A 3D Automatic Optimization Strategy for Design of Centrifugal Compressor Impeller Blades

K.F.C. Yiu and M. Zangeneh
Department of Mechanical Engineering, University College London, London WC1E 7JE, United Kingdom

Abstract

In deriving automatic numerical optimization algorithms for aerodynamic applications, the design parameters are usually chosen to be the unknown airfoil/blade profiles. However, there are certain advantages in using the pressure/velocity distribution as the design variable in some applications; the designed distribution can then be used in a 3D inverse design method to generate the actual profile shape. Here, this approach will be addressed. Two methods are used to parametrize the circulation distribution for compressor blades. Dawes's code is used to predict the viscous effect. An automatic optimization algorithm is developed to minimize the loss with respect to the design parameters.

Nomenclature

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>( r_{\theta} )</td>
<td>the circulation distribution</td>
</tr>
<tr>
<td>( p^+ )</td>
<td>pressure on the upper surface of blade</td>
</tr>
<tr>
<td>( p^- )</td>
<td>pressure on the lower surface of blade</td>
</tr>
<tr>
<td>( M )</td>
<td>the Mach number</td>
</tr>
<tr>
<td>( m )</td>
<td>the meridional coordinate</td>
</tr>
<tr>
<td>( s )</td>
<td>the non-dimensionalized ( m ) by chord length</td>
</tr>
<tr>
<td>( N_b )</td>
<td>number of blades</td>
</tr>
<tr>
<td>( W_{rel} )</td>
<td>relative velocity on blade surface</td>
</tr>
<tr>
<td>( M \bar{u} )</td>
<td>Mach no. based on stagnation temperature and impeller tip speed</td>
</tr>
</tbody>
</table>

Introduction

Since the development of computational fluid dynamics (CFD), a lot of efforts have been devoted to the derivation of automated methodology. The general objectives that such efforts are trying to accomplish are to decrease human involvements in searching for improved designs; it is hoped that designers will have a better tool than the use of cut-and-try approaches.

One popular approach to introduce automatic algorithms is to use numerical optimization methods. The design problem is formulated as a constrained optimization problem and standard techniques, such as the gradient methods, can be applied. In formulating the design problem, the design variables are usually the unknown shapes parametrized in certain ways. The first attempt is to restrict the changes to a relatively small number of smooth geometric modes and add to the original profile shape as perturbations. This approach has been applied by (Hicks and Henne, 1977) and (Vanderplaats, 1979). Gradually, the use of spline functions to represent the profile geometry directly has gained popularity because of their smoothness and flexibility in making localized modifications. The design variables are usually the control points of the spline curves. Because of the compact support property, the designers can modify the geometry in certain important regions only while keeping a smooth curvature variation for the rest of the profile. In this way, the dimension of the problem is greatly reduced which makes three-dimensional optimization designs possible. This approach has been applied by (Sadrehaghghi et al., 1995), (Burgreen and Baysal, 1994) for two-dimensional designs and by (Burgreen and Baysal, 1996) (Baysal et al., 1997) for three-dimensional designs.

Another approach to develop automatic design algorithms is the inverse method, which solves the governing partial differential equations inversely. Several successful inverse methods have been developed today, such as...
as (Drela, 1985), (Leonard and Van Den Braembusche, 1991) and (Zangeneh, 1991), in which a target pressure/velocity distribution is imposed as a boundary condition to the partial differential equations. One of the drawbacks of using an inverse method is the difficulty in getting a good target distribution. Because of this, inverse methods are kinds of semi-automated design methods which leaves the designers the vital task to input a suitable distribution.

Although the profile parametrization seems to be the logical way to formulate the optimization problem, many experienced designers may find it easier to handle the pressure/velocity distributions directly instead of the actual profile geometry, since distributions usually give more insight to the problem being solved than the shape itself. In addition, as pointed out by (van Egmond, 1989), by judging from the data published in (Aidala et al., 1983), relatively simple pressure/velocity distribution modifications lead to complicated variations in the profile geometry. It indicates that it may be advantageous to optimize the distribution directly by numerical optimization; this may lead to considerable savings in computations for some cases. This approach is first explored by (van Egmond, 1989) (van den Dam et al., 1990) who parametrized the pressure distributions for two-dimensional airfoil sections based on the flow characteristics in different sections. However, applications in a turbomachinery context for three-dimensional blade design with the use of B-splines and the full Navier-Stokes equations for the viscous effects have not been explored yet. This will be addressed here.

