

Adjoint Error Correction for Integral Outputs

Michael B. Giles

*Oxford University Computing Laboratory, Oxford, UK.
giles@comlab.ox.ac.uk*

Niles A. Pierce

*Applied & Computational Mathematics, California Institute of Technology,
Pasadena, CA 91125, USA.
niles@caltech.edu*

These lecture notes begin by observing that in many cases the most important engineering outputs of CFD calculations are one or two integral quantities, such as the lift and drag. It is then explained that the solution to an appropriate adjoint problem gives the effect of numerical approximations on the output functional of interest, facilitating the calculation of more accurate functional estimates. The theory is presented for both linear and nonlinear differential equations, incorporating a range of numerical examples illustrating the ability to obtain answers with twice the order of accuracy of the underlying numerical solution.

1 Introduction

1.1 Output functionals

Why do engineers perform CFD calculations? In the case of a transport aircraft at cruise conditions, a calculation might be performed to investigate whether there is an adverse pressure gradient near the leading edge of the wing, causing boundary layer separation and premature transition. Alternatively, one might be concerned about wing/pylon/nacelle integration, in which case one might be looking to see if there are any shocks on the pylon, leading to unacceptable integration losses. In both of these examples, qualitative information is being obtained from the computed flow field to understand and interpret the impact of the phenomena on the quantitative outputs of most concern to the aeronautical engineer, the lift and drag on the aircraft. The quality of the CFD calculation is judged, first and foremost, by the accuracy of the lift and drag predictions. The details of the flow field are much less important, and are used in a more qualitative manner to suggest ways in which the design may be modified to improve the lift or drag. This focus on a few output quantities is even clearer in design optimisation, when one is trying to optimise a single objective function, possibly subject to a number of constraints.

This interest in integral outputs, also referred to as *functionals*, arises in many applications of CFD. Occasionally, volume integrals are of importance. For example, the infra-red signature of a military aircraft will depend in part on a volume integral of some function of the temperature in the thermal wake behind the aircraft. However, usually it is surface integrals that are of most concern, as with lift and drag. Other aeronautical examples include: the roll moment produced by aileron deflection; the mass flow through a compressor blade row; the outflow flux of nitrous oxides from a combustor; the total heat flux into a high pressure turbine blade from the surrounding flow; average noise levels on the ground due to an aircraft landing or taking off.

The idea of output functionals is central to these lecture notes; they are concerned specifically with the analysis of the numerical error in these functionals, and a particular method of correction that very greatly reduces the error, typically doubling the order of accuracy for the functional relative to the underlying flow solution. This distinguishes this kind of error analysis from other approaches that focus on the maximum, root-mean-square or some other measure of the error in the whole flow field. The problem with such measures is that they can have little relation to the errors in the integral outputs of primary concern to the engineer.

As an example, consider the wake behind a wing. To adequately resolve the wake requires a fine grid locally, but it is often the case that the computed wake a chord or two downstream of the wing passes into a region in which the grid resolution is rather coarse. Grid adaptation based on error estimates that look at the whole solution, possibly by looking at the local truncation error, would cause the grid to be further refined in this region. However, the influence of errors in this region on the computed lift and drag would be very small, and a much greater reduction in the lift and drag errors could probably be achieved by adding the grid refinement closer to the wing, possibly near the

leading and trailing edge where very small errors can have an enormous impact on the lift and drag.

1.2 *A priori* and *a posteriori* error analysis

The adjoint error correction technique to be described later is applied as a post-processing step, and so it fits into the framework of *a posteriori* error analysis. This is error analysis based on the computed flow solution, as opposed to *a priori* error analysis that is based on some (usually limited) knowledge of the analytic solution without the benefit of any numerical solution.

A priori error analysis leads to an error bound of the form

$$\text{Error} < c h^p$$

where h is the representative grid spacing, and c, p are positive constants that do not depend on h .

The main point of *a priori* error analysis is finding the value of p , which determines how rapidly the error reduces as the computational grid is refined, uniformly. For most finite difference and finite volume methods, the error in output functionals is of the same order as the error in the flow solution, so it does not matter which error is considered. On structured grids with smooth flow solutions, the solution error is proportional to the truncation error and its order can be relatively easily determined.

The value of the other constant c depends on the details of the analytic problem being solved, the geometry of the computational domain, the boundary conditions, etc. It is extremely difficult to get a good value for c for anything but the simplest problems. Ridiculously large values such as 10^{10} are not uncommon in the literature. This makes the error bound useless in any practical sense, and as a consequence there is often no attempt made to evaluate c .

One area in which *a priori* error analysis is very helpful is applications with singularities in the solution. For such problems, careful analysis can reveal the degree of local grid refinement that is required to recover the order of accuracy (expressed as a function of the total number of grid points) that would be obtained for non-singular solutions.

With *a posteriori* error analysis, ideally one would like a *guaranteed* error bound of the form

$$\text{Error} < e(u_h)$$

where $e(u_h)$ is a computable function of the numerical solution u_h . If the error being considered is the error in the lift from a CFD calculation, this would enable an engineer to perform a calculation and know, with complete certainty, that the true value for the lift lay within certain limits.

For such a bound to be of use, it needs to be *tight*. The efficiency or tightness of the bound is measured by the ratio $e(u_h)/\text{Error}$. A value of 1 is perfect. In the range 2–10, it is useful, but if it were more than 1000 then it would be fairly useless for practical purposes.

Although *guaranteed* error bounds are the ideal, in practice they are extremely difficult to obtain for anything but the simplest of problems. Nonlinearity causes particular difficulties. Therefore, most *a posteriori* bounds are *asymptotic*, so that

$$\text{Error} < \epsilon(u_h) \quad \text{for all } h < h_0$$

The problem is that the value of h_0 is not known. All that is known is that a positive h_0 does exist, below which the asymptotic error bound will be valid. However, above this value the error may exceed the error bound.

As an example, if

$$\text{Error} = 1.36 h^2 + 0.77 h^4,$$

then $1.37 h^2$ is an asymptotic bound that is valid for $h^2 < 0.01/0.77$, but exceeded when $h = 1$.

The distinction between *guaranteed* and *asymptotic* bounds is important. With asymptotic bounds, a user must exercise their judgement to decide whether the grid is sufficiently fine that the bound is likely to be valid. With complex geometries and complex flow fields, this is not easy, particularly for a novice user. On the other hand, with a guaranteed bound one could start with an extremely coarse grid, and then use the guaranteed error bounds to drive grid adaptation until it produces a numerical solution within a user-specified tolerance. This would require no user judgement other than the choice of the error tolerance.

We conclude this discussion of error bounds with a comment on the issue of error bounds versus error correction. Error bounds based on adjoint solutions require a similar level of computational effort to the adjoint error correction to be discussed in these notes. If one has a precise estimate of the error, this could be used to form a near-perfect asymptotic error bound, or it could be used to correct the leading order terms in the error and thereby obtain a solution with a higher order of accuracy. The latter approach is the one that we follow.

1.3 An introduction to adjoints

The use of adjoints lies at the heart of error analysis for output functionals. The main theory will use adjoint differential equations, but here we introduce the ideas at an algebraic level.

Suppose we want to calculate the value of a vector scalar product

$$g^T u,$$

where the vector u is the solution of the system of linear equations

$$Au = f.$$

An equivalent *dual* treatment is to evaluate the product

$$v^T f,$$

where the v is the solution of the *adjoint* (or *dual*) equations

$$A^T v = g,$$

The equivalence of the two calculations comes from the simple identity

$$v^T (A u) = (A^T v)^T u, \tag{1.1}$$

from which it follows that

$$v^T f = g^T u.$$

So, to obtain a linear output functional from the solution of a linear system of equations, we can *either* solve the original equations (sometimes referred to as the *primal* equations) *or* solve the adjoint (dual) equations.

This simple result is the basis for all that follows later. With differential equations, the vector product becomes an integral inner product, the transposed matrix A^T becomes the adjoint differential operator, and the adjoint identity includes certain boundary integral terms, but in essence the equivalence is the same.

When the output is desired for a single f and g , there is no benefit in using the adjoint approach. Either method requires the solution of a linear system of equations of the same dimension, with the same computational cost. The benefit arises when the value of the output is wanted for a single g but several different vectors f . The direct approach would require the solution of the primal equations for each value of f , greatly increasing the computational cost, whereas the dual approach would still require just one adjoint calculation, to be followed by an inexpensive vector product $v^T f$, for each f .

We are now going to look at how this result can be used in two different contexts: design optimisation and error analysis. The motivation for beginning with design optimisation is that this is the primary reason why many research groups within academia and industry are developing adjoint Euler and Navier-Stokes codes. Design optimisation has a clear industrial “pay-off”, whereas the benefits of good error analysis are yet to be appreciated. For the same reason, design optimisation is also the most widely known application for the use of adjoints.

1.3.1 Design optimisation

Consider design optimisation using the ‘discrete’ algebraic approach [18, 17, 45, 1, 43, 20], as opposed to the ‘continuous’ differential approach [34, 39, 59, 5, 35, 37]; see [22, 31] for a discussion of their relative merits and [46] for an excellent review of research on adjoint design methods.

The starting point of the discrete approach is that U , the flow variables at a discrete set of points with coordinates X , is the solution of a system of nonlinear equations

$$N(U, X) = 0,$$

that come from the discretisation of the Euler or Navier-Stokes equations, together with appropriate boundary conditions.

