16th Australasian Fluid Mechanics Conference
Crown Plaza, Gold Coast, Australia
2-7 December 2007

Derivative Free Global Optimisation of CFD Simulations

R. C. Morgans[1], C. J. Doolan [1] and D. W. Stephens[2]

[1]School of Mechanical Engineering
Adelaide University, South Australia, 5005 AUSTRALIA

[2]CSIRO Minerals
Box 312, Clayton South, Victoria, 3169 AUSTRALIA

Abstract

This work reports on the use of numerical optimisation techniques to optimise objective functions calculated by Computational Fluid Dynamics (CFD) simulations. Two example applications are described, the first being the shape optimisation of a low speed wind tunnel contraction. A potential flow and viscous flow solver have been coupled to produce a robust computational tool, with the contraction shape defined by a two parameter Bézier curve. The second application is a simplified test case with a known minimum calculated using a commercial CFD code.

For the optimisation of complex CFD simulations, it is sometimes advantageous to use an efficient derivative free global optimisation algorithm because of potentially long simulation times, the objective function may contain multiple local minima and it is often difficult to evaluate analytical or numerical gradients. The Efficient Global Optimisation (EGO) algorithm sequentially samples results from an expensive calculation, does not require derivative information, uses an inexpensive surrogate to search for a global optimum, and is used in this current work.

For both applications, the EGO algorithm is able to efficiently and robustly find a global optimum that satisfies any constraints.

Introduction

Developments in Computational Fluid Dynamics (CFD), and widespread access through commercial and open source software, have allowed engineers and designers to improve product or process quality through simulation.

If a suitable measure of quality of the product or process can be defined, then mathematical optimisation techniques can potentially automate the design process. This measure, or "objective function" is used by the optimisation routine to systematically change the inputs to the simulation until the "best" objective has been achieved.

CFD calculations typically involve the solution of coupled nonlinear equations, which can limit the uptake of optimisation for a number of reasons:

Efficiency. The equations can exhibit behaviour over a large range of time and length scales. As such, they can be very computationally expensive, even when simplifying assumptions such as Reynolds averaged turbulence models are used. Optimisation methods that minimise the number of objective function calculations are required.

Non smooth behaviour. In some cases, CFD solutions may not exhibit "smooth" behaviour, in that derivatives with respect to input variable may not exist, either because of discontinuities in the underlying physics, or discontinuities introduced by the numerical method. Many optimisation methods require derivative information, hence are unsuitable in this case.

Local Minima. For some problems, there exist many locally optimal solutions where there is no possible improvement in the objective function in the neighbourhood of the solution. Many optimisation methods can only find local minima, and global methods are preferred.

Robustness. The non-linearity also makes the equations difficult to solve automatically for all conditions. If a solution fails, (often after a long computation), then methods that can deal with these "hidden constraints" are required.

The Efficient Global Optimisation (EGO) [6] algorithm sequentially samples results from an expensive calculation, does not require derivative information, uses an inexpensive surrogate to search for a global optimum. This makes it ideal for optimisation of expensive CFD simulations, and its application is investigated in this paper.

We present application of the EGO algorithm to two example problems, the shape optimisation of a low speed wind tunnel contraction and the minimisation of the pressure drop for flow between parallel plates with an analytic loss function. The structure of the paper is as follows: the EGO optimisation method is described; the problems and results are specified and presented; and finally conclusions are drawn and future work discussed.

Efficient Global Optimisation

EGO is a surrogate (or meta) modeling technique, where the expensive cost function evaluation is replaced with a model that is both cheap to construct and evaluate. The work of Jones et. al. [6], refined by Sasena [11] into the algorithm superEGO, has developed an efficient surrogate method for global optimization, called Efficient Global Optimization (EGO, which was originally called SPACE in Schonlau [12]).

