

HERRAMIENTAS DE SOFTWARE CON LICENCIA PÚBLICA GENERAL PARA EL MODELADO POR ELEMENTOS FINITOS

GENERAL PUBLIC LICENSE SOFTWARE TOOLS TO FINITE ELEMENT MODELING

CARLOS GALEANO

Profesor Asistente, MSc en Mecánica Aplicada. Universidad Nacional de Colombia, chgaleanou@unal.edu.co

JUAN MANTILLA

Profesor Asistente, MSc en Ingeniería Mecánica. Universidad Nacional de Colombia

CARLOS DUQUE

Profesor Asistente, MSc en Ingeniería Mecánica, Universidad Nacional de Colombia

MANUEL MEJÍA

Estudiante de Ingeniería Mecánica Universidad Nacional de Colombia

Recibido para revisar diciembre 07 de 2006, aceptado abril 20 de 2007 versión final abril 24 de 2007

RESUMEN: En el presente artículo se presentan los resultados de una revisión realizada a varios programas con licencias GPL para el modelamiento de problemas con elementos finitos. Este estudio permitió realizar la caracterización de estos programas teniendo en cuenta criterios como el tipo de análisis permitido, la compatibilidad con programas CAD, el tipo de soporte ofrecido, la actualización de las versiones, el tipo de interfaz de usuario implementado y la curva de aprendizaje para nuevos usuarios, entre otros. Posteriormente se estudiaron tres casos diferentes, los cuales fueron modelados con elementos finitos empleando distintos programas GPL. En la parte final del documento se comparan los resultados obtenidos con software libre, los arrojados por un programa comercial (Ansys), y los modelos analíticos derivados de la solución de la ecuación diferencial que rige el fenómeno particular, obteniendo una convergencia aceptable entre todos los resultados.

PALABRAS CLAVE: Licencia Pública General, Elementos Finitos, Modelamiento por elementos finitos, Análisis por elementos finitos, Software Libre (free software).

ABSTRACT: This work presents the results of a review made to several programs with GPL licenses for finite elements modeling. This study allowed to make the characterization of these programs having into account criteria like the type of allowed analysis, the compatibility with CAD programs, the type of offered support, the update of the versions, the type of implemented user interface and the learning curve for new users, among others. Later three different cases are studied, which were modeled with finite elements using different GPL programs. In the final part of the document the results obtained with free software are compared, with a commercial program (Ansys), and with the analytical models derived from the solution of the differential equations that governs the particular phenomenon, obtaining an acceptable convergence between all of the results.

KEYWORDS: General Public License, Finite Elements, Finite Elements Modeling, Finite Element Analysis, Free Software.

1. INTRODUCCIÓN

La licencia *GPL* (*General Public License* o licencia pública general) fue creada a mediados de los años 80 como un mecanismo para

impulsar el uso de software libre. La licencia *GPL* obliga al licenciatarario a hacer públicos los códigos fuente del software desarrollado, con el

fin de permitir a los usuarios del mismo poder hacer modificaciones, adaptaciones o mejoras al programa. Por otro lado el usuario tiene como obligación licenciar bajo el esquema *GPL* cualquier producto resultante de la modificación, adaptación o mejora a un software con licencia *GPL*. Esta filosofía de licenciamiento ha permitido el crecimiento acelerado del número de usuarios de estos programas y en forma paralela, casi como una consecuencia natural, el permanente desarrollo de estos paquetes *GPL*.

Dentro de este movimiento de desarrollo de nuevos programas con licencias públicas *GPL*, hacen parte también programas especializados como lo son los programas para el modelamiento por elementos finitos. Un software de elementos finitos se estructura normalmente en tres módulos fundamentales: el módulo de pre-procesamiento, el módulo de procesamiento y el módulo de post-procesamiento. El primer módulo se encarga de permitir al usuario el ingreso, de forma simple, de la información relacionada con el modelo a desarrollar. Este primer módulo comúnmente se compone de un paquete *CAD* (Dibujo) sencillo, de un algoritmo de mallado de geometrías y de librerías con modelos de comportamiento de materiales (elásticos, plásticos, hiperelásticos, etc.). También posee diversas herramientas para definir las condiciones de borde del problema y las condiciones iniciales, si se trata de problemas para análisis en estado transitorio.

El módulo de procesamiento transforma toda la información ingresada en la etapa previa de modelado en forma de un sistema de ecuaciones lineales, para luego darle solución mediante poderosos solucionadores o *solvers*, que permiten encontrar los valores de las variables buscadas (desplazamiento, velocidad, temperatura, presión, etc.) en cada uno de los nodos que hacen parte de la malla creada.

El último de los módulos normalmente se encarga de convertir toda la información numérica generada por el módulo de procesamiento, en gráficos, curvas y animaciones que le permiten al usuario realizar un análisis rápido y preciso de los resultados obtenidos con el modelo.

En este orden de ideas, existen cuatro elementos fundamentales que marcan la calidad de un buen software de elementos finitos: un modelador gráfico, una familia de algoritmos de mallado (discretización), un conjunto de solucionadores o *solvers* y un código para graficar datos.

No obstante el uso cada vez más común de los paquetes de elementos finitos, el empleo masivo de esta herramienta ha encontrado dos restricciones importantes que han retrasado el avance de la misma. Estas restricciones están relacionadas con las exigencias computacionales requeridas por los algoritmos de solución, sumado al alto costo de los programas comerciales. La primera limitante ha encontrado alivio en el acelerado desarrollo del hardware que ha permitido hacer realidad el desarrollo de modelos de elementos finitos convencionales en computadoras personales con características estándar. Sin embargo la inversión asociada a la adquisición de licencias no hace fácil el acceso a esta tecnología a pequeñas empresas o grupos de investigación.