Through the grid generation process, the grid coordinates depend on α which represents one or more geometric design variables. In wing design, for example, perturbations to α might change the thickness distribution and the camber of the wing. If there is only one design variable α , we can linearise about a flow solution for the baseline geometry to obtain

$$Au = f$$

where u is the sensitivity of the flow field to changes in α ,

$$u \equiv \frac{dU}{d\alpha},$$

and

$$A = \frac{\partial N}{\partial U}, \quad f = -\frac{\partial N}{\partial X} \frac{dX}{d\alpha}.$$

The aim of design optimisation is to minimise some objective function $J(U, X)$ which, for example, might be a discrete approximation to the drag. Linearising this function gives

$$\frac{dJ}{d\alpha} = g^T u + \frac{\partial J}{\partial X} \frac{dX}{d\alpha},$$

where

$$g^T = \frac{\partial J}{\partial U}.$$

In the adjoint approach, this sensitivity of the objective function to changes in α is obtained from

$$\frac{dJ}{d\alpha} = v^T f + \frac{\partial J}{\partial X} \frac{dX}{d\alpha},$$

where v satisfies the adjoint equations

$$A^T v = g.$$

If there are several design variables, each has a *different* f , but the *same* g , so the adjoint approach is much cheaper, requiring the solution of just one adjoint set of equations.

1.3.2 Error analysis

We now return to the original problem of evaluating $g^T u$ with u being the solution of the linear equations

$$Au = f.$$

The corresponding dual problem which is to evaluate $v^T f$ where v is the solution of the adjoint equations

$$A^T v = g.$$

Suppose we have approximate solutions \tilde{u}, \tilde{v} to each of these equations. We can then obtain the following result.

$$\begin{aligned}
g^T u &= g^T \tilde{u} + g^T (u - \tilde{u}) \\
&= g^T \tilde{u} + v^T A (u - \tilde{u}) \\
&= g^T \tilde{u} + \tilde{v}^T A (u - \tilde{u}) + (v - \tilde{v})^T A (u - \tilde{u}) \\
&= g^T \tilde{u} - \tilde{v}^T (A\tilde{u} - f) + (v - \tilde{v})^T A (u - \tilde{u}).
\end{aligned} \tag{1.2}$$

The first of the three terms in the final result is the value of the functional using the approximate solution \tilde{u} . The second term is also computable since it involves the known approximate solutions \tilde{u} and \tilde{v} .

The third term is not computable if the exact solutions u and v are not known. However, if \tilde{u} and \tilde{v} are close approximations to u and v , respectively, then the third term will be very small. Thus, the sum of the first two terms gives a very good approximation to the true value of $g^T u$ – a much better approximation in general than $g^T \tilde{u}$.

Note the form of the second term, which we refer to as the *adjoint error correction term*. $A\tilde{u} - f$ is the residual error in solving the equations $Au = f$. The approximate adjoint solution \tilde{v} provides the appropriate weighting for the residual error, giving the effect of the residual error on the output functional of interest. This inner product of a residual error and an adjoint weighting will be repeated throughout these notes.

To take it a step further, suppose now that we want to evaluate a nonlinear function $J(U)$, where U is the solution of the nonlinear equations

$$N(U) = 0.$$

Given an approximate solution \tilde{U} , we define u to be the solution error,

$$u = \tilde{U} - U,$$

and then linearise both the nonlinear equations and the functional to obtain

$$Au \approx f,$$

where

$$A = \frac{\partial N}{\partial U}, \quad f = -N(\tilde{U}).$$

and

$$J(U) \approx J(\tilde{U}) + g^T u,$$

where

$$g^T = \frac{\partial J}{\partial U}.$$

If v is defined to satisfy the adjoint equation

$$A^T v = g,$$

then we obtain

$$J(U) \approx J(\tilde{U}) + v^T f \approx J(\tilde{U}) - v^T N(\tilde{U}).$$

Hence, the quantity

$$J(\tilde{U}) - v^T N(\tilde{U})$$

is a more accurate estimate for $J(U)$ than $J(\tilde{U})$ alone. Again note that the adjoint error correction term is a product of an approximate adjoint solution and the residual error from the original nonlinear equations.

1.3.3 Automatic differentiation

An introduction to adjoints would not be complete without a mention of Automatic Differentiation (AD). This is a technique, implemented in a number of software packages, that starts with a code to compute a nonlinear vector function $F(U)$, and automatically generates codes to compute either

$$\frac{\partial F}{\partial U} \tilde{u}$$

for any \tilde{u} (forward mode), or

$$\left(\frac{\partial F}{\partial U} \right)^T \bar{v}$$

for any \bar{v} (reverse mode).

The forward mode is relatively easy to understand. A computer code can be decomposed into a sequence of binary operations

$$c = \text{op}(a, b),$$

where the operation is addition, subtraction, multiplication or division, plus a few unitary operations

$$c = \text{fn}(a),$$

where the function may be, for example, an exponential or a logarithm. If we treat unitary operations as a special case of binary operations, then linearising a binary operation gives

$$\tilde{c} = \frac{\partial c}{\partial a} \tilde{a} + \frac{\partial c}{\partial b} \tilde{b} = \begin{pmatrix} \frac{\partial c}{\partial a} & \frac{\partial c}{\partial b} \end{pmatrix} \begin{pmatrix} \tilde{a} \\ \tilde{b} \end{pmatrix}.$$

The forward mode AD software inserts the instructions to compute $\frac{\partial c}{\partial a}$ and $\frac{\partial c}{\partial b}$ and evaluate the output sensitivity \tilde{c} given the sensitivities of the two inputs. Carrying this out throughout the code gives the linear sensitivity of the output of the whole code to a specified combination of linear perturbations to the inputs.

The reverse mode AD software performs a task that seems much harder than the forward mode, but in fact it is only slightly harder. Looking again at the single binary instruction, suppose for simplicity that the variables a, b, c are used only once during

the whole code. Let $\bar{a}, \bar{b}, \bar{c}$ denote the sensitivity of the output of the whole code to perturbations in a, b, c . These are then related by the equation

$$\begin{pmatrix} \bar{a} \\ \bar{b} \end{pmatrix} = \begin{pmatrix} \frac{\partial c}{\partial a} & \frac{\partial c}{\partial b} \end{pmatrix}^T \bar{c},$$

which is the transpose of the linear sensitivity equation. The tricky thing with the reverse mode is that the adjoint steps have to be performed in the reverse order to the original nonlinear code. Therefore the AD software has to generate temporary storage for each operation in which to keep the linearisation coefficients such as $\frac{\partial c}{\partial a}$ and $\frac{\partial c}{\partial b}$. Other than this, the computational cost of reverse mode AD is similar to forward mode AD, typically no more than a factor 4 greater than the original nonlinear code.

The main AD packages are ADIFOR [3], Odyssée [19] and TAMC [25]. For further information, the reader is referred to the documentation for each of these, and the excellent book by Griewank [32], one of the original developers of ADIFOR.

The significance of these packages is that they can greatly simplify the task of writing an adjoint CFD code. For examples of the use of AD to generate such codes see [43, 42, 16]. However, there are limitations to their ability to carry out automatic differentiation of codes that use iterative solvers [24], so it is best to view them as aides rather than a black-box solution.

1.4 A brief overview of the literature

Here we give a very brief overview of some of the main developments in the literature concerning the use of adjoints for error analysis.

The subject begins in 1967 with the work of Aubin and Nitsche (see [55]), who used a suitably defined adjoint problem to derive *a priori* optimal order proofs of L_2 convergence of finite element approximations of elliptic p.d.e.'s. In 1978, Babuška and Rheinboldt [11, 10] built on this to develop an *a posteriori* error analysis that they applied to finite element approximations of the Poisson and Cauchy-Riemann equations.

In 1984, Babuška and Miller [7, 8] were perhaps the first to focus attention on integral functional outputs. Because their primary interest was in point functionals such as the maximum stress in structural analysis applications, they used “extraction functions” to convert the point quantities into integrals. A key feature of these papers is the *a priori* analysis of the superconvergence of the finite element approximations of the integral functionals. This will be discussed later in these lecture notes, but the essence is that the adjoint error correction term outlined previously is zero because of a particular feature of Galerkin finite element methods known as “Galerkin orthogonality”. As a result, the order of accuracy of the values for integral functionals is roughly double that of the underlying finite element solution.

In extending this work to the convection-diffusion equation, Barrett and Elliott [4] were the first to analyse a problem that is not self-adjoint, (i.e. one for which the adjoint differential operator is not the same as the original differential operator). This step was vital for CFD applications, none of which are self-adjoint.

The late 1990's saw an explosion of interest and research into *a posteriori* analysis of errors in integral functionals and related methods for optimal grid adaptation. Süli [26, 44, 58, 33], Johnson [36, 38] and Rannacher and Becker [12, 13, 14, 54, 6, 15] have used finite element methods that exhibit natural superconvergence and have focussed their attention on using *a posteriori* error bounds to derive good grid adaptation indicators. In outline, their approaches are similar, but with significant differences in the details.

Patera and Peraire [51, 53, 52] also focus on finite element methods, but they use a completely different *a posteriori* approach to derive error bounds for the functional computed on a “truth mesh” that is defined to be sufficiently fine that the discretisation errors may be neglected. Yet another approach for bounding the errors in functional outputs from finite element methods is that of Oden and Prudhomme [47, 48].

These lecture notes cover the adjoint error correction ideas developed by Giles and Pierce [28, 29, 50, 23]. One way in which they may be viewed is that they extend to finite volume methods the superconvergence that is natural for many finite element methods. This is achieved through the explicit evaluation of the adjoint correction term which is non-zero because of the lack of “orthogonality”. However, as will be shown later, it is also possible to apply the technique with finite element solutions to obtain functional values that are even more accurate than the superconvergent values that arise naturally from the finite element computation.

Venditti & Darmofal [60, 61] have used an algebraic version of the adjoint error correction to correct the functional errors computed on a “truth mesh” using a solution interpolated from the original mesh. They have also used the approach to derive grid adaptation criteria. This will be discussed later in these notes, and sample results will be shown.