In this paper, an automatic algorithm for the design of compressor impeller blades is developed. The design variable is the circulation (or swirl velocity) distribution for compressor blades. Two different ways of parametrizing the distribution are applied and compared. The first method is based on a three-segment description of the loading distribution used in (Zangeneh et al., 1996); the circulation is then sought by integrating the loading distribution. The second method is to represent the circulation distribution directly using a cubic-spline curve bounded by a control polygon. Once the circulation is defined, the inverse method developed by (Zangeneh, 1991) is used to calculate the blade wrap angle distribution for the whole three-dimensional blade row. With this blade geometry, Dawes's code (developed by (Dawes, 1988) and (Walker and Dawes, 1990)), which solves the full three-dimensional Navier-Stokes equations in turbomachinery blade rows, is evoked to predict the viscous effects. The loss is used as the objective function in the optimization algorithm. One important factor of this approach is that work (and therefore blade loading) is kept constant throughout the optimization process, since a reduction in blade loading automatically results in a reduction in loss. Although work is usually changed inevitably during optimization in profile design such as (Trigg et al., 1997), it can be fixed easily here by specifying the loading distribution to satisfy the constant work constraint. A numerical example is used to demonstrate how separation can be significantly reduced using this method.

The circulation distribution

The approach used here is to find the optimum swirl velocity (denoted by \( \tau \)) distribution at both the hub and shroud, and then interpolate linearly between these values to obtain the overall distribution on the meridional plane. This three-dimensional distribution can then be used as the input to the inverse method (Zangeneh, 1991). At the leading and trailing edges, the value of \( \tau \) are obtained from Euler's pump equation. Furthermore, since the jump in pressure across the blades is given by

\[
p^+ - p^- = \frac{2\pi}{N_b} (\rho \mathbf{W}_b \cdot \nabla \tau),
\]

the derivative of \( \tau \) in the meridional direction must be set to zero at the trailing edge in order to satisfy the Kutta condition; it is often set to zero as well at the leading edge to fulfill the no-incidence condition.

There are several ways to define the circulation distribution complying with the above criteria. As equation (1) suggests that the pressure distribution on the blades is directly related to the loading distribution (meridional derivative of \( \tau \)), the first approach is to use a smooth loading distribution and integrate to get \( \tau \); this will ensure a smooth pressure distribution. A typical parametrization applied here is depicted in Fig. 1. The distribution is parabolic up to the mesh point NC, then a linear variation between mesh points NC and ND with a specified slope; This is followed by another parabolic segment which reduces the loading to zero at the trailing edge (mesh point NB) so that the Kutta condition is satisfied. The \( \partial \tau / \partial m \) distribution is set up in such a way that the area under it is fixed, thereby ensuring fixed specific work design. More details are given in (Zangeneh et al., 1996). There are a total of 8 parameters that will vary the overall \( \tau \) distribution, namely \( (\delta_{\text{hub}}, \theta_{\text{hub}}, N_{\text{C_{hub}}, N_{\text{D_{hub}}}}) \) and \( (\delta_{\text{tip}}, \theta_{\text{tip}}, N_{\text{C_{tip}}, N_{\text{D_{tip}}}}) \). It is important to point out that by varying \( NC \) and \( ND \), the loading distribution changes from fore-loaded to evenly-loaded and then to rear-loaded easily which is a fairly comprehensive set. In practice, bounds are imposed on the variables for both the hub and shroud as

\[
0 \leq \delta \leq 2, -78^\circ \leq \theta \leq 78^\circ, 5 \leq NC \leq ND \leq M - 5,
\]

where \( M \) is the number of meridional mesh points.

