic de Catalunya, (2000).