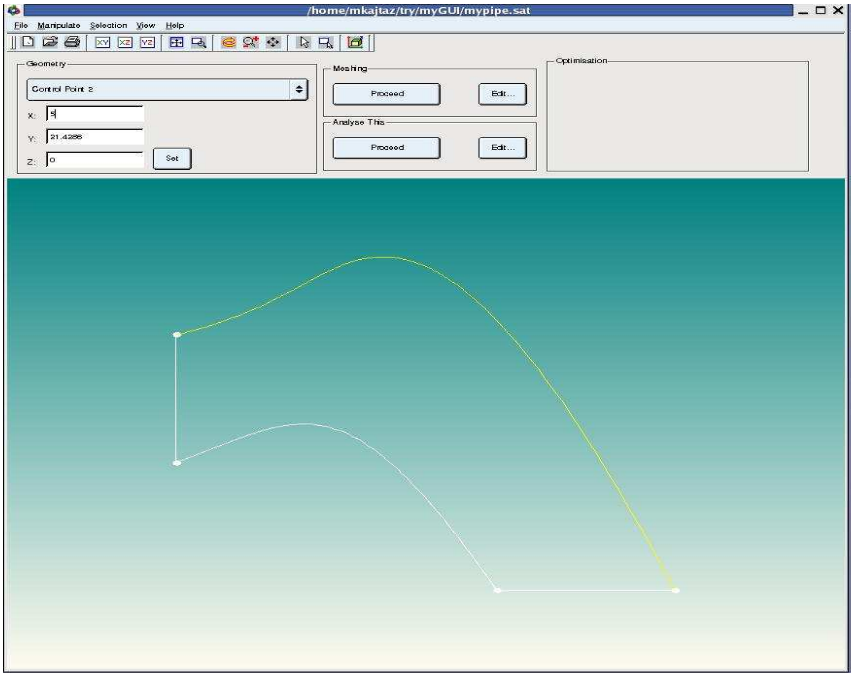
Geometric Optimisation of Turbulent Pressure Drop in a 2D/3D Duct



Student: Mladenko KAJTAZ

Association: Victorian Partnership for Advanced Computing

Supervisors: Chris Seeling (Head of CfCP), Jarrod Sinclair (Computational Engineering)

Discipline: Computational Engineering

Research Objective

The objective of the research was to build a software tool that would provide an insight into the effects of shape and route a pipe have on the turbulent pressure drop in the pipe. By discovering important parameters and relationships, a user would be equipped with the knowledge to find the optimal shape that would minimise pressure drop. The solution to the problem was approached from two different aspects. First, to use an artificial intelligence to find the optimal solution from a defined search spectrum. The other approach was to provide a visual tool, allowing a user to modify the shape and the route manually, and view the changes, until the optimal solution is found. In both approaches, similar tools have been used. The common tools have been: ACIS Modeller - used to create geometry, GAMBIT – used to mesh the geometry and FLUENT – used to perform CFD analysis of the geometry. The approach specific tools vary from the GAlib – genetic algorithms library for the first approach, and HOOPS Framework – for visualisation and Qt as a GUI Toolkit, for the second approach.

Motivation / Significance

The mechanics of fluids is a relatively unexplored area when compared to other engineering fields. Solutions to problems from this field are heavily approximated due to lack of exact science/knowledge. These approximations often involve large errors due to uncertainties in behaviour of fluids. Having a tool that can provide an insight into the behaviour of fluids would improve the approximation and therefore by being more efficient, eliminate unnecessary overheads.

The research found its motivation in a standard problem, designers from the automotive industry face when designing an intake manifold. The designers are required to come up with a design that would deliver a required mass flow rate at as high pressure as possible. The main obstacle in achieving the aim is a restricted space under a bonnet. The designers are given the starting and the ending points/planes of the manifold, and finding a route and a shape of the manifold that fits the space restriction and at the same time minimises pressure drop is left to their discretion. The current design procedure is based on a repetitive two-step process: 1. make a design and 2. run CFD simulation on the design. The problem with this is a long turnaround period of the design, which may become frustrating to the designers because it involves long waiting periods that may effect/restrict their creativity and flow of new ideas/improvements.