En este documento se presenta una atractiva alternativa para acceder a la tecnología de los software de elementos finitos, mediante el uso de paquetes licenciados bajo la filosofía *GPL*. Inicialmente se hace una revisión de diferentes software *GPL* de elementos finitos, en los cuales se analizan diferentes aspectos, buscando una caracterización completa de los mismos, lo que permita al lector adquirir elementos de juicio para la selección adecuada de alguno de estos paquetes. Posteriormente se desarrollan tres casos de estudio en diferentes campos (estructural, térmico y de dinámica de fluidos), comparando los resultados arrojados por diferentes software *GPL* y un software comercial (ANSYS®). Finalmente se concluye acerca del desempeño y exactitud de estos software *GPL* para aplicaciones de análisis por elementos finitos (*FEA*).

2. SOFTWARE CON LICENCIAS GPL EN APLICACIONES DE ELEMENTOS FINITOS

En el creciente mundo de los programas con licencias *GPL*, se encuentra una amplia variedad de software de elementos finitos, con diferentes características, capacidades, campos de aplicación, modos y plataformas de trabajo. Sin embargo todos tienen en común el libre uso y acceso a los códigos fuente, que le posibilitan al usuario el mejoramiento y adaptación del programa a sus necesidades particulares.

A continuación se hace una revisión de las características más relevantes de los programas *GPL* más populares para el modelamiento por elementos finitos (MEF).

2.1 Los Programas *gpl* Más Comunes Para El Mef

Una primera forma de clasificar los softwares de elementos finitos es de acuerdo a su campo de aplicación. Desde este punto de vista se pueden encontrar programas para uso básico general y para uso especializado. En el primer grupo se encuentra una buena parte de los software comerciales y muchos de los software *GPL* para MEF, los cuales normalmente permiten el análisis de problemas de mecánica estructural (estudio de cuerpos deformables), de transferencia de calor en estado estable y transitorio, y además cuentan con elementos formulados para el modelamiento de problemas en electromagnetismo y en algunos casos para el modelamiento en dinámica de fluidos. En este grupo de programas se pueden mencionar:

- *Calculix* [1], software *GPL* desarrollado por Guido Dhondt y Klaus Wittig desde 1998. Cuenta con un amplio conjunto de campos para el modelamiento de problemas.
- *CodeAster* [8], software creado por el departamento de investigación y desarrollo de EDF[®] - *Electricité de France*.
- *Tochnog* [9], producto de *FEAT (Finite Element Application Technology)*.

- *FreeFem* [6], desarrollado por *Laboratoire Jacques-Louis Lions* de *Université Pierre et Marie Curie*.

- *FElt* [12], desarrollado por Jason Gobat, cuenta con elementos formulados para análisis estructural lineal estático y dinámico, así como para problemas de tipo térmico.

El segundo grupo de software de elementos finitos, se encuentra constituido por programas para uso particular o específico. Se encuentran en este grupo programas *GPL* tales como:

- *Z88* [3], programa desarrollado en *University of Bayreuth*, que permite el análisis de problemas estructurales lineales, bidimensionales y tridimensionales, empleando elementos serendripitos. Su característica principal se encuentra en la alta afinidad con los programas *CAD* tradicionales y especialmente con el formato de exportación geométrica *DXF (Drawing Exchange Format)*.

- *SLFFEA* [13], es un programa escrito en lenguaje C para el análisis de problemas estructurales lineales en dos y tres dimensiones.

- *OpenFoam* [5], aunque tiene algunos elementos para el análisis estructural lineal estático y térmico, es un software especializado para el modelado de problemas de dinámica de fluidos con volúmenes finitos. Cuenta con una interfaz para usuario, pero su objetivo principal está orientado en ser una librería para la programación de problemas de *CFD (Computational Fluid Dynamics)* en lenguaje C++.

- *OpenFlower* [7], software para análisis *CFD* con volúmenes finitos desarrollado por Alexandre Chatelain.

- *SLFCFD* [14], desarrollado en ANSI C y se especializa en análisis de problemas por *CFD*.

- *Impact* [2], es un programa desarrollado en JAVA, especializado en el análisis de problemas de dinámica explícita.

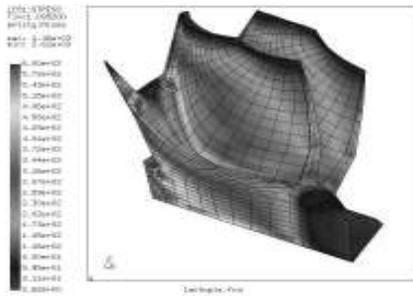


Figura 1. Análisis del rodete de un turbocompresor realizado en *Calculix*. (Imagen tomada de [1])

Figure 1. Turbocharger impeller analysis using *Calculix*. (Image taken from [1])

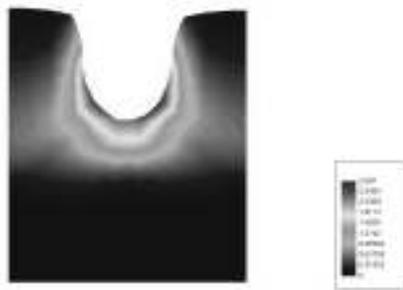


Figura 2. Análisis del forjado de una placa metálica desarrollado en *Tochnog*. (Imagen tomada de [9])

Figure 2. Metallic sheet forging analysis in *Tochnog*. (Image taken from [9])

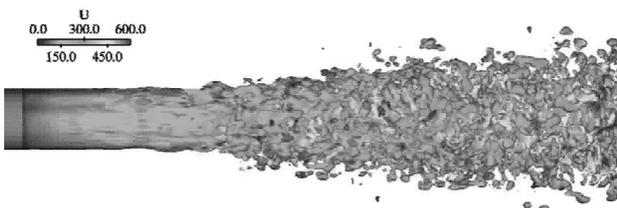


Figura 3. Análisis del rompimiento de chorro en un inyector Diesel desarrollado en *OpenFoam*. (Imagen tomada de [5])

Figure 3. Injector break-up diesel spray analysis using *OpenFoam*. (Image taken from [5])

En la Tabla 1. Clasificación de los programas *GPL* para *FEM* según sus campos de aplicación. se muestra una clasificación de los programas anteriormente presentados según el tipo de

análisis permitido o los campos de aplicación cubiertos.