2 Linear adjoint error correction

In this section we develop the adjoint correction theory for linear differential equations. We begin with a restricted version without boundary terms because it has the greatest similarity to the algebraic error correction presented in the previous section.

2.1 Theory without boundary terms

Let u be the solution of the linear differential equation

$$Lu = f,$$

on some domain Ω , subject to homogeneous boundary conditions for which the problem is well-posed when $f \in L_2(\Omega)$ (meaning that f is a square-integrable function).

The adjoint differential operator L^* and associated homogeneous boundary conditions are defined by the identity

$$(v, Lu) = (L^*v, u), \tag{2.1}$$

that must hold for all u, v satisfying the respective boundary conditions. Here the notation $(.,.)$ denotes an integral inner product over the domain Ω , i.e.

$$(v, Lu) \equiv \int_{\Omega} v^T Lu \, dV,$$

allowing for the possibility that u , and hence v , may be a vector function rather than just a scalar.

The appropriate definition for L^* can be constructed by integration by parts, starting from (v, Lu) , until all of the derivatives are acting on v rather than u . In the process, the adjoint boundary conditions come from the requirement that the boundary terms that arise from the integration by parts must be zero. Examples of this will be given later.

Suppose now that we are concerned with the value of the functional $J = (g, u)$, for a given function $g \in L_2(\Omega)$. An equivalent dual formulation of the problem is to evaluate the functional $J = (v, f)$, where v satisfies the adjoint equation

$$L^*v = g,$$

subject to the homogeneous adjoint boundary conditions. The equivalence of the two forms of the problem follows immediately from the definition of the adjoint operator.

$$(v, f) = (v, Lu) = (L^*v, u) = (g, u).$$

Suppose that u_h and v_h are approximations to u and v , respectively, and satisfy the homogeneous boundary conditions. The subscript h indicates that the approximate solutions are derived from a numerical computation using a grid with average spacing h . When using finite difference or finite volume methods, u_h and v_h might be created

by interpolation through computed values at grid nodes. With finite element solutions, one might simply use the finite element solutions themselves, or one could again use interpolation through nodal values and thereby obtain approximate solutions that are smoother than the finite element solutions.

It is assumed that u_h and v_h are sufficiently smooth that Lu_h and L^*v_h lie in $L_2(\Omega)$. If u_h and v_h were equal to u and v , then the *residual errors* $Lu_h - f$ and $L^*v_h - g$ would be zero. Thus, the magnitude of the residual errors is a computable indication of the extent to which u_h and v_h are not the true solutions.

Now, using the definitions and identities given above, we obtain the following expression for the functional:

$$\begin{aligned} (g, u) &= (g, u_h) - (L^*v_h, u_h - u) + (L^*v_h - g, u_h - u) \\ &= (g, u_h) - (v_h, L(u_h - u)) + (L^*(v_h - v), u_h - u) \\ &= (g, u_h) - (v_h, Lu_h - f) + (v_h - v, L(u_h - u)). \end{aligned} \quad (2.2)$$

The first term in the final expression is the value of the functional obtained from the approximate solution u_h . The second term is an inner product of the residual error $Lu_h - f$ and the approximate adjoint solution v_h . The adjoint solution gives the weighting of the contribution of the local residual error to the overall error in the computed functional. Therefore, by evaluating and subtracting this adjoint error term we obtain a more accurate value for the functional.

The third term is the remaining error after making the adjoint correction. If $Lu_h - f = L(u_h - u)$ is of the same order of magnitude as $u_h - u$ then the remaining error has a bound that is proportional to the product $\|u_h - u\| \|v_h - v\|$ (using L_2 norms), and thus the corrected functional value is superconvergent. For example, if the solution errors $u_h - u$ and $v_h - v$ are both $O(h^p)$ then the error in the functional is $O(h^{2p})$.

Furthermore, the remaining error term can be expressed as

$$\begin{aligned} (v_h - v, L(u_h - u)) &= (v_h - v, L L^{-1}(Lu_h - f)) \\ &= (L^*(v_h - v), L^{-1}(Lu_h - f)) \\ &= (L^*v_h - g, L^{-1}(Lu_h - f)). \end{aligned}$$

This has the computable *a posteriori* bound $\|L^{-1}\| \|Lu_h - f\| \|L^*v_h - g\|$. The problem with this bound is obtaining a value for the operator norm $\|L^{-1}\|$. This can be calculated analytically in the simplest cases, but for harder problems it may be necessary to estimate it numerically.

Note the similarity between this analysis and the algebraic version in the first section. The adjoint identities (1.1) and (2.1) look almost identical, as do the expressions for the functional, (1.2) and (2.2).

2.2 Galerkin finite element methods

If the approximate solutions u_h and v_h are the finite element solutions from a Galerkin finite element discretisation, then the correction term

$$(v_h, Lu_h - f)$$

is automatically zero, due to the requirement that the finite element residual is orthogonal to all members of the finite element space [55]. Thus, the Galerkin finite element method gives naturally superconvergent estimates for integral outputs, in the sense that a single order of accuracy improvement in the solution, through increasing the degree of the polynomials in the finite element space, leads to two orders of accuracy improvement in the value of the functional.

However, there is usually a loss of accuracy because of a lack of smoothness in the finite element solution. Typically, if the solution errors are $O(h^p)$, then the residual error $Lu_h - f$ is $O(h^{p-m})$ where m is the degree of the differential operator, the degree of the highest derivative in the operator. Hence, the remaining error in the functional is $O(h^{2p-m})$.

If one takes the finite element solution and reconstructs smoother solutions u_h and v_h , then there is the possibility of recovering $O(h^{2p})$ accuracy for the functional, at the cost of carrying out an adjoint calculation to evaluate the adjoint error correction. This will be demonstrated in the second of the two examples to follow.

2.3 First example: 1D Poisson equation

The first example is the one-dimensional Poisson equation,

$$\frac{d^2u}{dx^2} = f,$$

on the unit interval $[0, 1]$ subject to the homogeneous boundary conditions $u(0) = u(1) = 0$.

The dual problem is the Poisson equation,

$$\frac{d^2v}{dx^2} = g,$$

subject to the same homogeneous boundary conditions. The adjoint identity is easily verified, taking into account that u and v are zero at each end.

$$\int_0^1 v \frac{d^2u}{dx^2} dx = - \int_0^1 \frac{dv}{dx} \frac{du}{dx} dx = \int_0^1 \frac{d^2v}{dx^2} u dx.$$

The Poisson equation is approximated numerically on a uniform grid, with spacing h , using a second order finite difference discretisation,

$$h^{-2}\delta_x^2 u_j = f(x_j).$$

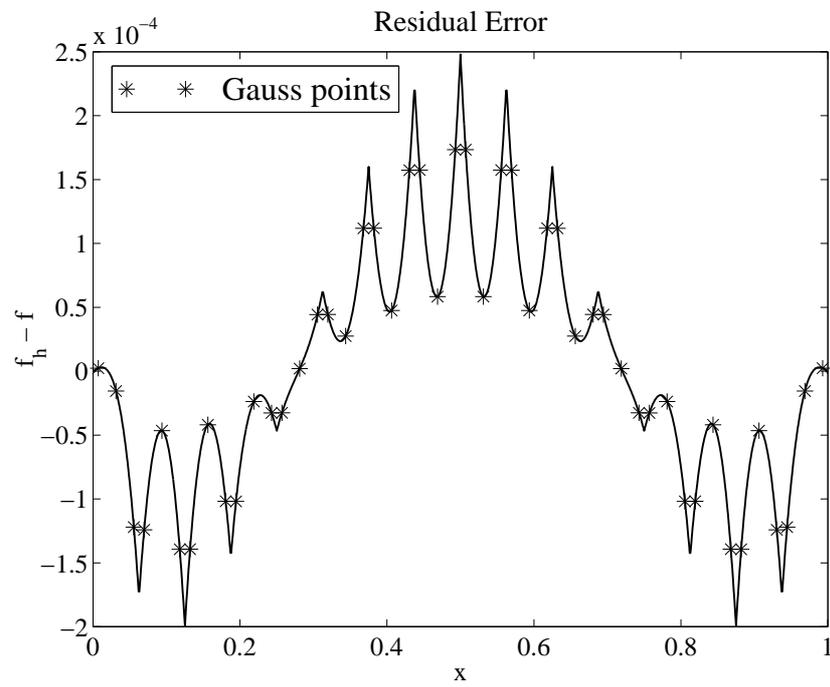


Figure 1: Residual error for 1D Poisson equation.

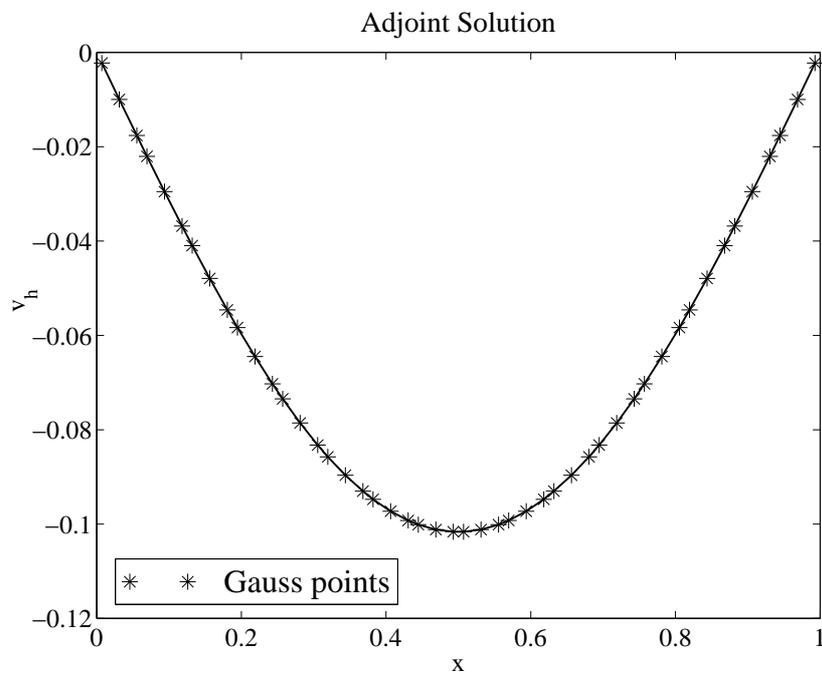


Figure 2: Adjoint solution for 1D Poisson equation.