The research aimed to develop a tool or a set of tools (a tool chain) that would get the designers out of the loop and the waiting period, allowing them to focus on what they are the best at (creativity and designing). The repetitive tasks should be left to less expensive computer power.

The project branched off in two directions. The first branch focused on building a tool that helps a user to investigate relationships between shape/route of a 2D pipe and pressure drop through it, which is presented in this document in more detail. The second one focused on optimising shape/route of a 3D pipe, and results will be presented at conferences in a couple of months. The first one will allow users to understand how a fluid behaves in different pipe shapes and assist in solving problems that could be modelled as 2D internal flow problems, whereas the second one finds more application in areas such as the described design of an intake manifold.

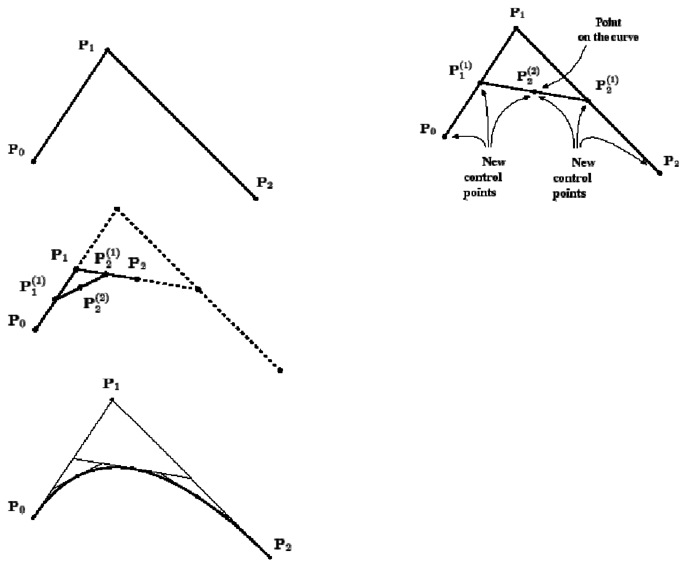
Science Background

The Geometric Modelling

The physical world is often desired to be symbolically modelled. There have been many attempts by mathematicians to do so in the past. However, it is only recently that the knowledge of geometry is used in automated systems known as Geometrical Modellers. In such modellers the mathematics of shapes is encoded in computer programs, therefore, geometric entities such as points, vectors, planes etc, are defined in programs and from them more complex geometric models could be built. Points are defined by coordinates, one for each dimension. Lines are defined by two points. The entity that derives from a line is a vector; a line segment between two points that possesses magnitude (length), direction and sense. It is defined as a difference between the two bounding points i.e. $\vec{v} = \langle Q - P \rangle$, $\vec{v} = \langle x_Q - x_P, y_Q - y_P, z_Q - z_P \rangle$.

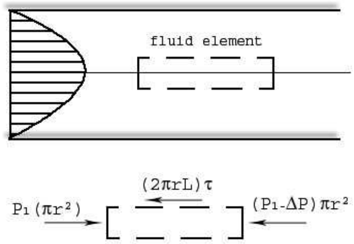
After a point (*zero-dimensional* entity), and a line (*one-dimensional* entity), comes a plane as a *two-dimensional* entity. A plane is defined by three points or a line and a point (which is again three points since a line is defined by two points). In 3D space, a plane is a locus of points.

A curve as a complex entity could be approximated by a series of vectors added up together. This mathematical curve formulation is extremely useful in modelling and design and also easily adopted to be used on a computer. According to this formulation, a curve is defined by a series of control points. These are the ends of a polygon that the vectors enclose.



The Frictional Pressure Drop

In situations such is fluid distribution networks, a fluid flow is forced through ducts. Due to viscosity of the fluid, shear forces/stresses exist near the duct's walls. This friction is directly related to the pressure drop.