Otro tipo de software esta compuesto por aquellos programas que no integran los tres módulos básicos de un programa de elementos finitos (pre-procesador, procesador y post-procesador), y que están diseñados para interactuar complementariamente con otros programas. En este grupo se encuentran normalmente programas con módulos de pre y post-procesamiento, los cuales exportan e importan información en formatos estándar (tales como formatos de *Abaqus* [17], *Nastran* [18], *Cosmos* [19], etc.) hacia otros programas encargados de hacer los cálculos o solución del problema. Dentro de este grupo de programas se encuentran *NetGEN* [10] y *NGSolve* [11], que hacen parte del proyecto *hp-fem* desarrollado en *Johannes Kepler University Linz*, en donde el primer programa es el módulo de pre y postprocesamiento, mientras el segundo integra el componente de cálculo o solución. Una asociación similar la constituye *Gmsh* [15] y el solver *GetDP* [16], software desarrollados en *University of Liège* y *Case Western Reserve University*, respectivamente. *Salome* [4] es una plataforma desarrollada por *Open Cascade* y *Eads Matra Datavision Company*, que integra un poderoso módulo de modelamiento geométrico y mallado, así como una interfaz para la visualización de resultados.

Tabla 1. Clasificación de los programas *GPL* para *FEM* según sus campos de aplicación.

Table 1. *GPL* and *FEM* software classification by working fields

Programa	1	2	3	4	5	6	7	8
<i>Calculix</i>	R	R	R	R	R			
<i>Z88</i>	R							
<i>OpenFOAM</i>	R			R	R	R	R	R
<i>Impact</i>								
<i>Code-Aster</i>	R	R	R	R	R			
<i>Gmsh-GetDP</i>	R		R	R	R			
<i>Tochnog</i>	R	R	R	R	R	R	R	R
<i>NetGen-NGSolve</i>	R			R	R			
<i>FreeFem</i>	R	R	R	R	R	R	R	R
<i>OpenFlower</i>								
<i>FElT</i>	R		R					

Tabla 1. Clasificación de los programas GPL para FEM según sus campos de aplicación.

Table 1. GPL and FEM software classification by working fields

Programa	9	10	11	12	13	14	15
<i>Calculix</i>						☑	
<i>Z88</i>							
<i>OpenFOAM</i>	☑	☑	☑	☑	☑		
<i>Impact</i>							☑
<i>Code-Aster</i>						☑	
<i>Gmsh-GetDP</i>						☑	
<i>Tochnog</i>	☑	☑	☑	☑	☑		
<i>NetGen-NGSolve</i>						☑	
<i>FreeFem</i>	☑	☑	☑	☑	☑	☑	
<i>OpenFlower</i>	☑		☑		☑		
<i>FElt</i>							

Campos de aplicación:
Working fields:

- (1)Módulo de análisis estático lineal.
- (2)Módulo para el análisis estático no-lineal, p.ej. elasticidad no lineal, grandes deformaciones, hiperelasticidad, plasticidad, etc.
- (3)Módulo para análisis dinámico, p.ej. análisis modal.
- (4)Módulo para modelado de problemas de campo en estado estacionario, p. ej. problemas térmicos de conducción-convección o problemas acusticos.
- (5)Módulo para el análisis de problemas térmicos en régimen transitorio.
- (6)Módulo de CFD para fluidos newtonianos.
- (7)Módulo de CFD para fluidos no-newtonianos.
- (8)Módulo de CFD para estado estable.
- (9)Módulo de CFD para régimen transitorio.
- (10)Módulo de CFD para flujo laminar.
- (11)Módulo de CFD para flujo turbulento.
- (12)Módulo de CFD para flujo compresible.
- (13)Módulo de CFD para flujo incompresible.
- (14)Módulo para el modelado de problemas electromagnéticos.
- (15)Módulo para el análisis de problemas con dinámica explícita.

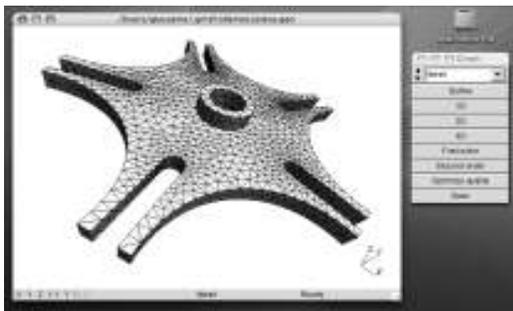


Figura 4. Modelamiento geométrico 3D realizado en *Gmsh*. (Imagen tomada de [15])

Figura 4. 3D geometrical modeling in *Gmsh*. (Image taken from [15])



Figura 5. Modelo tridimensional de un eje analizado con *NGSolve*. (Imagen tomada de [11])

Figure 5. Tridimensional model of an axle using *NGSolve*. (Imagen tomada de [11])

En la Tabla 2 se muestran los módulos integrados en cada uno de los programas GPL consultados.

Tabla 2. Clasificación de los programas *GPL* para *FEM* de acuerdo a los módulos integrados en estos.

Table 2. GPL FEM software classification by internal modules within.