The second way is to represent \( \tau \) by a cubic B-spline curve which has continuous second order derivatives and the variation diminishing property (de Boor, 1978). Using a control polygon with coordinates \( \{b_i, i = 0, \ldots, n\} \), the
cubic B-spline curve is defined as

\[ P(t) = \sum_{i=1}^{n} b_i B_i(t) \]  \hspace{1cm} (3)

where \{B_i(t)\} are the cubic B-spline basis functions with the set of knots

\[ N = (t_0, \cdots, t_n). \]  \hspace{1cm} (4)

Because the basis functions have compact local support, changing a point in the corresponding control polygon changes the course of the curve only locally and the influence of each control point can be pinpointed precisely. It is also possible to introduce additional control points easily; therefore, less control points may be used initially to search for improved designs.

In particular, if the set of knots (4) is chosen to be the integer set

\[ N = (0, \cdots, n), \]  \hspace{1cm} (5)

(3) becomes the uniform B-spline curve which can be simplified to (Coons, 1974)

\[ P(t) = \frac{1}{6} \begin{pmatrix} t^3 \\ t^2 \\ t \\ 1 \end{pmatrix}^T \begin{pmatrix} 1 & 3 & 3 & 1 \\ 3 & 6 & 3 & 0 \\ 3 & 0 & 3 & 0 \\ 1 & 4 & 1 & 0 \end{pmatrix} \begin{pmatrix} b_i \\ b_{i+1} \\ b_{i+2} \\ b_{i+3} \end{pmatrix}, \]  \hspace{1cm} (6)

where \( t \in [0, 1] \).

In order to fix the incidence condition at the leading edge and to satisfy the Kutta condition at the trailing edge, the second and the penultimate control points are fixed to give the required slope. Since \( r\theta_g \) are also given at the leading and trailing edge by the Euler's pump equation, two extra control points are used, one at each end extended by an equal distance and an equal slope to the respective segment. A typical \( r\theta_g \) together with its control polygon is shown in Fig. 2. The loading distribution can be sought by differentiating \( r\theta_g \).

In practice, although the cubic B-spline ensures the continuity and differentiability of the loading distribution, it does not control the number of inflection points in \( r\theta_g \) and the loading distribution might have excessive numbers of turning points which in turn gives rise to an impractical blade wrap angle distribution. In order to avoid this, the total curvature of the circulation distribution is controlled via

\[ \int_{\gamma_{in}} |r\theta_g''(s)| ds < \epsilon_k, \]  \hspace{1cm} (7)

where \( \epsilon_k \) is a predefined constant.

**Optimization algorithms**

There are various optimization techniques developed today (Fletcher, 1987). In general, apart from the techniques being used in mathematical programming, most of the methods are based on search directions such that

\[ L(x^{(k)} + \alpha^{(k)} s^{(k)}) < L(x^{(k)}) \]  \hspace{1cm} (8)

in the \( k \)th iteration, where \( L \) denotes the objective function, \( s^{(k)} \) is the search direction and \( \alpha^{(k)} \) is the step size. Methods used to find a suitable \( s^{(k)} \) can usually be categorized into two types. The first category is the gradient methods or the quasi-Newton methods in which the gradient information is used to define the search directions. The second category is the no-derivative methods in which the search directions are selected based on intuition.
In order to choose the right method, typical characteristics of the problem are assessed which include:

1. The cost to evaluate the objective function.
2. The cost to acquire the gradient information and the accuracy of it.
3. The smoothness of the objective function, i.e. whether the function contains a lot of local minimums.
4. The number of unknown parameters in the problem.

In general, if the objective function is highly nonsmooth, gradient techniques usually perform poorly since they can easily be trapped by local minimums near to the initial guess. In addition, if the objective function evaluation is very expensive computationally, or if the number of unknown parameters is large, the adjoint equation, which can be derived either analytically (Pironneau, 1994) (Reuther and Jameson, 1995) (Baysal, 1997) or via automatic differentiations (Bischof et al., 1992), is required to calculate the gradient information in order to give a computationally viable method.