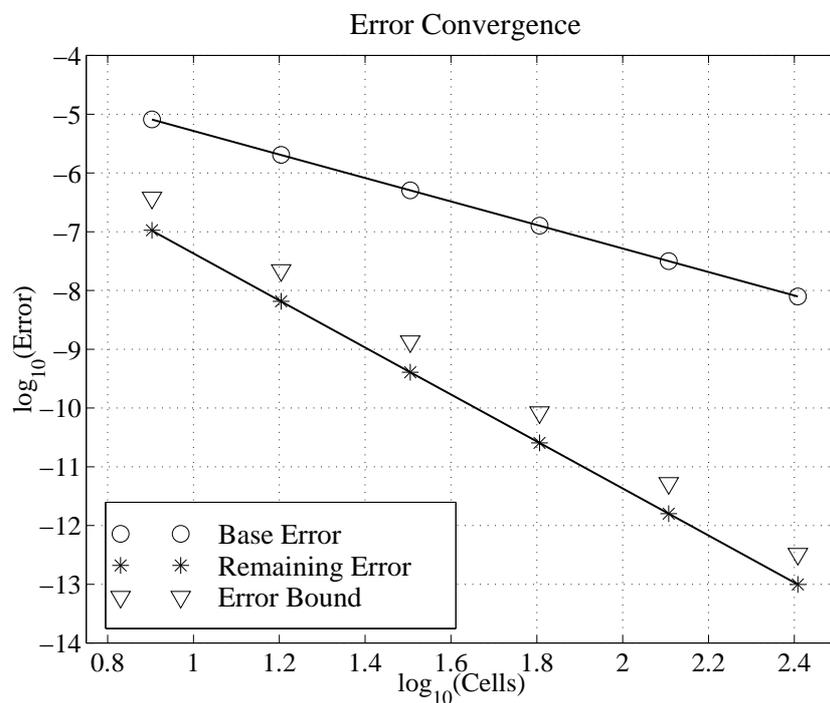


Figure 3: Functional error convergence for 1D Poisson equation.

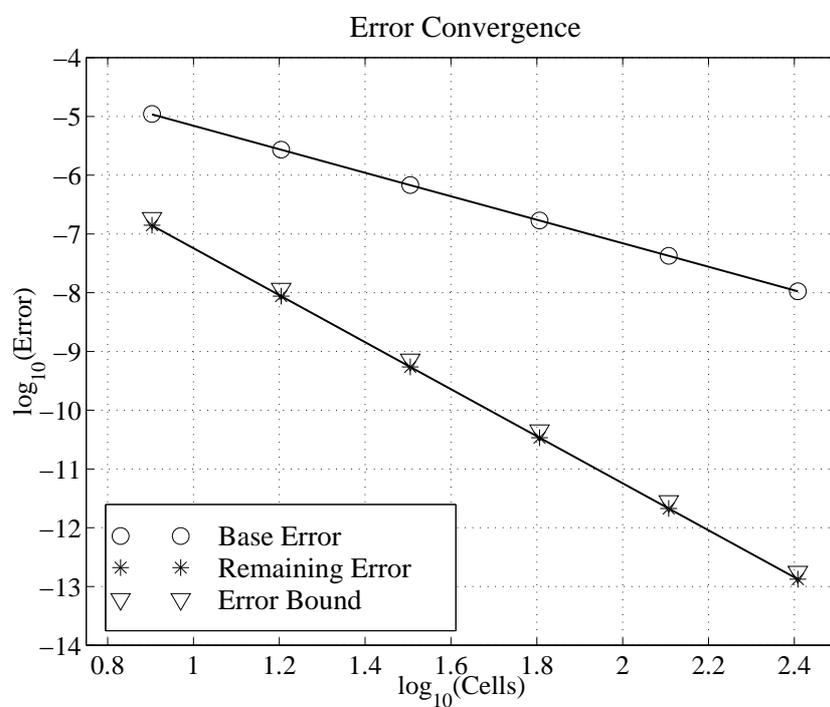


Figure 4: Functional error convergence for 2D Poisson equation.

The approximate solution $u_h(x)$ is then defined by cubic spline interpolation through the nodal values u_j . The adjoint solution v_h is obtained in exactly the same manner.

Numerical results have been obtained for the case

$$f = x^3(1-x)^3, \quad g = \sin(\pi x).$$

Figure 1 shows the residual error $Lu_h - f$ when $h = \frac{1}{32}$, as well as the three Gaussian quadrature points on each sub-interval that are used in the numerical integration of the inner product $(v_h, Lu_h - f)$. Since u_h is a cubic spline, $f_h \equiv \frac{d^2 u_h}{dx^2}$ is continuous and piecewise linear. The best piecewise linear approximation to f has an approximation error whose dominant term is quadratic on each sub-interval; this explains the scalloped shape of the residual error. Figure 2 shows the approximate adjoint solution v_h , which simply illustrates that the residual error in the center of the domain contributes most to the overall error in the functional.

Figure 3 is a log-log plot of three quantities versus the number of cells: the error in the base value of the functional (g, u_h) ; the remaining error after subtracting the adjoint correction term $(v_h, Lu_h - f)$; the *a posteriori* error bound $\|L^{-1}\| \|Lu_h - f\| \|L^* v_h - g\|$. The superimposed lines have slopes of -2 and -4 , confirming that the base solution is second order accurate while the error in the corrected functional and the error bound are both fourth order. It is also worth noting that on a grid with 16 cells, which might be a reasonable choice for practical computations, the error in the corrected value of the functional is over 200 times smaller than the uncorrected error.

2.4 Second example: 2D Poisson equation

The second example is the two-dimensional Poisson equation,

$$\nabla^2 u = f,$$

on the unit square $[0, 1] \times [0, 1]$ subject to homogeneous Dirichlet boundary conditions. The dual problem is

$$\nabla^2 v = g,$$

with the same boundary conditions, and the adjoint identity is again easily verified,

$$\int_{\Omega} v \nabla^2 u \, dA = - \int_{\Omega} \nabla v \cdot \nabla u \, dA = \int_{\Omega} \nabla^2 v \, u \, dA.$$

For this example, the equations are approximated using a Galerkin finite element method with piecewise bilinear elements on a uniform Cartesian grid. Finite element error analysis reveals that the solution error for the primal problem, and the error in the computed functional using the finite element solution are both $O(h^2)$. However, using bi-cubic spline interpolation through the computed nodal values, one can reconstruct an improved approximate solution $u_h(x, y)$ with an error that is still $O(h^2)$, but much smoother, so that the residual error is also $O(h^2)$. Using a similarly reconstructed approximate adjoint solution $v_h(x, y)$, one can then compute the adjoint error correction

term resulting in a corrected functional whose accuracy is $O(h^4)$. All inner product integrals are approximated by 3×3 Gaussian quadrature on each square cell to ensure that the numerical quadrature errors are of a higher order.

Figure 4 shows the numerical results obtained for the functions

$$f(x, y) = x(1-x)y(1-y), \quad g(x, y) = \sin(\pi x) \sin(\pi y).$$

The ordinate is the log of the number of cells in each dimension, and lines of slope -2 and -4 are again superimposed. As predicted by the analysis, the base error in the functional is clearly second order whereas the error in the corrected value of the functional as well as the error bound are again fourth order.

2.5 Theory with boundary terms

We now extend the theory to include inhomogeneous boundary conditions for the primal and dual problems, and boundary integrals in their output functionals.

Let u be the solution of the linear differential equation

$$Lu = f,$$

in the domain Ω , subject to the linear boundary conditions

$$Bu = e,$$

on the boundary $\partial\Omega$. In general, the dimension of the operator B may be different on different parts of the boundary (e.g. inflow and outflow sections for the convection p.d.e.).

The output functional of interest is taken to be

$$J = (g, u) + (h, Cu)_{\partial\Omega},$$

where $(\cdot, \cdot)_{\partial\Omega}$ represents an integral inner product over the boundary $\partial\Omega$. The boundary operator C may be algebraic (e.g. $Cu \equiv u$) or differential (e.g. $Cu \equiv \frac{\partial u}{\partial n}$), but must have the same dimension as the adjoint boundary condition operator B^* to be defined shortly. Note that the components of h may be set to zero if the functional does not have a boundary integral contribution.

The corresponding linear adjoint problem is

$$L^*v = g,$$

in Ω , subject to the boundary conditions

$$B^*v = h,$$

on the boundary $\partial\Omega$. The fundamental identity defining L^* , B^* and the boundary operator C^* is

$$(v, Lu) + (C^*v, Bu)_{\partial\Omega} = (L^*v, u) + (B^*v, Cu)_{\partial\Omega}, \quad (2.3)$$

for all u, v . This identity is obtained by integration by parts. Examples will be given later, but see also [27] for the construction of the appropriate adjoint operators for the linearised Euler and Navier-Stokes equations.