	Pre-procesador	Procesador	Post-procesador
<i>Calculix</i>	✓	✓	✓
<i>Z88</i>		✓	✓
<i>OpenFOAM</i>	✓	✓	✓
<i>Impact</i>	✓	✓	✓
<i>Code-Aster</i>		✓	
<i>Gmsh</i>	✓		✓
<i>GetDP</i>		✓	
<i>Tochnog</i>		✓	
<i>NetGen</i>	✓		✓
<i>NGSolve</i>		✓	
<i>FreeFem</i>		✓	✓
<i>OpenFlower</i>		✓	
<i>FElt</i>		✓	✓

2.2 Caracterización de los Programas *GPL* Revisados

Con el ánimo de evaluar de forma clara y objetiva los programas para MEF revisados, se analizaron algunas características relevantes en cada uno de estos: calificando su desempeño, calidad en el soporte brindado, la permanente evolución (mejoramiento) del programa, el nivel

de difusión y conocimiento en el medio, entre otros aspectos. A continuación se presentan los resultados obtenidos en cada uno de estos temas.

2.2.1 Evolución y mejoramiento

El gran dinamismo en la evolución y mejoramiento permanente de los programas con licencias GPL es una característica inherente a estos, derivada del activo aporte de los usuarios al código original. Con el fin de evaluar este aspecto se revisaron las fechas de publicación de las diferentes versiones lanzadas, desde el nacimiento del programa hasta la fecha, y se construyó el gráfico mostrado en Figura 6.

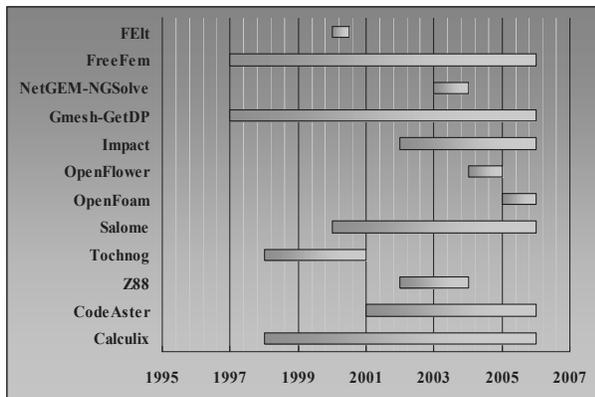


Figura 6. Evolución de las versiones de los programas GPL analizados.

Figure 6. GPL software review evolution.

2.2.2 Calidad de la documentación y soporte brindado a los usuarios

En este punto se revisó la documentación, tal como manuales de instalación, de usuario, tutoriales, etc., que puede ser consultada por el usuario para iniciar y profundizar en el manejo del programa; evaluando la pertinencia de las instrucciones y recomendaciones realizadas, así como la sencillez para entender las mismas. De igual forma se revisaron las otras formas de soporte o asesoría en el manejo o instalación del programa. Se encontró que buena parte de los programas cuentan con foros de discusión o lista de correos mediante los cuales el usuario puede realizar las consultas. En casi todos los casos se pudo verificar la calidad y rapidez en la asesoría prestada, la cual en algunas ocasiones provenía

directamente de alguno de los autores del programa.

Tabla 3. Calidad de la documentación y soporte de los programas.

Table 3. Review software quality and client support.

Programa	Manuales y Tutoriales					Foros de discusión			
	1	2	3	4	5	1	2	3	4
Calculix	☆	☆	☆	☆	☆				☞
Z88	☆	☆	☆					☞	
OpenFOAM	☆	☆	☆	☆	☆				☞
Impact	☆	☆	☆					☞	
Code-Aster	☆	☆							☞
Gmsh-GetDP	☆	☆	☆	☆				☞	
Tochnog	☆						☞		
NetGen-NGSolve	☆	☆				☞			
FreeFem	☆	☆	☆	☆				☞	
OpenFlower	☆							☞	
FELt	☆	☆	☆	☆		☞			

En el caso particular del programa CodeAster, éste cuenta con una completa documentación, sin embargo la misma esta escrita en francés y su consulta se hace especialmente difícil. En otros casos, como en el del programa OpenFlower, la documentación se encuentra en desarrollo. En la Tabla 3 se muestra la evaluación para estos aspectos de acuerdo a una escala ascendente.

2.2.3 Tipo de interfaz de usuario y curva de aprendizaje

La facilidad de crear un modelo, realizar los cálculos y visualizar los resultados, así como la rapidez con la que un usuario se habitúa al trabajo con un nuevo software, son quizás dos de los elementos más relevantes que se tienen en cuenta al momento de evaluar un programa, especialmente si la evaluación esta basada en términos de la productividad. En este ítem se hace clasificación de los programas para MEF consultados, de acuerdo a la experiencia obtenida con el trabajo en cada uno de ellos; la Tabla 4 reúne esta información.

Tabla 4. Calidad de interfaz de usuario y curva de aprendizaje.

Table 4. User interface and learning curve quality.

Programa	Clasificación			
	1	2	3	4
<i>Calculix</i>			✌️	
<i>Z88</i>		👍		
<i>OpenFOAM</i>			✌️	
<i>Impact</i>				👉
<i>Gmsh-GetDP</i>			✌️	
<i>Tochnog</i>	👎			
<i>NetGen-NGSolve</i>		👍		
<i>FreeFem</i>		👍		
<i>FElt</i>		👍		

Con el fin de sustentar la anterior evaluación se comentan algunos aspectos relacionados con el manejo de cada uno de los programas.