In the present application, since each evaluation of the objective function requires a solution of the Navier-Stokes equations, it is therefore very expensive to calculate. Also, as the present Navier-Stokes solver is used as a black-box solver, it is very difficult to derive the adjoint equation like (Burgreen and Baysal, 1994). If the gradients are calculated approximately via numerical differentiation, it will be very expensive and very inaccurate. This is particularly so here because one of the convergence criteria for the Navier-Stokes solver is to achieve a specified mass-flow rate within a certain percentage of fluctuation by varying the exit pressure (more details in the next section), the accuracy of numerical differentiations is therefore greatly impaired by narrow fluctuations in the solutions, while restricting the convergence criteria further will mean excessive and unnecessary computational effort.

Another crucial selection criterion is the number of local minimums, which is expected to be fairly large because of the nonlinearity of the problem. This is even worse in the present application as some of the unknowns in the three-segment parametrization are integers; such optimization problems are well-known to have numerous local minimums and the objective function is also non-differentiable. Thus, the gradient methods are not applicable.

Since the dimension of unknown parameters remains fairly low, and it has been reported earlier that best results have been obtained using a simple line search method rather than the use of a sophisticated gradient technique, under these circumstances, the alternative variable method is employed. The search directions are simply chosen to be

\[ x_i = e_i \quad i = 1, \ldots, n \]  

where \( e_i \) denote the Cartesian basis vectors and \( n \) denotes the number of parameters. Specifically, for \( i = 1, \ldots, n \), the point \( x^{(k+1)} \) is calculated from \( x^{(k)} \) by changing the \( k \)th component of \( x^{(k)} \) so that

\[ L(x^{(k+1)}) = \min_{a} L(x^{(k)} + \alpha s^{(k)}) \]  

where \( \alpha = (e_\alpha, 2e_\alpha, \cdots) \cup (-e_\alpha, -2e_\alpha, \cdots) \); the search is stopped only if

\[ |L(x^{(k)} + \alpha s^{(k)}) - L(x^{(k)})| > \epsilon_1 \]  

where \( \epsilon_1 \) is a pre-defined tolerance, or if the constraint defined in (2) or (7) are violated. The searches used in (11) resemble the spirit of the continuation method in solving a system of nonlinear equations when the starting guess is far from the final solution. Moreover, since the search will not be terminated until (12) or (2)/(7) are violated, a large \( \epsilon_1 \) will allow \( L(\alpha) \) to increase as well as decrease and this helps to by-pass those local minimums around \( x^{(k)} \).

**Numerical results**

The algorithm was applied to the design of a high specific speed industrial centrifugal compressor. The objective is to minimize the loss. An initial blade is chosen which has performance of the same level as a state-of-art compressor impeller. Both the design and the viscous analysis codes are executed as subroutines to the optimization main program. Viscous calculations were performed on a \((25\times81\times25)\) mesh, with 51 points inside the blade region and a non-uniform pitchwise and spanwise mesh spacing clustering towards the blades. Since the impeller is shrouded, tip leakage effects were not modelled. The \( r\theta \) is non-dimensionalized by the tip speed and the tip radius. The design conditions are as follows:
The final loading distributions at points C and D are then re-analysed using Dawes’s code but with a very small tolerance cm on the mass-flow rate convergence. This gives a final loss of 5.812% for point C and 5.416% for point D. The improvement on the final loss comes with no surprise as B-spline curve is a more flexible curve than the three-segment one which implies a larger solution space. Off-design performance of the design at point D is assessed for different mass-flow rates using the Navier-Stokes solver. The results are shown in Fig. 19 and 20. Clearly, there is quite a reasonable range of mass-flow rates with efficiency greater than 94% for the designed blade. Also, as mass-flow rates decrease, exit pressures exhibit a rising characteristic which lowers the chances of stalling.