Using the adjoint identity, one immediately obtains the equivalent dual form of the output functional,

$$J = (v, f) + (C^*v, e)_{\partial\Omega}.$$

Given approximate solutions u_h, v_h , we obtain the following result for the functional.

$$\begin{aligned} (g, u) + (h, Cu)_{\partial\Omega} &= (g, u_h) + (h, Cu_h)_{\partial\Omega} \\ &\quad - (L^*v_h, u_h - u) - (B^*v_h, C(u_h - u))_{\partial\Omega} \\ &\quad + (L^*v_h - g, u_h - u) + (B^*v_h - h, C(u_h - u))_{\partial\Omega} \\ &= (g, u_h) + (h, Cu_h)_{\partial\Omega} \\ &\quad - (v_h, L(u_h - u)) - (C^*v_h, B(u_h - u))_{\partial\Omega} \\ &\quad + (L^*(v_h - v), u_h - u) + (B^*(v_h - v), C(u_h - u))_{\partial\Omega} \\ &= (g, u_h) + (h, Cu_h)_{\partial\Omega} \\ &\quad - (v_h, Lu_h - f) - (C^*v_h, Bu_h - e)_{\partial\Omega} \\ &\quad + (v_h - v, L(u_h - u)) + (C^*(v_h - v), B(u_h - u))_{\partial\Omega}. \end{aligned}$$

In the final result, the first line is the functional based on the approximate solution u_h . The second line is the computable adjoint error correction that now includes a term related to the residual error in satisfying the primal boundary conditions. The third line is the remaining error. In principle, an *a posteriori* error bound for this can again be found, but the main point is that each of the terms involves an inner product of two small quantities, so we again have the result that the corrected functional is superconvergent relative to the underlying solutions to the primal and adjoint problems.

2.6 Example: 2D Laplace equation

For the 2D Laplace equation

$$\nabla^2 u = 0,$$

with Dirichlet boundary conditions $u = e$, and functional

$$J = \int_{\partial\Omega} h \frac{\partial u}{\partial n} \, ds,$$

we have the operators

$$Lu = \nabla^2 u, \quad Bu = u, \quad Cu = \frac{\partial u}{\partial n}.$$

Integrating by parts gives

$$\begin{aligned} \int_{\Omega} v Lu \, dA &= - \int_{\Omega} \nabla v \cdot \nabla u \, dA + \int_{\partial\Omega} v \frac{\partial u}{\partial n} \, ds \\ &= \int_{\Omega} \nabla^2 v u \, dA + \int_{\partial\Omega} \left(v \frac{\partial u}{\partial n} - \frac{\partial u}{\partial n} v \right) \, ds, \end{aligned}$$

so the adjoint identity is satisfied for all u, v if we define

$$L^*v = \nabla^2 v, \quad B^*v = v, \quad C^*v = \frac{\partial v}{\partial n}.$$

To construct an analytic testcase with curved boundaries and a singularity in the solution, we use a conformal mapping. We start by defining the domain in a complex Z -plane to be the region between two circles centered at $(X, Y) = (-0.1, 0)$ with radii of $R_1 = 1.1$ and $R_2 = 3.0$. Application of the Joukowski mapping

$$z = Z + \frac{1}{Z},$$

then produces a computational domain between a cusped airfoil ($\partial\Omega_{z_1}$) and a smooth outer boundary ($\partial\Omega_{z_2}$). Using cylindrical coordinates R, θ defined by

$$X + 0.1 = R \cos \theta, \quad Y = R \sin \theta,$$

the function

$$U(X, Y) = \frac{R^2 - R_1^2}{R} \sin \theta,$$

is a solution of the Laplace equation in the Z -plane. Furthermore, by a well-known feature of conformal mappings, the function $u(x, y) = U(X, Y)$ is also a solution of the Laplace equation in the z -plane.

Evaluating $u(x, y)$ on the inner and outer boundaries gives the Dirichlet boundary condition for the test problem. As illustrated in Figure 5, the solution corresponds to the stream function for incompressible inviscid flow around the airfoil, with zero circulation.

The functional, expressed in the Z -plane, is chosen to be

$$\int_0^{2\pi} \sin \theta \left. \frac{\partial U}{\partial n} \right|_{R=R_1} d\theta.$$

Its analytic value is -2π . When mapped into the z -plane, the corresponding expression for the functional is

$$\left(h, \frac{\partial u}{\partial n} \right)_{\partial\Omega},$$

where $h = R_1^{-1} \sin \theta$, on the inner boundary $\partial\Omega_{z_1}$, and $h = 0$ on the outer boundary. Hence the dual problem is the Laplace equation subject to the Dirichlet boundary conditions $v = h$. As seen in Figure 5, the gradient of the adjoint solution is singular at the cusped trailing edge of the airfoil.

The numerical results for both the primal and dual problems are calculated using the bilinear Galerkin finite element method. The grid points in the z -plane are generated by the conformal mapping of a regular polar grid in the Z -plane, but the use of isoparametric elements in the z -plane means the cells in the z -plane are quadrilaterals, and do not have curved edges.

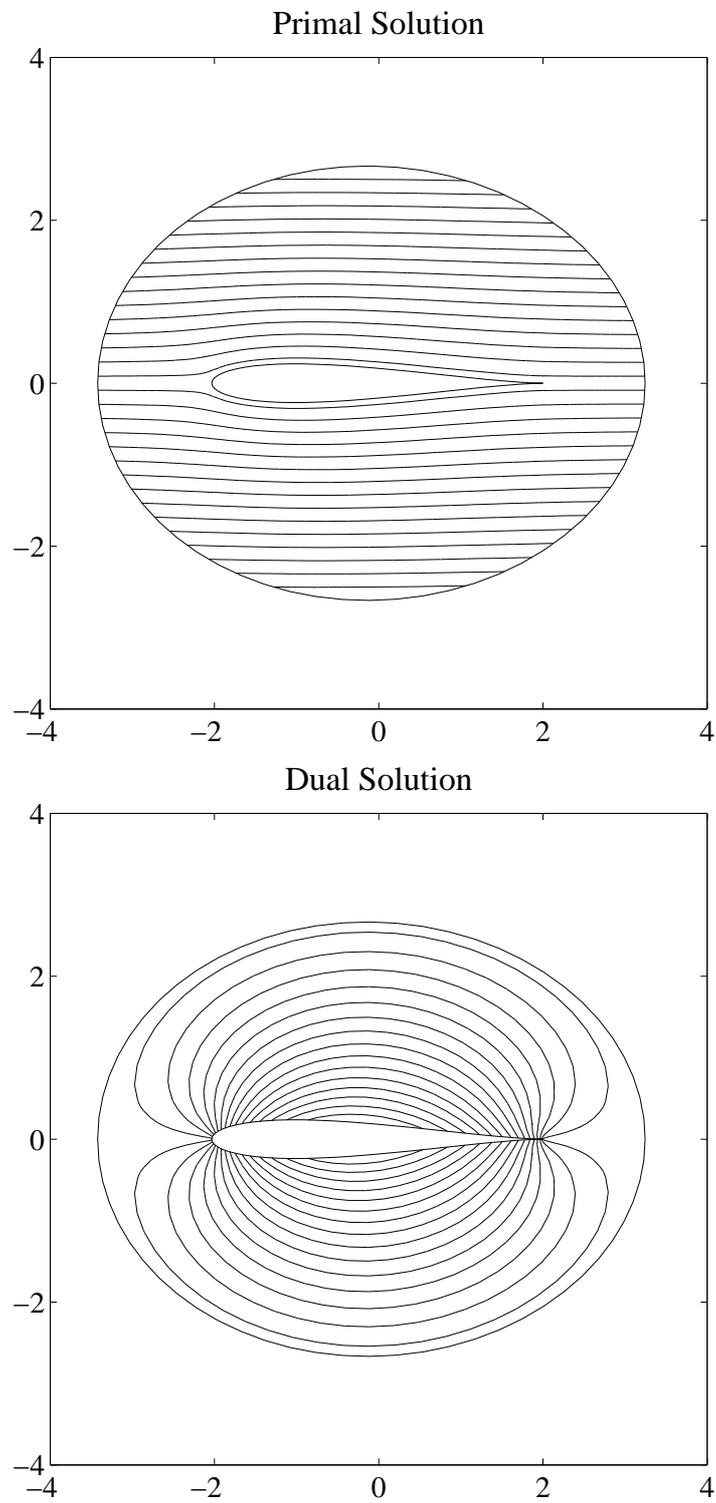


Figure 5: The reconstructed primal and dual solutions for a 2D Laplace problem around a Joukowski airfoil.

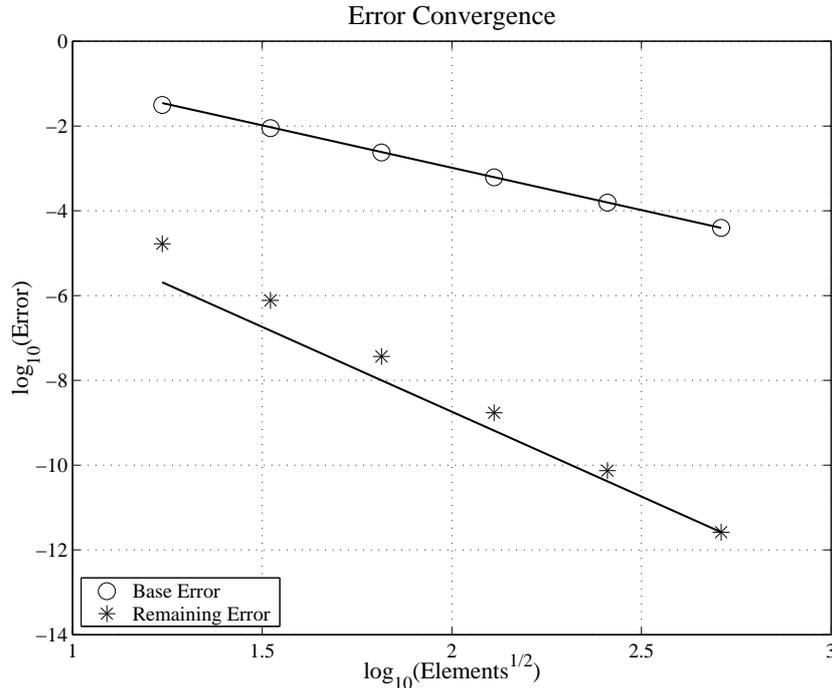


Figure 6: Error convergence of a boundary functional for a 2D Laplace problem around a Joukowski airfoil.