This technique uses a Kriging [7] surrogate model to predict the values of the objective function at a few, sparsely distributed sample points. These sample points are generally chosen by a space filling sampling method. Kriging, developed in the geostatistics and spatial statistics fields, models the variation of the unknown function as a constant value plus the variation of a normally distributed stochastic variable. It is essentially a method of interpolation between known points that gives a mean prediction, $\hat{y}(x)$, in addition to a measure of the variability of the prediction, $s(x)$, the estimated standard deviation. Another suitable global optimisation technique such as the DIRECT method [3] is then employed to solve an auxiliary problem to find the next best place to sample for a minimum primary objective function. The secondary objective function used to solve the auxiliary problem in this application is the Expected Improvement ($\mathbf{E}[I]$) objective function. The improvement function (I) is defined as the improvement of the current prediction, $\hat{y}(x)$, at point x over the minimum value of the current set of samples, y_{min}, i.e.

$$I = \max\left(y_{min} - \hat{y}(x), 0\right) \tag{1}$$

1. An initial set of input parameters is selected.
2. The true objective function is evaluated for all new members of the set.
3. A Kriging surrogate model is fitted to the values of the objective function.
4. The expected improvement objective function, calculated using values from the computationally inexpensive Kriging model, is minimised using any suitable global optimisation method.
5. The result of the minimisation (the next input parameters most likely to improve the true objective function) is added to the set.
6. The process repeats from step 2 until a predetermined number of iterations is reached.

Figure 1: Efficient Global Optimisation algorithm

The expected improvement, defined as the expectation of the improvement, is given by

$$\mathbf{E}[I] = (y_{min} - \hat{y}(x))\,\mathrm{CDF}\left(\frac{y_{min} - \hat{y}(x)}{s(x)}\right) + s(x)\,\mathrm{PDF}\left(\frac{y_{min} - \hat{y}(x)}{s(x)}\right) \quad (2)$$

where CDF is the standard normal cumulative density function, and PDF is the standard normal probability density function. The point at which the value of the expected improvement is maximised gives the best point at which to calculate the true objective function. The Expected Improvement is constructed to search for both local and global minima [12, 6]. The surrogate model is then updated to include the newest sampled point, and the operation repeated until the sampling point does not change and the global minimum of the objective function has been found. An overview of the algorithm appears in Figure 1.

An implementation of EGO developed in Morgans [9] will be used in this work, with simple penalty constraints.

Contraction shape optimisation

The design of low speed wind tunnel contractions is extremely important for the provision of a quality test flow in the working section. Contractions increase the mean velocity of the test flow so that the flow quality improving devices (screens, honeycomb) act in a low velocity environment where pressure losses are reduced. They also reduce the mean and fluctuating velocity variations to a smaller fraction of the average velocity [8]. The design of a contraction requires the flow to remain attached along the length of the contraction and to minimise the outlet boundary layer height as well as outlet flow non-uniformity.

To compute the flow within a wind tunnel contraction, instead of solving the potentially time consuming laminar Navier-Stokes equations, a classical approach is taken. The inviscid flow within the contraction is calculated using a finite volume three-dimensional potential flow solver (OpenFoam [5]). This solver returns the velocity in the contraction $\mathbf{U}$. The velocity distribution at the walls is then used with an integral method (Thwaites' method) to generate laminar boundary layer solutions. The boundary layer solver returns the Reynolds number based on momentum thickness Re_θ and skin friction coefficient C_f at critical locations on the surface. The minimum value of $C_f(x)$ along the length of the contraction surface must be greater than zero for the flow to remain attached. The outlet flow non-uniformity is measured buy the standard deviation of the axial velocity at the exit plane. A complete description of the contraction, the numerical method used and its validation can be found in Doolan and Morgans, 2007 [2].

The shape of the contraction is prescribed by a Bézier curve, a flexible shape defined by a third order polynomial. It is specified by two vectors, with the curve tangent to the start of each vector, and the "strength" of attachment to the vector determined by its length [10]. Figure 2 shows a range of contraction shapes constructed by controlling the length of the vectors using control parameters. The parameters are limited to vary between 0 and 1, and are defined as the length of the vector normalised by the contraction length. Parameter a controls the inlet curvature using the upper vector and b controls the outlet curvature using the lower vector.