**Conclusion**

In this paper, an automatic optimization algorithm has been developed which was applied to minimize the loss for the design of a high specific speed industrial centrifugal compressor. The usefulness of the algorithm has been demonstrated in this case by the reduction in separation in the three-dimensional flow field. The method acts as an interface between the inverse design method of (Zangeneh, 1991) and the Navier-Stokes solver, which makes use of the viscous solutions from the Navier-Stokes solver to improve the target circulation distribution input to the inverse method. Two methods of parametrizing the circulation distribution have been discussed. The first method provides a very smooth loading distribution and a very practical tool for design. The second method based on B-spline curves is a more general method which can be used for this optimization study, the normal thickness distribution corresponding to the original conventional impeller was specified; the inviscid version of the inverse design method was used; the meridional geometry, rotational speed, mass-flow rate were kept constant and only the \( r_{\theta} \) distribution was changed during optimization. In using Dawes’s code, the exit pressure is specified and the mass-flow rate is predicted. In order to achieve a certain mass-flow rate, an additional loop is constructed to iterate on this pressure until the desired mass-flow rate is reached. The iterations can be summarized as follows:

1. Specify an initial exit pressure and a target mass-flow rate \( M_f \).
2. Solve the Navier-Stokes equations for the mass-flow rate \( M_f \) via time stepping. The iterations stop if either a maximum number of iterations is reached or the loss converges to satisfy \( L^{(i+T)} - L^{(i)} < \varepsilon_i \) (a constant), where \( T \) is a pre-defined number of time steps.
3. If \( |M_f - M| > \varepsilon_m \) (a constant), update the exit pressure via secant iterations and goto 2.

The choice of the tolerance \( \varepsilon_m \) on the mass-flow rate convergence was studied very carefully by testing the optimization algorithm on a similar but slower speed problem with different tolerant values. It turns out that \( \varepsilon_m \) can be increased to about \( \pm 10\% \) of the target mass-flow rate without affecting the optimization result. It is therefore a practical choice during optimization searches, both giving accurate loss predictions and avoiding excessive iterations in the Navier-Stokes solver.

When the three-segment parametrization is used, the convergence history is depicted in Fig. 3. Only one complete cycle was executed so that the one-dimensional minimization (11) was executed once for each variable. The up and down movement in the convergence is because of the criterion (12) which allows for the loss going up as well as going down. However, after each one-dimensional minimization, only the result corresponding to the least loss is used to start the search for the next variable.

When the B-spline curve is used, the convergence history is depicted in Fig. 4. Three spline control points were initially used for both the hub and shroud \( r_{\theta} \) distribution, which were then switched to seven control points after one complete cycle had been executed. Clearly, most of the loss reductions come from the first cycle while the cycle of the seven control points makes no further reduction in this case.

For both parametrizations, the same initial loading distribution is used (Fig. 5). The loading distributions for points A,B,C,D marked in the convergence histories are compared between Fig. 5-8 while the corresponding Mach number distributions are shown between Fig. 9-12. The initial and the final \( r_{\theta} \) distribution for the spline optimization is also shown in Fig. 13. In Fig. 14, the wrap angles for points A,B,C,D are compared which shows a uniformly smaller hub wrap angles for the design using spline than the design using the three-segment parametrization. Judging from the velocity vectors produced by the 3D viscous code at the suction side of the meridional plane (Fig. 15-18), the same phenomenon can be concluded for both parametrizations. When the loading distribution is more evenly loaded as in point A, a clear separation is visible near to the shroud (Fig. 15). However, as the rear-loading at the hub increases as in point B, separation starts to decrease (Fig. 16). When the hub loading is skewed fully towards the rear as in points C and D, separation almost vanishes (Fig. 17 and 18). It is interesting to notice that changes come mainly from the hub loading distribution whereas the net effect occurs mostly near to the shroud region.

<table>
<thead>
<tr>
<th>No. of blades</th>
<th>17</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mu2</td>
<td>1.03</td>
</tr>
<tr>
<td>Tip diameter</td>
<td>0.5m</td>
</tr>
<tr>
<td>Rotational speed</td>
<td>13369rpm</td>
</tr>
<tr>
<td>Mass-flow rate</td>
<td>12.5kg/s</td>
</tr>
<tr>
<td>Exit axial width</td>
<td>0.0424m</td>
</tr>
<tr>
<td>Exit tangential velocity</td>
<td>240m/s</td>
</tr>
</tbody>
</table>