The approximate solutions u_h and v_h are formed by bi-cubic spline interpolation. The coordinate data is also splined, so that the solutions and coordinates, u_h, v_h, x, y , are all defined parametrically as functions of the two spline coordinates ξ, η .

The error correction integral, which in this case is simply

$$(v_h, \nabla^2 u_h)$$

is evaluated by transforming it into an integral over (ξ, η) , and then using 3×3 Gaussian quadrature on each quadrilateral cell.

Figure 6 shows the numerical values obtained for the functional, with and without the adjoint error correction, plotted versus the square root of the number of cells, which is a measure of $1/h$, an average inverse grid spacing. The superimposed lines of slope -2 and -4 show that the base solution is again second order accurate whereas the corrected value for the functional is fourth order accurate. This improvement is achieved despite the presence of the singularity at the trailing edge, and the added complications of the curved boundaries and the boundary functional.

3 Linear defect error correction

Adjoint error correction is not the only means of improving the accuracy of numerical calculations. In this section, based on Reference [23], we look at the use of defect correction [9, 40, 56, 57], and show that it can be extremely effective in reducing the errors in a model 1D Helmholtz problem; the combination of defect and adjoint error correction is even better.

The primary motivation for this investigation is the need for high order accuracy for aeroacoustic and electromagnetics calculations. In steady CFD calculations, grid adaptation can be used to provide high grid resolution in the limited areas that require it. However, using standard second order accurate methods, the wave-like nature of aeroacoustic and electromagnetic solutions would lead to grid refinement throughout the computational domain in order to reduce the wave dispersion and dissipation to acceptable levels. The preferable alternative is to use higher order methods, allowing one to use fewer points per wavelength, which can lead to a very substantial reduction in the total number of grid points for 3D calculations. The difficulty with this is that one often wants to use unstructured grids because of their geometric flexibility, and the construction of higher order approximations on unstructured grids is complicated and computationally expensive.

3.1 Problem description and Galerkin method

The model problem to be solved is the 1D Helmholtz equation

$$u'' + \pi^2 u = 0, \quad 0 < x < 10,$$

subject to the Dirichlet boundary condition $u = 1$ at $x = 0$ and the radiation boundary condition $u' - i\pi u = 0$ at $x = 10$. The analytic solution is $u = \exp(i\pi x)$ and the domain contains precisely five wavelengths. The output functional of interest is the value $u(10)$ at the right hand boundary. This can be viewed as a model of a far-field boundary integral giving the radiated acoustic energy in aeroacoustics, or the radar cross-section in electromagnetics [44].

Integrating by parts, the weak form of the inhomogeneous equation

$$u'' + \pi^2 u = f, \quad 0 < x < 10,$$

subject to the same boundary conditions is

$$-(w', u') + \pi^2(w, u) + i\pi w^*(10) u(10) = (w, f),$$

for any differentiable $w(x)$ with $w(0) = 0$. One important feature of this Helmholtz problem is that the solution is complex. Therefore the inner product (w, u) is defined as

$$(w, u) \equiv \int_0^{10} w^* u \, dx,$$

with w^* denoting the complex conjugate of w .

The Galerkin solution on the irregular grid $x_j, j = 0, 1, 2, \dots, N$, is defined as

$$U(x) = \sum_{j=0}^N U_j \phi_j(x)$$

where the $\phi_j(x)$ are the usual piecewise linear ‘hat’ functions for which $\phi_j(x_i) = \delta_{ij}$. The value U_0 is given by the Dirichlet boundary condition. The values of the other coefficients U_j for $j > 0$ are obtained from the equations

$$-(\phi'_i, U') + \pi^2(\phi_i, U) + i\pi\phi_i(10)U(10) = 0, \quad i = 1, 2, \dots, N.$$

It is well established that this discretisation is second order accurate, producing dispersion but no dissipation on a uniform grid.

3.2 Defect correction

The first step in the defect correction is to define a new approximate solution $u_h(x)$ by cubic spline interpolation of the nodal values U_j . The choice of end conditions for the cubic spline is very important. A natural cubic spline would have $u_h'' = 0$ at both ends, but this would introduce small errors at each end since $u'' \neq 0$ for the analytic solution. Instead, at $x = 10$ we require the splined solution to satisfy the analytic boundary condition by imposing $u_h' - i\pi u_h = 0$. At $x = 0$, the analytic boundary condition is already imposed through having the correct value for the end point $U(0)$. Therefore, here we require that $u_h'' + \pi^2 u_h = 0$ so the splined solution satisfies the o.d.e. at the boundary.

The solution error, $e = u(x) - u_h(x)$ satisfies the inhomogeneous Helmholtz equation

$$e'' + \pi^2 e = -(u_h'' + \pi^2 u_h), \quad 0 < x < 10,$$

the right-hand-side of which is the residual error of the approximation $u_h(x)$. Given the homogeneous Dirichlet boundary condition at $x = 0$, and the same radiation boundary condition at $x = 10$, the Galerkin approximation to the error is given by the equations

$$-(\phi'_i, E') + \pi^2(\phi_i, E) + i\pi\phi_i(10)E(10) = -(\phi_i, u_h'' + \pi^2 u_h), \quad i = 1, 2, \dots, N.$$

Adding the nodal corrections E_j to the original nodal values U_j gives a corrected solution. The whole procedure can then be repeated to improve the accuracy. This follows the procedure described by Barrett *et al* who also showed that it converges to a solution of an appropriately defined Petrov-Galerkin discretisation, with the trial space being the space of cubic splines, while the test space is the space of piecewise linear functions [9].

3.3 Adjoint error correction

To apply the linear theory to the Helmholtz problem, the first step is to construct the appropriate adjoint problem. Integration by parts reveals that the Helmholtz equation is self-adjoint, so

$$L^*v \equiv v'' + \pi^2 v,$$

and

$$(v, Lu) - (L^*v, u) = \left[\vec{v}^H \vec{A} \vec{u} \right]_0^{10},$$

where

$$\vec{u} = \begin{pmatrix} u \\ \frac{du}{dx} \end{pmatrix}, \quad \vec{v} = \begin{pmatrix} v \\ \frac{dv}{dx} \end{pmatrix},$$

and

$$\vec{A} = \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix}.$$

At $x = 10$ we have

$$Bu \equiv u' - i\pi u \equiv \vec{B} \vec{u}, \quad \vec{B} = (-i\pi \ 1),$$

and

$$Cu \equiv u \equiv \vec{C} \vec{u}, \quad \vec{C} = (1 \ 0).$$

To satisfy the adjoint identity (2.3) we require \vec{B}^* and \vec{C}^* such that

$$\vec{A} = \begin{pmatrix} -\vec{C}^* \\ \vec{B}^* \end{pmatrix}^H \begin{pmatrix} \vec{B} \\ \vec{C} \end{pmatrix}.$$

Solving this gives

$$\begin{pmatrix} -\vec{C}^* \\ \vec{B}^* \end{pmatrix} = \begin{pmatrix} \vec{B} \\ \vec{C} \end{pmatrix}^{-H} \vec{A}^H = \begin{pmatrix} 1 & 0 \\ -i\pi & -1 \end{pmatrix}$$

and hence $B^*v \equiv -v' - i\pi v$ and $C^*v \equiv -v$. Similarly, at $x=0$, we obtain $B^*v = v$ and $C^*v = v'$.

Now, noting that in our application $f = g = 0$, and h has value 0 at $x=0$ and 1 at $x=10$, then the full specification of the adjoint problem is

$$v'' + \pi^2 v = 0, \quad 0 < x < 10,$$

with $v=0$ at $x=0$ and $-v' - i\pi v = 1$ at $x=10$.

Let v_h be an approximate solution of this problem, obtained by the same Galerkin and cubic spline reconstruction approach as u_h , with or without defect correction. Noting that the cubic spline reconstruction ensures that the boundary conditions are satisfied exactly, the corrected approximation to the value $u(10)$ is

$$u_h(10) - (v_h, u_h'' + \pi^2 u_h).$$

The theory gives the error in this corrected functional as being

$$(v_h - v, u_h'' + \pi^2 u_h).$$

In the absence of defect correction, both terms in this inner product are second order in the average grid spacing and so the error is fourth order. With defect correction, the first term is fourth order while the second term remains second order. Therefore, the error remaining after the adjoint error correction is sixth order.

3.4 Numerical results

Numerical results have been obtained for grids with 4, 8, 16, 32, 64 and 128 points per wavelength. To test the ability to cope with irregular grids, the coordinates for the grid with N intervals are defined as

$$x_0 = 0, \quad x_N = 10, \quad x_j = \frac{10}{N} (j + \sigma_j), \quad 0 < j < N,$$

where σ_j is a uniformly distributed random variable in the range $[-0.3, 0.3]$.

Figure 1 shows the L_2 norm of the error in the reconstructed cubic spline solution before and after defect correction. Without defect correction, the error is second order, while with defect correction it is fourth order. Note that a second application of defect correction makes a significant reduction in the error even though it remains fourth order. This is because one application of the defect correction procedure gives a correction that is second order in magnitude, with a corresponding error that is second order in relative magnitude and therefore fourth order in absolute magnitude. It is this error that is corrected by a second application of the defect correction procedure.

Figure 2 shows the error in the numerical value for the output functional $u(10)$. Without any correction, the error is second order. Using either defect correction or adjoint error correction on their own increases the order of accuracy to fourth order, but using them both increases the accuracy to sixth order. Note that the calculation with 8 points per wavelength plus both defect and adjoint error correction gives an error which is approximately 2×10^{-3} . This is more accurate than the calculation with 128 points per wavelength and no corrections, and comparable to the results using 14 points and defect correction, or 30 points with adjoint error correction.