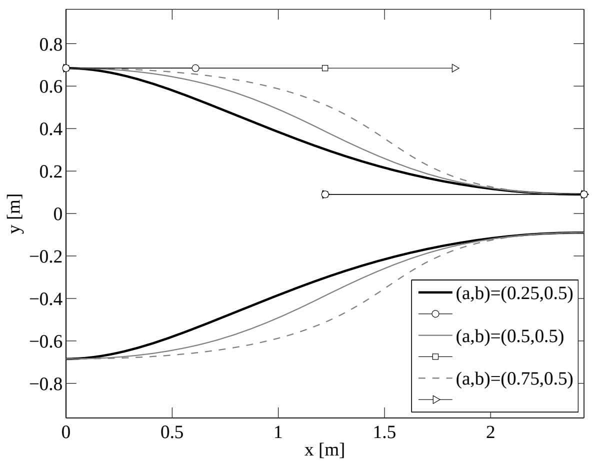

Figure 2: Contractions shapes Bézier geometry

The objective function must represent the goals of the optimisation, which is to minimise the exit plane boundary layer height, keep the flow attached and keep exit plane flow uniformity to a reasonable value. These criteria can be described mathematically as an objective function.

$$\min Re_\theta(x = L) \quad (3)$$

with constraints

$$\min C_f(x) > 0 \quad (4)$$
$$\mathrm{std}(U_x(x = L)) < 1\,\% \quad (5)$$

For some extreme values of the control parameters the computation of the contraction fails. This is caused by poor mesh quality from the automated meshing procedure. This *hidden* constraint is dealt with implicitly through modifying the objective function. If the computation is unable complete, a value of 600 is returned for Re_θ. The maximum value of Re_θ for completed calculations is also limited to 600, a value well above previous standard designs.

The constraints are enforced using a penalty method. This adds a term to the objective function that measures the degree to which any constraints are violated.

$$\hat{f}(x) = f(x) + \sum_{i=1}^{n} p_i \min(0, c_i) \quad (6)$$

where $f(x)$ is the true objective function (Equation 3), $\hat{f}(x)$ is the penalised objective function (equal to the true objective function in the feasible region where all constraints are satisfied), n is number of constraints, p_i is the penalty parameter for constraint i and c_i is the measure of constraint violation. The penalty parameter is set to 600 for both constraints, which was found to be large enough to penalise violated constraints. If the penalty parameter is too large, then there may be difficulty in fitting a surrogate to the sampled data points. For the first constraint, the measure is zero if Equation 5 is true, and one if it is

Table 1: Comparison of optimization performance with Bell and Mehta profile.

Contraction shape	Re_θ	Exit Plane Uniformity
Optimized	431.9	1%
Bell and Mehta	484.1	0.6%

not. The second measure is given as

$$\log_{10}\left(\mathrm{std}\left(U_x\left(x=L\right)\right)\right)-2\ . \tag{7}$$

Results

An optimisation of the contraction shape was performed using EGO. Because the method of initial point selection is random, the EGO algorithm is non-deterministic. The EGO method was run 3 times, and was able to find reasonable solutions in 33, 43 and 98 function evaluations. The calculation time of an individual calculation is quite small in this case, around 1 minute on a Intel Core Duo T2400 processor. The large variability in function evaluations is probably due to difficulty in fitting a Kriging model to the objective function including the penalty function.

In Doolan and Morgans, 2007 [2] a comparison of the performance of EGO was made with other methods. Sequential Quadratic Programming, a deterministic local optimiser using finite difference gradients, was found to perform better than EGO provided a good initial starting condition was known. DIRECT [3], a deterministic derivative free global optimiser, was able to robustly find solutions at the expense of efficiency.

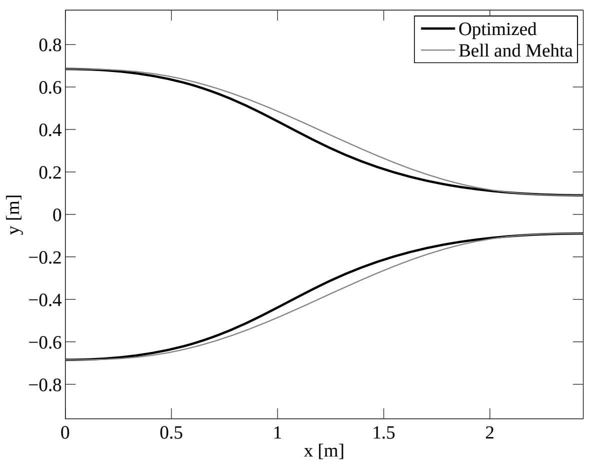

Figure 3: Comparison of optimized profile with that of Bell and Mehta.

Figure 3 compares the shape of the optimised contraction profile with 5th order polynomial used by Bell and Mehta [1], and Table 1 compares the performance of the two designs. The Bell and Mehta design has thicker boundary layers, but more exit plane uniformity. The optimised design trades off the exit plane uniformity for thinner boundary layers.