For this optimization study, the normal thickness distribution corresponding to the original conventional impeller was specified; the inviscid version of the inverse design method was used; the meridional geometry, rotational speed, mass-flow rate were kept constant and only the \( r_{\theta} \) distribution was changed during optimization. In using Dawes's code, the exit pressure is specified and the mass-flow rate is predicted. In order to achieve a certain mass-flow rate, an additional loop is constructed to iterate on this pressure until the desired mass-flow rate is reached. The iterations can be summarized as follows:

1. Specify an initial exit pressure and a target mass-flow rate \( M_f \).
2. Solve the Navier-Stokes equations for the mass-flow rate \( M_f \) via time stepping. The iterations stop if either a maximum number of iterations is reached or the loss converges to satisfy \( L^{(i+T)} - L^{(i)} < \varepsilon_i \) (a constant), where \( T \) is a pre-defined number of time steps.
3. If \( |M_f - M| > \varepsilon_m \) (a constant), update the exit pressure via secant iterations and goto 2.

The choice of the tolerance \( \varepsilon_m \) on the mass-flow rate convergence was studied very carefully by testing the optimization algorithm on a similar but slower speed problem with different tolerant values. It turns out that \( \varepsilon_m \) can be increased to about \( \pm 10\% \) of the target mass-flow rate without affecting the optimization result. It is therefore a practical choice during optimization searches, both giving accurate loss predictions and avoiding excessive iterations in the Navier-Stokes solver.

When the three-segment parametrization is used, the convergence history is depicted in Fig. 3. Only one complete cycle was executed so that the one-dimensional minimization (11) was executed once for each variable. The up and down movement in the convergence is because of the criterion (12) which allows for the loss going up as well as going down. However, after each one-dimensional minimization, only the result corresponding to the least loss is used to start the search for the next variable.

When the B-spline curve is used, the convergence history is depicted in Fig. 4. Three spline control points were initially used for both the hub and shroud \( r_{\theta} \) distribution, which were then switched to seven control points after one complete cycle had been executed. Clearly, most of the loss reductions come from the first cycle while the cycle of the seven control points makes no further reduction in this case.

For both parametrizations, the same initial loading distribution is used (Fig. 5). The loading distributions for points A,B,C,D marked in the convergence histories are compared between Fig. 5-8 while the corresponding Mach number distributions are shown between Fig. 9-12. The initial and the final \( r_{\theta} \) distribution for the spline optimization is also shown in Fig. 13. In Fig. 14, the wrap angles for points A,B,C,D are compared which shows a uniformly smaller hub wrap angles for the design using spline than the design using the three-segment parametrization. Judging from the velocity vectors produced by the 3D viscous code at the suction side of the meridional plane (Fig. 15-18), the same phenomenon can be concluded for both parametrizations. When the loading distribution is more evenly loaded as in point A, a clear separation is visible near to the shroud (Fig. 15). However, as the rear-loading at the hub increases as in point B, separation starts to decrease (Fig. 16). When the hub loading is skewed fully towards the rear as in points C and D, separation almost vanishes (Fig. 17 and 18). It is interesting to notice that changes come mainly from the hub loading distribution whereas the net effect occurs mostly near to the shroud region.
to supplement the first method in situations where the loss cannot be reduced by the first method. Work is currently underway to apply the algorithm for the suppression of secondary flows.

Acknowledgements

The authors wish to thank Ebara Research Co. Ltd. for supporting this work.

References


Fig. 3: The convergence of the 3 segment technique

Fig. 4: The convergence of the spline technique

Fig. 5: The loading distribution at point A

Fig. 6: The loading distribution at point B

Fig. 7: The loading distribution at point C

Fig. 8: The loading distribution at point D
Fig. 9: The Mach number distribution at point A

Fig. 10: The Mach number distribution at point B

Fig. 11: The Mach number distribution at point C

Fig. 12: The Mach number distribution at point D

Fig. 13: The comparison of circulation distributions for the spline optimization

Fig. 14: The comparison of wrap angles
Fig. 15: The suction side velocity at point A

Fig. 16: The suction side velocity at point B

Fig. 17: The suction side velocity at point C

Fig. 18: The suction side velocity at point D

Fig. 19: Efficiency against mass-flow rates

Fig. 20: Exit pressures against mass-flow rates