In 3D, the computational cost is proportional to the cube of the number of points per wavelength, so this indicates the potentially huge savings offered by the combination of defect and adjoint error correction. The cost of computing the corrections is five times the cost of the original calculation, due to the additional two calculations for the defect correction, and the one adjoint calculation plus its two defect corrections. In practice, the second defect correction for the primal and adjoint calculations make negligible difference to the value obtained after the adjoint error correction, so these can be omitted, reducing the cost of the corrections to just three times the cost of the original calculation.

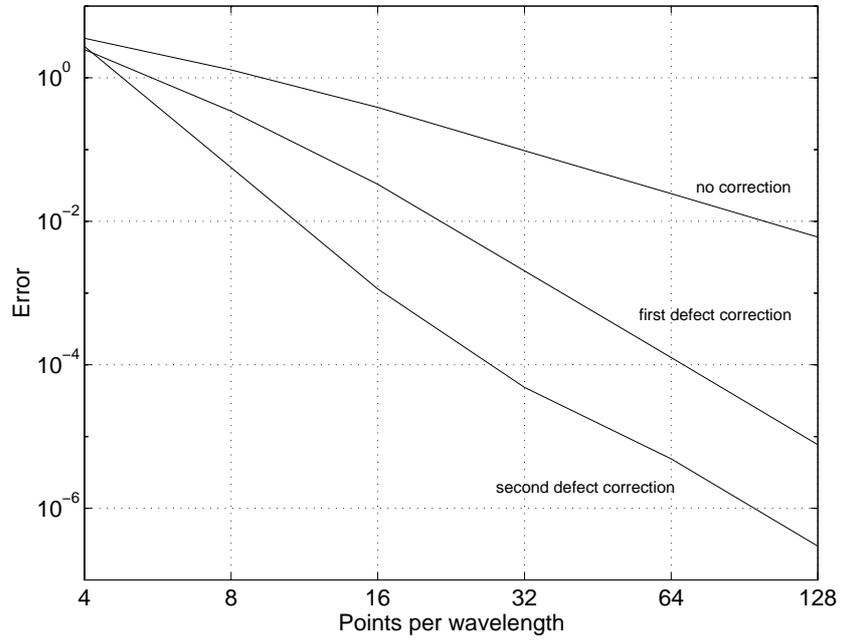


Figure 7: L_2 error in the numerical approximation to $u(x)$

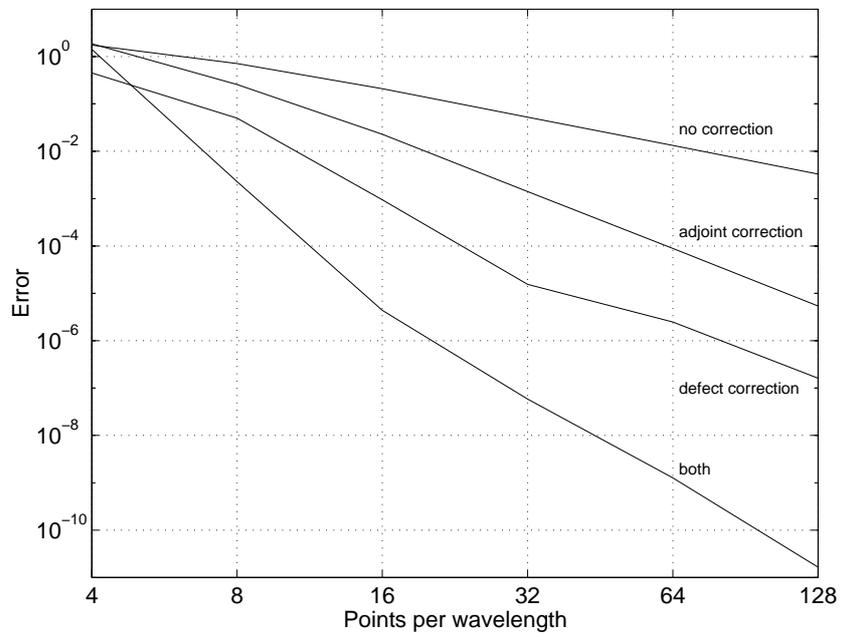


Figure 8: Error in the numerical approximation to $u(10)$

4 Nonlinear adjoint error correction

This section looks at the extension from the linear theory to handle nonlinear problems. It begins with some preliminaries that address the key issues in linearising nonlinear functions and operators.

4.1 Preliminaries

If u is a scalar variable and $f(u)$ is a nonlinear scalar function then a standard Taylor series expansion gives

$$f(u_2) = f(u_1) + f'(u_1) (u_2 - u_1) + O((u_2 - u_1)^2).$$

However, one can obtain an exact expression without any remainder terms by starting from

$$\frac{d}{d\theta} f(u_1 + \theta(u_2 - u_1)) = f'(u_1 + \theta(u_2 - u_1)) (u_2 - u_1),$$

and then integrating this from $\theta=0$ to $\theta=1$ to obtain

$$f(u_2) - f(u_1) = \overline{f'}_{(u_1, u_2)} (u_2 - u_1),$$

where

$$\overline{f'}_{(u_1, u_2)} \equiv \int_0^1 f'(u_1 + \theta(u_2 - u_1)) d\theta.$$

If u and f are vectors, we need to define the Jacobian matrix

$$A_u = \left. \frac{\partial f}{\partial u} \right|_u,$$

with the subscript u denoting the fact that the value of the Jacobian matrix depends on the value of u around which $f(u)$ is linearised. We then obtain

$$\frac{d}{d\theta} f(u_1 + \theta(u_2 - u_1)) = A_{u_1 + \theta(u_2 - u_1)} (u_2 - u_1)$$

so integrating over θ gives

$$f(u_2) - f(u_1) = \overline{A}_{(u_1, u_2)} (u_2 - u_1),$$

where

$$\overline{A}_{(u_1, u_2)} = \int_0^1 \left. \frac{\partial f}{\partial u} \right|_{u_1 + \theta(u_2 - u_1)} d\theta.$$

The next step is to consider a nonlinear operator $N(u)$. The linearised operator L_u is called a Fréchet derivative, and it is formally defined by

$$L_u \tilde{u} \equiv \lim_{\varepsilon \rightarrow 0} \frac{N(u + \varepsilon \tilde{u}) - N(u)}{\varepsilon}$$

Again the subscript u denotes the fact that the linear operator matrix depends on the value of u around which $N(u)$ is linearised. For example, if

$$N(u) = \frac{\partial}{\partial x} \left(\frac{1}{2} u^2 \right) - \nu \frac{\partial^2 u}{\partial x^2}$$

then

$$L_u \tilde{u} = \frac{\partial}{\partial x} (u \tilde{u}) - \nu \frac{\partial^2 \tilde{u}}{\partial x^2}$$

The final step in these preliminaries is to start from

$$\frac{d}{d\theta} N(u_1 + \theta(u_2 - u_1)) = L_{u_1 + \theta(u_2 - u_1)}(u_2 - u_1)$$

and then integrate over θ to obtain

$$N(u_2) - N(u_1) = \bar{L}_{(u_1, u_2)}(u_2 - u_1),$$

where

$$\bar{L}_{(u_1, u_2)} = \int_0^1 L|_{u_1 + \theta(u_2 - u_1)} d\theta.$$

Thus $\bar{L}_{(u_1, u_2)}$ is the average value of the linear operator L_u over the “path” from u_1 to u_2 .

4.2 Nonlinear theory

Let u be the solution of the nonlinear differential equation

$$N(u) = 0,$$

in the domain Ω , subject to the nonlinear boundary conditions

$$D(u) = 0,$$

on the boundary $\partial\Omega$.

The linear differential operators L_u and B_u are defined to be the Fréchet derivatives of N and D , respectively,

$$L_u \tilde{u} \equiv \lim_{\epsilon \rightarrow 0} \frac{N(u + \epsilon \tilde{u}) - N(u)}{\epsilon},$$

$$B_u \tilde{u} \equiv \lim_{\epsilon \rightarrow 0} \frac{D(u + \epsilon \tilde{u}) - D(u)}{\epsilon}.$$

It is assumed that the nonlinear functional of interest, $J(u)$, has a Fréchet derivative of the following form,

$$\lim_{\epsilon \rightarrow 0} \frac{J(u + \epsilon \tilde{u}) - J(u)}{\epsilon} = (g(u), \tilde{u}) + (h, C_u \tilde{u})_{\partial\Omega}.$$

Here the dimension of the operator C_u (which may be differential) is required to equal the dimension of the adjoint boundary operator B_u^* , to be defined shortly.

The corresponding linear adjoint problem is

$$L_u^* v = g(u)$$

in Ω , subject to the boundary conditions

$$B_u^* v = h$$

on the boundary $\partial\Omega$. The adjoint identity defining L_u^* , B_u^* and the boundary operator C_u^* is

$$(v, L_u \tilde{u}) + (C_u^* v, B_u \tilde{u})_{\partial\Omega} = (L_u^* v, \tilde{u}) + (B_u^* v, C_u \tilde{u})_{\partial\Omega}, \quad (4.1)$$

for all \tilde{u}, v .

We now consider approximate solutions u_h, v_h of the primal and dual problems, respectively. The analysis will use the quantities

$$L_{u_h}^* v_h, \quad B_{u_h}^* v_h, \quad C_{u_h}^* v_h.$$

Note that these can be evaluated since u_h and v_h are both known, whereas we would not be able to evaluate the Fréchet derivatives based on the unknown analytic solution u .