One-dimensional flow with analytic loss function

In order to gain confidence in the accuracy of commercial CFD codes it is often necessary to study simple flow problems with known solutions. In this way issues such as mesh requirements, choice of solver parameters and implementation of user-defined subroutines may be investigated thoroughly before the code is applied to more complex problems in which the solution is unknown. This step is particularly important in optimisation problems as the solutions can often be complex and non-intuitive.

Consider the case of steady state, incompressible flow in a horizontal channel of constant cross section. It is assumed the one-dimensional flow occurs so the all quantities have uniform (i.e., constant) profiles at any x position. Under these assumptions the continuity equation simplifies to

$$\frac{du}{dx}=0 \tag{8}$$

and the momentum equation to

$$\rho\frac{d\left(uu\right)}{dx}=-\frac{dp}{dx}+2\mu\frac{du}{dx}+S_m \tag{9}$$

where p, u, S_m, μ, and ρ are the pressure, velocity, momentum source, fluid viscosity and density respectively.

A source/sink of momentum was used in whole computational domain to include the analytical pressure drop relationship. The analytical pressure drop per unit length had no physical meaning and was selected solely as a test function for the optimisation algorithm and was given by

$$\frac{dp}{dx}_{analytical}=\sin\left(4x\right)+\sin\left(\frac{8}{3}x\right)+4x^{0.24} \tag{10}$$

This function contains multiple local minima and requires a global optimiser to find a robust solution, and is shown in Figure 5.

The source term applied in equation 9 was evaluated using the following expression

$$S_m=-\frac{dp}{dx}_{analytical} \tag{11}$$

Numerical method

The steady state equations (8 to 11) were solved using ANSYS-CFX Version 11 [4] for the one-dimensional flow test problem using the geometry shown shown in Figure 4. The parallel plates are 10 m long with a plate separation of 0.05 m. A uniform hexahedral mesh of 2000 cells in the x direction, 32 cells in the y direction and 1 cell in the z direction was used for all simulations. One-dimensional flow can be implemented on a two-dimensional geometry by applying a boundary condition of zero shear stress at the channel walls. Uniform Dirichlet boundary conditions were imposed at the inlet and a Neumann boundary condition was imposed at the outlet. The fluid used for the simulations was water with a density of 1000 kg/m^3 and viscosity of 1×10^{-3} kg/ms. Convergence testing was applied to all variables with a convergence target on the normalised residuals of 1×10^{-5}. An initial guess of zero was applied to all variables at the start of the simulations. As ANSYS-CFX uses a false time-stepping method when solving steady state problems a false timestep of 10000 s was used. The calculation time for this simulation is approximately 5 minutes using a Intel Pentium 4 531 processor.

The objective function was to minimise the pressure drop per metre between the velocity bounds of 0.05 m/s and 5 m/s.

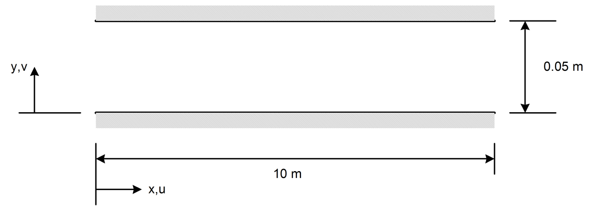

Figure 4: Channel geometry used for simulations.

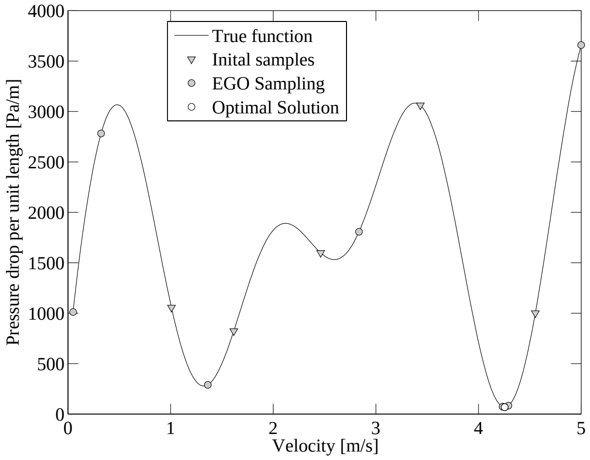

Figure 5: Optimisation results for analytic loss function.