- *Calculix*: la interfaz de usuario la constituye el programa *cgx*, el cual permite crear el modelo geométrico, realizar el mallado, generar los archivos de cargas y restricciones y el post-procesamiento de la información. Aunque contiene un área de visualización gráfica del modelo y ciertas actividades de selección son posibles por medio del ratón del computador, la mayor parte del trabajo se hace digitando texto en una línea de comando. Por lo anterior resulta imprescindible el conocimiento del nombre y sintaxis de cada uno de los comandos, o por lo menos la consulta constante del manual de usuario. A pesar de esto la calidad de la documentación y la lógica manejada en los comandos hace que el programa sea fácilmente manejable.
- *Z88*: En el manejo de este programa se destaca la carencia de un pre-procesador, si el modelo se trata de un problema bidimensional esta etapa del modelado debe ser desarrollada en un programa *CAD* que permita la exportación de archivos gráficos en formato *DXF*. Este archivo debe contener la información geométrica, del material, así como los datos básicos para la generación de la malla. Si el problema es tridimensional, el pre-procesamiento debe hacerse en un programa para MEF que permita la exportación en formato de *Cosmos* o *Nastran*.
- *OpenFoam*: La principal herramienta de trabajo de *OpenFoam* es un programa

denominado *FoamX*. Este programa es una interfaz en Java desde donde se ejecutan las demás herramientas del paquete. Sin embargo la visualización gráfica del modelo en tiempo real no es posible, por lo que no se puede realizar un seguimiento permanente al estado del modelo, dificultando el trabajo geométrico-espacial.

- *Impact*: Cuenta con una interfaz ágil y agradable, separada en módulos de pre-procesamiento, procesamiento y post-procesamiento. El manejo de los comandos e instrucciones se realiza por medio del ratón del computador, lo que hace más llamativa la operación del programa.
 - *Gmsh-GetDP*: *Gmsh* es un paquete de pre y post-procesamiento que posee una interfaz gráfica simple e intuitiva, que le permite al usuario una rápida identificación de los comandos necesarios para generar el modelo. Por otro lado, en *GetDP* el procesador necesita la elaboración de un archivo que contiene el modelo matemático del problema.
 - *NetGen-NGSolve*: Esta pareja complementaria de programas tiene características similares a la conformada por *Gmsh* y *GetDP*, con la diferencia que el programa *NetGen* requiere la importación de la geometría del modelo.
 - *FreeFem*: Es un programa que permite una amplia formulación de problemas, sin embargo su manejo se limita a la edición manual de archivos de texto, con lo cual la adaptación y aprendizaje se dificulta enormemente.
 - *FElt*: A pesar de que el archivo de entrada debe ser generado por el usuario manualmente, la versión para Windows cuenta con un editor que hace simple esta tarea.
- Es importante aclarar que esta clasificación solo tiene en cuenta los módulos de pre y post-procesamiento, apartando el módulo de procesamiento, el cual se evaluará con algunos casos estudiados.

2.2.4 Plataforma de trabajo

Pese a que el movimiento del software libre *GNU* inicia con aplicaciones en sistemas *Unix*, difundándose posteriormente en sistema operativo *Linux*, resulta conveniente contar con versiones de los programas compatibles en diferentes sistemas operativos. Desde este punto

de vista se revisaron las versiones de los software consultados y se tabularon de acuerdo a la disponibilidad de versiones en los sistemas operativos más disponibles en nuestro medio. En la Tabla 5 se muestra la información consultada en este ítem.

Tabla 5. Sistema operativo o plataforma de trabajo.
Table 5. Operating system.

	Linux	Windows
<i>Calculix</i>		
<i>Z88</i>		
<i>OpenFOAM</i>		
<i>Impact</i>		
<i>Code-Aster</i>		
<i>Gmsh</i>		
<i>GetDP</i>		
<i>Tochnog</i>		
<i>NetGen</i>		
<i>NGSolve</i>		
<i>FreeFem</i>		
<i>OpenFlower</i>		
<i>FElt</i>		

2.2.5 Lenguaje de desarrollo

Recordando que uno de los principios del software libre es la libre distribución de las fuentes, es decir, el código escrito en algún lenguaje de programación, donde se encuentran las diferentes funciones que hacen que el programa trabaje. Resulta importante conocer el lenguaje de desarrollo de modo que se puedan entender las diferentes posibilidades para la manipulación o modificación de dichos códigos. En la Tabla 6 se listan los lenguajes de programación en los cuales están escritos los códigos revisados.

Tabla 6. Lenguaje de desarrollo.
Table 6. Base language

	Lenguaje
<i>Calculix</i>	C++
<i>Z88</i>	C
<i>OpenFOAM</i>	C++
<i>Impact</i>	Java
<i>Code-Aster</i>	Fortran
<i>Gmsh</i>	C++
<i>GetDP</i>	C++
<i>Tochnog</i>	C
<i>NetGen</i>	C
<i>NGSolve</i>	C
<i>FreeFem</i>	C
<i>OpenFlower</i>	C
<i>FElt</i>	Fortran

2.2.6 Formatos para la importación y exportación de datos

Muchos de los programas *GPL* revisados tiene una gran posibilidad de intercambio de datos (importación y/o exportación) con programas de *CAD* (dibujo), de elementos finitos, o con programas especializados en graficar datos. Este aspecto es importante si se requiere suplir las deficiencias de un programa, desarrollando ciertas partes del modelo en otro software.

Tabla 7. Tipos de archivos soportados por los programas
Table 7. File types.

	Nativo		Importa	Exporta
	Entrada	Salida		
<i>Calculix</i>	Abaqus,	.FRD	.stl, .frd, Duns, OpenFOAM	Abaqus, Ansys, CodeAster, Duns, OpenFOAM, .stl
<i>Z88</i>	Z88i	Z88o	.dxf, Nastran, Cosmos.	
<i>OpenFOAM</i>	OpenFoam	OpenFoam	Fluent	Ansys, Cfx, Gambit, Gmesh, Kiva, Adventure System, Netgen, Star-CD
<i>Impact</i>	.in	.res	Nastran, Gmsh, .stl	-
<i>Code-Aster</i>	-	-	Gmsh, Ideas, Gibi	-
<i>Gmsh-GetDP</i>	-	-	-	-
<i>Tochnog</i>	-	-	-	-
<i>NetGen-NGSolve</i>	.geo, .pde	-	.geo, .stl, .stlb, .step, .stp, .in2d, .igs, .iges	Tecplot, Tochnog, Abaqus, Fluent, Permas, Feap, Elmer, VrmI, Gmsh, .stl
<i>FreeFem</i>	.pde	.pde	-	-
<i>OpenFlower</i>	.flw	.pos	Gmsh	
<i>FElt</i>	-	-	-	-