If a cylindrical fluid element of radius *r* and length *L* oriented coaxially with the duct carrying incompressible, steady and fully developed flow is considered; a relationship between pressure drop, Δ**P** and shear stress, τ could be obtained. Solving the static forces of the control volume:

$$P_1 \pi r^2 - (P_1 - \Delta P)\pi r^2 - 2\pi r L \tau = 0$$

$$\frac{\Delta P}{L} = \frac{2\tau}{r} \Rightarrow \tau = \frac{\Delta P}{2L} r$$

Since the shear stress varies linearly from zero at the centreline to the maximum at the duct surface, the above expression for the maximum frictional pressure drop becomes:

$$\frac{\Delta P}{L} = \frac{2\tau_w}{D/2} \Rightarrow \Delta P = \frac{4L}{D}\tau_w$$

As it can be seen, the pressure drop depends on the duct's geometry and τ_w, the wall shear stress, which is a result of a friction between the duct walls and the fluid. $\tau_w \mid (\mu, V)$. Some elaborate mathematics has been used to express the relationship between the wall shear stress, the velocity and the viscosity of the fluid for different flow regimes and pipes geometries. The well known the Darcy-Weisbach equation $\Delta p_f = f \frac{L}{D} \rho \frac{V^2}{2}$, manages to reasonably approximate the relationship but because the equation is based on many assumptions, its use is bound by some conditions. Therefore, for some special cases experimental correction factors are introduced to the equation $\Delta p_f = K \mid f \frac{L}{D} \rho \frac{V^2}{2}$

The Optimisation (Genetic Algorithm)

Improvements in computer science and engineering modelling equipped designers to use computational models instead of real world prototypes in many situations. The computational models are cheaper and faster to run and therefore allow users to explore various design scenarios, searching for the optimal solution. Even though the computational prototyping is fast it is not fast enough to cover all possible cases/situations. Therefore, the users have to settle for near optimal but still acceptable solutions. There are many optimisation algorithms that make selections of input parameters based on statistics and probability, so on this way a potential optimal solution is discovered quicker. One such algorithm is the Genetic Algorithm. The Genetic Algorithms are part of evolutionary programming, inspired by Darwin's theory of evolution. In other words, problems are solved by an evolutionary process resulting in a best solution (fittest survivor). They are search algorithms based on mechanics of natural genetics. The advantage is that they can search complex and large amount of space efficiently and locate *near* optimal solution rapidly.

Methodology

The requirements are broken down in smaller modules.
The modules that have a common features or tasks are grouped together to form new modules
Literature and references are consulted to check how the same or similar requirements are met
Depending on the findings in the literature and the references, either new approaches are created or the existing ones are modified
The project is broken down in three major modules:
Interface: a module that interacts with a user, collects values for the necessary parameters, creates geometry and presents the results
Meshing: a module that prepares geometry for numerical analysis, i.e. creates mesh of elements
CFD analysis: performs calculations and returns the results
Familiarisation with the libraries used so, at least, the functional requirements are met
Each of the modules is developed to meet the basic functionality needs
Modules are integrated in a program
Integration issues of modules and programs are addressed, sometimes requiring the complete re-design of the modules
The modules are patched up to add new functionalities
Some re-designing may take place so the software could be more object oriented
Functional requirements and some other additional requirements are met, then testing proceeded
Debugging – solve the problems outlined in the testing phase
The source code is cleaned up/prepared for release
The source code is commented and documented
The software documentation is developed
HOW-TO Guides are created
The final report is written and submitted

Coding Aspects

The project was an interesting blend of several C++ libraries and applications; therefore the choice of programming language was dictated by the compatibilities of the libraries.

The ACIS Modeller is a collection of dynamically linked libraries (DLLs) and C/C++ header files that are used to create a 2/3D geometric and ACIS entities such as VERTEX, EDGE, FACE, PLANE, line etc.