Results

The optimisation on the objective function was tested using EGO. The EGO method was run using an initial sampling of 5 points to build the surrogate (the shaded triangles in Figure 5. A further 7 expensive objective function evaluations (the shaded circles in Figure 5 were required to find the minimum pressure drop per metre (the white circle in Figure 5 at a flow velocity of 4.25 m/s within a 1×10^{-4} m/s accuracy compared to the analytical solution.

The results show repeated sampling around the global minimum, and that the EGO sampling has managed to find both local minima that are close in value to the global minimum (at flow velocities of 0.05 m/s and 1.5 m/s) but avoid sampling at the third local minimum (at a flow velocities of 3.7 m/s). Because the method of initial point selection is random, the EGO algorithm is non-deterministic. The EGO method was run 3 times (each with an initial sampling of 5 points), and was able to find reasonable solutions in 12, 13 and 16 function evaluations.

Conclusions and future work

For the contraction optimisation, the EGO algorithm had difficulty with the penalty constraint method. Improvements to the algorithm to implement constraints in different ways could result in improved performance [11]. For this particular problem finite difference gradients were successful, SQP was able to find a solution from a good initial solution (which is generally known by the design) and this method would be recommended for industrial practice. However this problem has proved useful for testing the EGO algorithm because of its relatively short computation time.

The one-dimensional flow case with the analytic loss effectively "filters" a known analytic solution through a numerical discretisation procedure. It also allows the interface between the solver and the optimisation routine to be set up with a known solution. The EGO algorithm is able to find the unconstrained global minimum of the analytic function embedded in the ANSYS-CFX commercial solver with relatively few function evaluations. This should extend to multiple dimensions [9].

There are a number of improvements to be made to the EGO algorithm for use on expensive objective functions. Currently the algorithm is run for a fixed number of search iterations. A robust stopping criteria would be useful. Most expensive optimisations do not need exact optima, and the search could proceed on a grid of defined tolerance.

Overall the EGO algorithm has been found to perform well on simple (computationally inexpensive) problems. It needs to be tested on very computationally expensive problems, which the authors intend to do in the near future.

Acknowledgements

This work has been supported by the Sir Ross and Sir Keith Smith Fund, and CSIRO Materials and Manufacturing Technology.

References

[1] Bell, J. and Mehta, R., Contraction design for small low speed wind tunnels, Technical Report NASA-CR-177488, NASA, 1988.

[2] Doolan, C. J. and Morgans, R. C., Numerical evaluation and optimization of low speed wind tunnel contractions, in *18th AIAA CFD Conference proceedings*, 2007.

[3] Finkel, D. E., *DIRECT Optimisation Algorithm User Guide*, Center for Research in Scientific Computation, North Carolina State University, 2003.

[4] Inc, A., *ANSYS-CFX Version 11.0 Documentation*, Pittsburgh, Pennsylvania, United States of America, 2007.

[5] Jasak, H., Weller, H. and Nordin, N., In-cylinder CFD simulation using a C++ object orientated toolkit, Technical Report SAE Technical Paper 2004-01-0110, Society of Automotive Engineers, 2004.

[6] Jones, D. R., Schonlau, M. and Welch, W. J., Efficient global optmization of expensive black-box functions, *Journal of Global Optimization*, **13**, 1998, 455–492.

[7] Lophaven, S. N., Nielsen, H. B. and Søndergaard, J., DACE - a Matlab Kriging toolbox, version 2.0, Technical Report IMM-REP-2002-12, Informatics and Mathematical Modelling, Technical University of Denmark, 2002.

[8] Mehta, R. and Bradshaw, P., Design rules for small low speed wind tunnels, *Aeronautical Journal*, **73**, 1979, 443–449.

[9] Morgans, R. C., *Optimisation techniques for horn loaded loudspeakers*, Ph.D. thesis, University of Adelaide, 2004.

[10] Plant, D., The math behind the Bézier curve, Web page, 1996.

[11] Sasena, M. J., *Flexibility and Efficiency Enhancements for Constrained Global Design Optimization with Kriging Approximations*, Ph.D. thesis, University of Michigan, 2002.

[12] Schonlau, M., *Computer experiments and global optimization*, Ph.D. thesis, University of Waterloo, 1997.