3. CASOS DE ESTUDIO

De acuerdo a las características revisadas anteriormente, se escogieron tres de los programas GPL: *Calculix*, *Z88* y *OpenFoam*) para desarrollar tres modelos diferentes. L primer modelo es de tipo de tipo estructural estático, el segundo de tipo térmico y el tercero de dinámica de fluidos. A continuación se exponen los modelos desarrollados y los resultados obtenidos en los programas revisados, comparando estos mismos con los datos arrojados por un programa comercial de elementos finitos (*Ansys*) y con la solución analítica.

3.1 Modelado del concentrador de esfuerzos en una placa con agujero centrado

El primer modelo consistió en una placa cuadrada con agujero central sometida a carga axial. En la Figura 7 se muestra el problema desarrollado, así como la malla empleada para los modelos en todos los programas. En la Figura 8 se muestran los gráficos de los contornos de esfuerzo normal en la dirección *x*, obtenidos con los modelos desarrollados en los programas *Calculix*, *Z88* y *OpenFoam*.

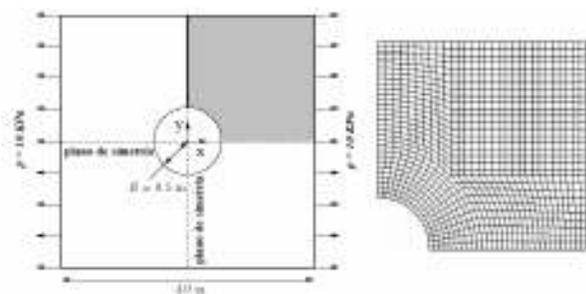


Figura 7. Modelo de placa con agujero central desarrollado.
Figure 7. Central orifice plate developed model.

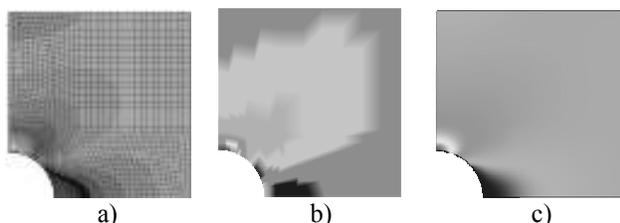


Figura 8. Gráficos de contorno para el esfuerzo normal en *x* obtenidos con el software
a) *Calculix* b) *Z88* c) *OpenFoam*.

Figure 8. Contour graphs to normal stress in *x* direction using the software: a) *Calculix* b) *Z88* c) *OpenFoam*.

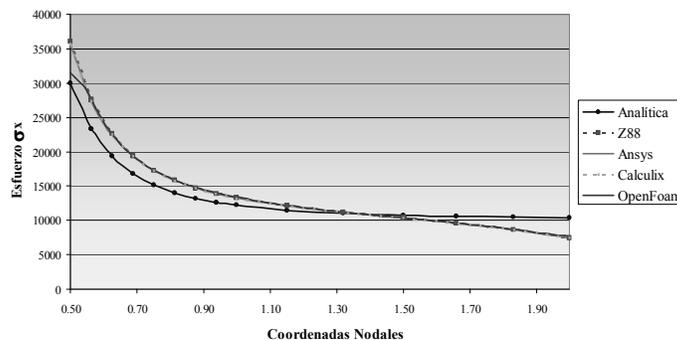


Figura 9. Perfiles de esfuerzo sobre la línea X=0, obtenidos con los diferentes programas empleados.
Figure 9. Stress profiles over the X=0 line.

En la Figura 9 se muestra el perfil de esfuerzo obtenido con cada uno de estos programas, comparándolos adicionalmente con el resultado del modelo desarrollado en *Ansys* y con el perfil analítico, el cual se expresa como:

$$\sigma_x = p \left(1 + \frac{R^2}{2y^2} + \frac{3R^4}{2y^4} \right) \quad |y| \geq R \quad (1)$$

3.2 Modelado de un intercambiador de aletas circulares

El segundo modelo desarrollado consistió en un intercambiador de calor de aletas circulares, tal como el mostrado en la Figura 10. El intercambiador fue modelado a partir de una geometría bidimensional axi-simétrica, empleando elementos de cuatro lados y ocho nodos para la ecuación de transferencia de calor en estado estable.

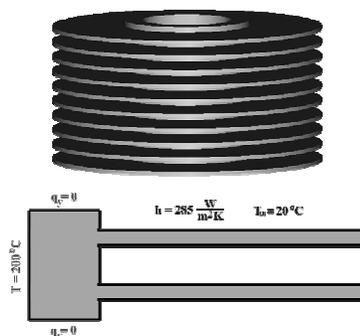


Figura 10. Modelo de un intercambiador de aletas circulares.

Figure 10. Circular fins heat exchanger model

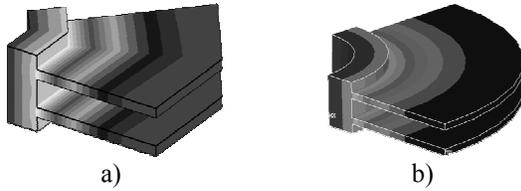


Figura 11. Gráficos de contorno para la temperatura obtenidos con el software a) *Calculix* b) *Ansys*.