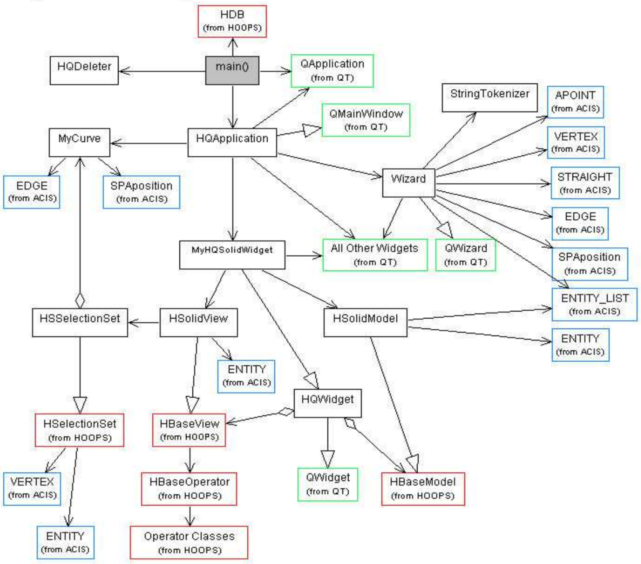
The Qt GUI Toolkit is a collection of GUI components for C++ that are platform independent.

The GAlib is a collection of C++ libraries and C/C++ header files that are used in the automated optimisation of the geometry.

And finally, the HOOPS Framework is a collection of dynamically linked libraries and C/C++ header files that are used for purposes of visualising 2/3D objects. Amongst many other formats, the HOOPS Framework supports the ACIS geometry and the ACIS file format, so the two can collaborate without major difficulties. Also, the HOOPS Framework could easily be plugged into the Qt GUI via its widget, which is created by extending the Qt's major object, `QWidget`.

The figure bellow shows how objects from all three libraries coexist and make up the software. The objects in the blue boxes are pure ACIS objects from the ACIS Modeller Library. The objects in the green boxes are Qt objects and the objects in the red boxes are objects from the HOOPS Framework. The black boxes mark objects that are application-specific objects derived from either of the libraries. The derivation is shown by the white arrows.

Object Relationships



A detailed description and an explanation of objects are given in the Programming Guide's section 3, Component Object Relationships. Also, the proper class diagram of the software could be found on the same page.

The software could be easily split in two parts: the front end or GUI and the back end. The back end comprises of system calls to GAMBIT and FLUENT. The calls could be edited, so instead of running GAMBIT and FLUENT on a local machine, they could be run off the Linux cluster and use the advantages of parallel computing to improve on efficiency. Also, the genetic algorithm could be easily extended to be used in a parallel, making the software created during the project, very open to parallelisation.

Results

The three months long project has yielded a piece of software that is able to demonstrate how several different engineering tools could be integrated in one integrated design environment, helping engineers to solve common problems and probably revolutionising the engineering product development cycle in the near future. The software had the goal of bringing together geometry definition, meshing and analysis in a single, easy to use design environment.

Geometry Definition

The software meets the software requirements in their full extent in this area. The implemented functionality allows a user to specify where a pipe is to be located and where the inlet and the outlet planes are. The user is, also, allowed to specify the shape and route of the pipe and if modifications to the pipe are necessary, the user is given a full control over it. At this stage, the software is limited to edits via the input boxes only rather than via Click and Drag features, which can be inconvenient for the user. However, apart from the inconvenience, none of the required functionality is sacrificed. On the contrary, more precise position of the control points could be set with the input boxes.

Meshing

The software requirement was to mesh the geometry on a click of a button, which is met. The user is also, able to specify all mesh parameters giving a full control over this operation to the user. The limitation of the software is that the meshing operation is platform dependant and requires presence of GAMBIT on the system. Meaning that a path to the GAMBIT executable has to be in the system variable $PATH.

Analysis

Like the meshing part, this one depends on the external software as well. It relies on the FLUENT CFD Solver, which is also invoked via a platform dependant call. The implementation and limitations of this part are the same as the meshing part. Full control is given over the parameters used in an analysis and FLUENT has to be in the system variable $PATH.