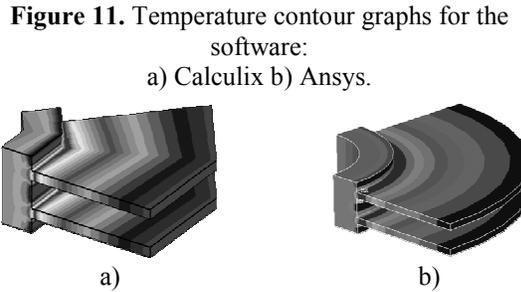


Figura 12. Gráficos de contorno para flujo de calor obtenidos con el software: a) *Calculix* b) *Ansys*.

Figure 12. Heat flux contour graphs for the software: a) *Calculix* b) *Ansys*.

En la Figura 11 se muestran las graficas de contorno obtenidas para los valores de temperatura encontrados con los modelos desarrollados en *Calculix* y *Ansys*. De forma similar en la Figura 12 se encuentran dos gráficos para los valores de flujo de calor en la aleta calculados con los mismos programas. En la Figura 13 y en la Figura 14 se muestran los perfiles de temperatura y flujo de calor, respectivamente, medidos a lo largo de la aleta, comparándolos con el perfil analítico derivado de la solución de la ecuación diferencial tipo Bessel que rige el problema.

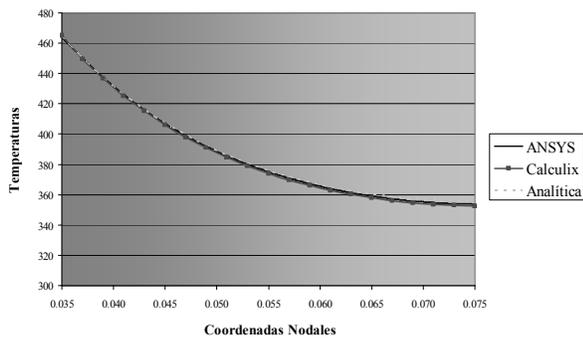


Figura 13. Comparación del perfil de temperatura a lo largo de la aleta obtenido con los programas *Calculix* y *Ansys* y el perfil analítico real.

Figure 13. Temperature through the fin profile comparison between *Calculix* and *Ansys* software and the analytical solution.

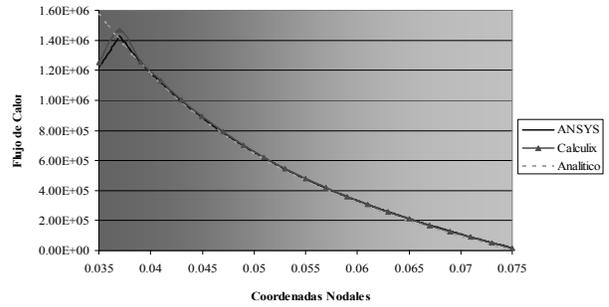


Figura 14. Comparación del perfil de flujo de calor a lo largo de la aleta obtenido con los programas *Calculix* y *Ansys* y el perfil analítico real.

Figure 14. Heat flux through the fin profile comparison between *Calculix* and *Ansys* software and the analytical solution.

Es de aclarar que la diferencia entre los perfiles de flujo de calor obtenidos con los modelos de elementos finitos y el perfil trazado a partir de la solución analítica, se debe a que el dominio del modelo analítico esta definido únicamente por la aleta, mientras que el análisis por elementos finitos tiene en cuenta el efecto distorsionador producido por el cuerpo cilíndrico al cual está pegado la aleta.

3.3 Modelado De Un Flujo Potencial A Través De Un Cilindro

El tercer modelo desarrollado consistió en un flujo, incompresible, no viscoso, circulando alrededor de un cilindro como se ilustra en la Figura 15. Para este caso se realizo un análisis bidimensional con elementos de cuatro lados, ocho nodos y suponiendo un flujo de tipo potencial, empleando igual numero y distribución de elementos para cada uno de los modelos desarrollados.

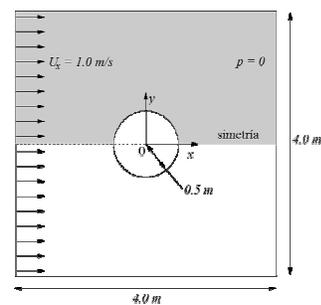


Figura 15. Modelo de flujo alrededor de un cilindro.

Figure 15. Model for the Flow over a cylinder.

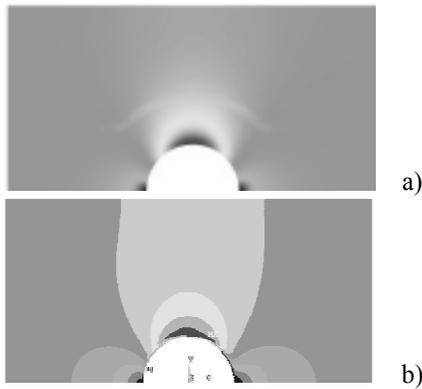


Figura 16. Gráficos de contorno de velocidad total obtenidos con el software a) *OpenFoam* b) *Ansys*.
Figure 16. Total velocity contour graphs for yhe software a) OpenFoam b) Ansys.

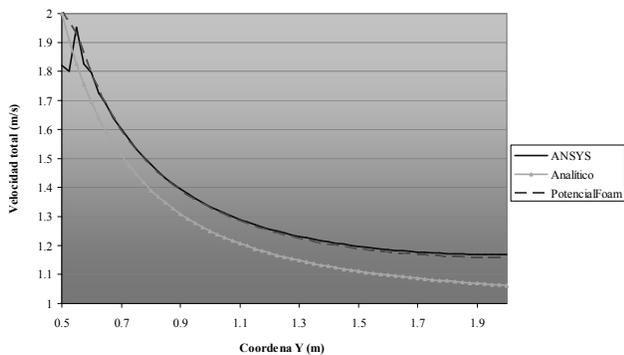


Figura 17. Perfiles de velocidad total a lo largo de la línea $x=0$. a) *OpenFoam* b) *Ansys*.
Figure 17. Total velocity profiles on the $x=0$ line a) OpenFoam b) Ansys.

En la Figura 16 se comparan los gráficos de contorno obtenidos para la velocidad total del flujo en *OpenFoam* y en *Ansys*, mientras que en la Figura 17 se muestran los perfiles de velocidad total del flujo, trazados a lo largo de la dirección $x=0$. El perfil analítico trazado en esta última figura obedece a la solución definida por la ecuación:

$$U_r = U \sqrt{\left[1 - \left(\frac{r}{d}\right)^2 \cos 2\theta\right]^2 + \left[\left(\frac{r}{d}\right)^2 \sin 2\theta\right]^2} \tag{2}$$

donde:

- U_r : es la velocidad total de un punto del fluido.
- U : es la velocidad del fluido a la entrada.
- r : es el radio del cilindro.
- d : es la distancia del punto al centro del cilindro.

θ : es la posición angular del punto medida a partir del eje x .

4. CONCLUSIONES

Después de estudiar y caracterizar los programas GPL más populares para el modelamiento por elementos finitos, se pueden plantear las siguientes conclusiones:

- Aunque en muchos casos la filosofía de trabajo o la interfaz de usuario asociada a un programa específico hace difícil la adaptación para nuevos usuarios, los resultados obtenidos en los casos de estudio permiten concluir sobre la buena exactitud y confiabilidad de los algoritmos numéricos implementados en estos programas.

- Algunos programas como *GetDP*, *FreeFem* o *NGSolve*, permiten una gran flexibilidad en el estudio de problemas de diferente naturaleza, pues admiten la formulación de modelos no tradicionales a partir de la definición de la ecuación diferencial parcial asociada al fenómeno que se desea estudiar. Esta característica tiene un impacto fuerte en la metodología de trabajo y en la adaptación al manejo del programa, especialmente para usuarios que migran de paquetes comerciales, en los cuales la formulación de los elementos esta establecida de antemano.

- La conectividad con diferentes programas de CAD (dibujo), de elementos finitos o de graficación de datos, es una característica común en la gran mayoría de los programas GPL consultados. De esta forma los programas sustentan sus debilidades en las capacidades de otros programas, permitiendo a los desarrolladores potenciar las fortalezas propias del programa. Por este motivo es muy común encontrar programas GPL para MEF que solo incorporan algunos de los tres módulos comunes en un programa de elementos finitos (pre-procesamiento, procesamiento y post-procesamiento), permitiendo la exportación o importación de datos hacia o desde otros programas, normalmente también licenciados bajo el esquema GPL.

- Se encontró que buena parte de los programas *GPL* consultados cuentan con un ágil y ameno soporte para el manejo o instalación del programa. Estas ayudas se encuentran a través de la documentación desarrollada para el programa y/o por medio de listas de correo o foros, en los cuales usuarios o desarrolladores brindan información pertinente para el desarrollo de los modelos. Dado que la filosofía de las licencias *GPL* le permite a los usuarios el conocimiento y edición del código fuente del programa, el número de usuarios capacitados que pueden brindar un soporte es cada vez es mayor.
- Muchos de los programas *GPL* analizados son vigentes y tienen un gran dinamismo en la evolución y mejoramiento del código, lo cual se refleja en frecuencia de lanzamiento de nuevas versiones; esto es consecuencia del aporte permanente de los usuarios al código fuente del programa.

REFERENCIAS

- [1] DHONDT, G., WITTIG K. Calculix: a free software three-dimensional structural finite Element Program. <http://www.dhondt.de/index.html>
- [2] HORTON, R., FORSELL, J. Impact: a free explicit dynamic finite element program. <http://impact.sourceforge.net/>
- [3] RIEG, F. Z88: a fast, powerful and compact Finite Elements Analysis Program. <http://z88.uni-bayreuth.de/>
- [4] OPEN CASCADE SAS (OCC). SALOME: The Open Source Integration Platform for Numerical Simulation. <http://www.salome-platform.org/home/presentation/overview/>
- [5] OPENCFD LTD. OpenFOAM: The Open Source CFD Toolbox. <http://www.opencfd.co.uk/openfoam/index.html#openfoam>
- [6] PIRONNEAU, O., HECHT, F., LE HYARIC, A. FreeFem. <http://www.freefem.org/>
- [7] CHATELAIN, A. OpenFlower. <http://openflower.sourceforge.net/>
- [8] EDF - R&D, CodeAster. <http://www.codeaster.net/>
- [9] FEAT, Tochnog: Finite Element Analysis. <http://www.feat.nl/tochnog/index.html>
- [10] SCHÖBERL, J. NetGen:automatic mesh generator. <http://www.hpfem.jku.at/netgen/index.html>
- [11] SCHÖBERL, J. NGSolve:3D Finite Element Solver <http://www.hpfem.jku.at/ngsolve/index.html>
- [12] GOBAT, J. FELt: Finite Element Analysis. <http://felt.sourceforge.net/>
- [13] SAN LE. SLFFEA: San Le's Free Finite Element Analysis. <http://slffea.sourceforge.net/index.html>
- [14] SAN LE. SLFCFD: San Le's Free Computational Fluid Dynamics. <http://slfcfd.sourceforge.net/>
- [15] GEUZAIN, C., REMACLE J.F. Gmsh: a three-dimensional finite element mesh generator with built-in pre- and post-processing facilities. <http://www.geuz.org/gmsh/>
- [16] GEUZAIN, C., REMACLE J.F. GetDP: a general environment for the treatment of discrete problems. <http://geuz.org/getdp/>
- [17] ABAQUS Homepage. <http://www.hks.com/>
- [18] NASTRAN Homepage. <http://www.mscsoftware.com/products/nastran.cfm?Q=131&Z=401>
- [19] COSMOS Homepage. <http://www.cosmosm.com/>