Visualisation

The user is able to zoom at different zoom levels, manipulate camera views and make selections of geometries, which meets the software requirements. Having the visualisation operations implemented, the user is able to create correlation between a shape presented in the display area and an output (pressure loss) presented by FLUENT, which is the major purpose of the software.

Discussion:

Three months ago, the project scope was to create a software tool that would help engineers solve a common optimising problem. A tool that would compare a series of geometrical shapes to obtain the one that has the best score. As the project progressed and the benefits were obvious, it was decided to change the initial requirements and expand the project to include some other features, which would demonstrate potential benefits of the Virtual Integrated Design Environment (VIDE). The VIDE is a long term goal. The piece of software created during this project, together with other software pieces already created or under development, will be the stepping stone for the further development of the VIDE.

Apart from the demonstrational purposes of the software, the software will allow its users to get more insight into effects changes on a pipe shape have on a 2D internal turbulent flow. The software could be applied to a range of 2D internal pipe flow problems, which require evaluation of a pressure drop between the pipe's inlet and outlet. In essence, the software could be used as a fancy pressure drop calculator. The visualisation side of the software could also be used as an ACIS file viewer because the software is capable to view and manipulate views of any 2/3D object saved in ACIS file format. The software in its current shape could also find use in some limited optimisation problems. It allows the users to find the optimal shape but at much slower pace than the variation of the software, which involves the genetic algorithm (the future releases may implement a proper high speed optimisation module).

The (initial) optimisation code written in the project and idea of having the optimisation tool, in general, has not been abandoned completely, but it is used in a different aspect and with a different purpose. It is going to be integrated in case studies, which are going to be submitted in and presented at two conferences. The first conference is "2004 ASME Heat Transfer/Fluids Engineering Summer Conference, Charlotte, North Carolina, USA, July 11-15, 2004" and the second one is "CTAC 2004: 12th Biennial Computation Techniques and Applications Conference Workshops. Melbourne, Australia, Sep 27 - October 1, 2004" The case studies are going to present application of the code in a smooth pipe turbulent airflow and "wider applications of the code to more complex routing fluid flow problems and non-routing based applications" [the Abstract of the paper by Jarrod Sinclair].

The project has opened a new frontier that may include artificial intelligence in the engineering product development cycle. Eliminating human supervision from repetitive tasks will create appreciable savings and increase the efficiency. The current procedure involves a lot of long waiting periods, which may suppress a designer's creativity and a flow of new ideas/improvements.

Also, the project may initiate a new research that would aim to find better correlations between the pipe shape and the pressure drop than the existent correlations. Thus, the software developed could be a very useful tool during such research.

Conclusion

The optimisation tool that has been developed is able to create geometry, edit/delete geometry, mesh geometry by incorporating GAMBIT and perform a CFD analysis with a help of FLUENT, performing actions necessary for an optimisation of a 2D pipe. The tool includes a display area, allowing visualisation of geometry used in the analysis/optimisation. Results produced by the tool allow a user to draw conclusions on how shape influences pressure loss and/or what shape is the most optimal for use.

Therefore, the created piece of software fully answers the research objective.

References

Jonathan Corney, Theodore Lim, "3D Modeling with ACIS", 2nd Edition, Saxe-Coburg Publication, 2001, ISBN 1-874672-14-8

ACIS Modeller Documentation and Help, downloaded with the package, [http://www.spatial.com]

HOOPS Framework Documentation and Help, downloaded with the package, [http://www.spatial.com]

GAlib Documentation and Help, downloaded with the package, [http://lancet.mit.edu/ga]

Qt GUI Toolkit Documentation and Help, part of Qt Development Package [http://www.trolltech.com]

GAMBIT Documentation and Help [http://www.fluent.com]

FLUENT Documentation and Help [http://www.fluent.com]

Favourite Search Engine: http://www.google.com

Acis Alliance Mailing List: http://lists.cs.columbia.edu/mailman/listinfo/acis-alliance