The analysis also requires averaged Fréchet derivatives defined by

$$\begin{aligned} \bar{L}_{(u, u_h)} &= \int_0^1 L|_{u+\theta(u_h-u)} d\theta, \\ \bar{B}_{(u, u_h)} &= \int_0^1 B|_{u+\theta(u_h-u)} d\theta, \\ \bar{C}_{(u, u_h)} &= \int_0^1 C|_{u+\theta(u_h-u)} d\theta, \\ \bar{g}(u, u_h) &= \int_0^1 g(u + \theta(u_h - u)) d\theta, \end{aligned}$$

so that, as explained in the preliminaries,

$$\begin{aligned} N(u_h) - N(u) &= \bar{L}_{(u, u_h)}(u_h - u), \\ D(u_h) - D(u) &= \bar{B}_{(u, u_h)}(u_h - u), \\ J(u_h) - J(u) &= (\bar{g}(u, u_h), u_h - u) + (h, \bar{C}_{(u, u_h)}(u_h - u))_{\partial\Omega}. \end{aligned}$$

We now obtain the following:

$$\begin{aligned}
J(u_h) - J(u) &= (\bar{g}(u, u_h), u_h - u) + (h, \bar{C}_{(u, u_h)}(u_h - u))_{\partial\Omega} \\
&= (L_{u_h}^* v_h, u_h - u) + (B_{u_h}^* v_h, C_{u_h}(u_h - u))_{\partial\Omega} \\
&\quad - (L_{u_h}^* v_h - \bar{g}(u, u_h), u_h - u) \\
&\quad - (h, (C_{u_h} - \bar{C}_{(u, u_h)})(u_h - u))_{\partial\Omega} \\
&\quad - (B_{u_h}^* v_h - h, C_{u_h}(u_h - u))_{\partial\Omega} \\
&= (v_h, L_{u_h}(u_h - u)) + (C_{u_h}^* v_h, B_{u_h}(u_h - u))_{\partial\Omega} \\
&\quad - (L_{u_h}^* v_h - \bar{g}(u, u_h), u_h - u) \\
&\quad - (h, (C_{u_h} - \bar{C}_{(u, u_h)})(u_h - u))_{\partial\Omega} \\
&\quad - (B_{u_h}^* v_h - h, C_{u_h}(u_h - u))_{\partial\Omega} \\
&= (v_h, \bar{L}_{(u, u_h)}(u_h - u)) + (C_{u_h}^* v_h, \bar{B}_{(u, u_h)}(u_h - u))_{\partial\Omega} \\
&\quad - (L_{u_h}^* v_h - \bar{g}(u, u_h), u_h - u) \\
&\quad - (h, (C_{u_h} - \bar{C}_{(u, u_h)})(u_h - u))_{\partial\Omega} \\
&\quad - (B_{u_h}^* v_h - h, C_{u_h}(u_h - u))_{\partial\Omega} \\
&\quad + (v_h, (L_{u_h} - \bar{L}_{(u, u_h)})(u_h - u)) \\
&\quad + (C_{u_h}^* v_h, (B_{u_h} - \bar{B}_{(u, u_h)})(u_h - u))_{\partial\Omega} \\
&= (v_h, N(u_h)) + (C_{u_h}^* v_h, D(u_h))_{\partial\Omega} \\
&\quad - (L_{u_h}^* v_h - \bar{g}(u, u_h), u_h - u) \\
&\quad - (h, (C_{u_h} - \bar{C}_{(u, u_h)})(u_h - u))_{\partial\Omega} \\
&\quad - (B_{u_h}^* v_h - h, C_{u_h}(u_h - u))_{\partial\Omega} \\
&\quad + (v_h, (L_{u_h} - \bar{L}_{(u, u_h)})(u_h - u)) \\
&\quad + (C_{u_h}^* v_h, (B_{u_h} - \bar{B}_{(u, u_h)})(u_h - u))_{\partial\Omega}.
\end{aligned}$$

In the final result, the first line is the adjoint correction term taking into account the residual errors in satisfying both the p.d.e. and the boundary conditions. The other lines are the remaining errors, which include the consequences of nonlinearity in L, B, C and g as well as residual errors in approximating the adjoint problem.

If the solution errors for the nonlinear primal problem and the linear adjoint problem are of the same order, and they are both sufficiently smooth that the corresponding residual errors are also of the same order, then the order of accuracy of the functional approximation after making the adjoint correction is twice the order of the primal and adjoint solutions. However, rigorous *a priori* and *a posteriori* analysis of the remaining errors is much harder than in the linear case [49] and practical *a posteriori* error bounds have yet to be obtained for the quasi-1D and 2D Euler equations.

4.3 Quasi-1D Euler equations

The steady quasi-1D Euler equations for the flow of an ideal compressible fluid in a variable area duct are

$$\frac{d}{dx}(AF) - \frac{dA}{dx}P = 0,$$

where $A(x)$ is the cross-sectional area of the duct and U , F and P are defined as

$$U = \begin{pmatrix} \rho \\ \rho q \\ \rho E \end{pmatrix}, \quad F = \begin{pmatrix} \rho q \\ \rho q^2 + p \\ \rho q H \end{pmatrix}, \quad P = \begin{pmatrix} 0 \\ p \\ 0 \end{pmatrix}.$$

Here ρ is the density, q is the velocity, p is the pressure, E is the total energy and H is the stagnation enthalpy. The system is closed by the equation of state for an ideal gas.

The functional of interest is the ‘lift’

$$J = \int p \, dx.$$

The Fréchet derivative operator is

$$L_u \tilde{u} \equiv \frac{d}{dx} \left(A \frac{\partial F}{\partial u} \tilde{u} \right) - \frac{dA}{dx} \frac{\partial P}{\partial u} \tilde{u},$$

and therefore the corresponding adjoint equations are

$$L_u^* v \equiv -A \left(\frac{\partial F}{\partial u} \right)^T \frac{dv}{dx} - \frac{dA}{dx} \left(\frac{\partial P}{\partial u} \right)^T v = \left(\frac{\partial p}{\partial u} \right)^T.$$

The equations are approximated using a standard second order finite volume method with characteristic smoothing on a uniform computational grid. The linear adjoint problem is approximated by the so-called ‘continuous’ method, in which one discretises the analytic adjoint equations on the same uniform grid as the flow solution [2, 35]. In the alternative ‘discrete’ approach, one starts with the discretised nonlinear flow equations, linearises them and then uses the transpose of the linear matrix as the discrete adjoint operator [18]. Previous research has shown that both approaches produce approximate solutions which converge to the analytic adjoint solution, which has been determined in closed form for the quasi-1D Euler equations [30].

The approximate solution $u_h(x)$ is constructed from the discrete flow solution by cubic spline interpolation of the nodal values of the three components of the state vector U . Similarly, the approximate adjoint solution $v_h(x)$ is obtained by cubic spline interpolation of the nodal values of the three components of the discrete adjoint solution. The integrals that form the base value for the functional and the adjoint correction are approximated by 3-point Gaussian quadrature.

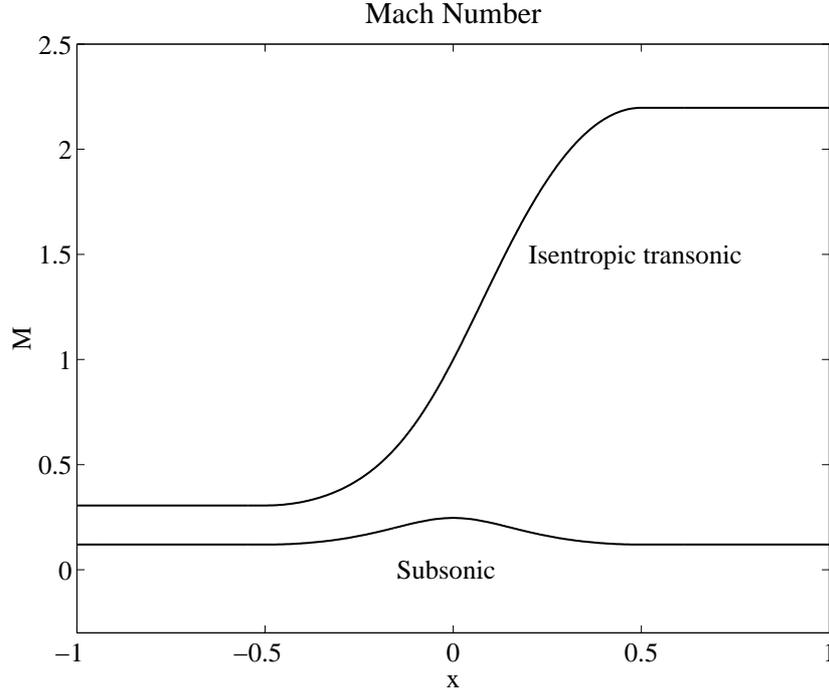


Figure 9: Mach number distributions for quasi-1D Euler equation test cases.

4.3.1 Subsonic flow

The first case is smooth subsonic flow in a converging-diverging duct corresponding to the Mach number distribution depicted in Figure 9. Figure 10 shows the error convergence for the computed functional. The superimposed lines of slope -2 and -4 show that the base error is second order whereas the error in the corrected functional is fourth order. This is in agreement with an *a priori* error analysis [49] that proves that $u_h - u$, $v_h - v$ and their first derivatives are all $O(h^2)$ for the particular finite volume scheme that is used, and hence the error in the corrected functional is $O(h^4)$.

4.3.2 Isentropic transonic flow

Figure 11 shows the error convergence for a transonic flow in a converging-diverging duct corresponding to the Mach number distribution of Figure 9. The flow is subsonic at the inflow boundary and upstream of the throat (located at $x = 0$), and supersonic downstream of the throat and at the outflow boundary. Again the results show that the base error is second order while the remaining error after the adjoint correction is fourth order, even though there is logarithmic singularity in the adjoint solution at the throat [30].

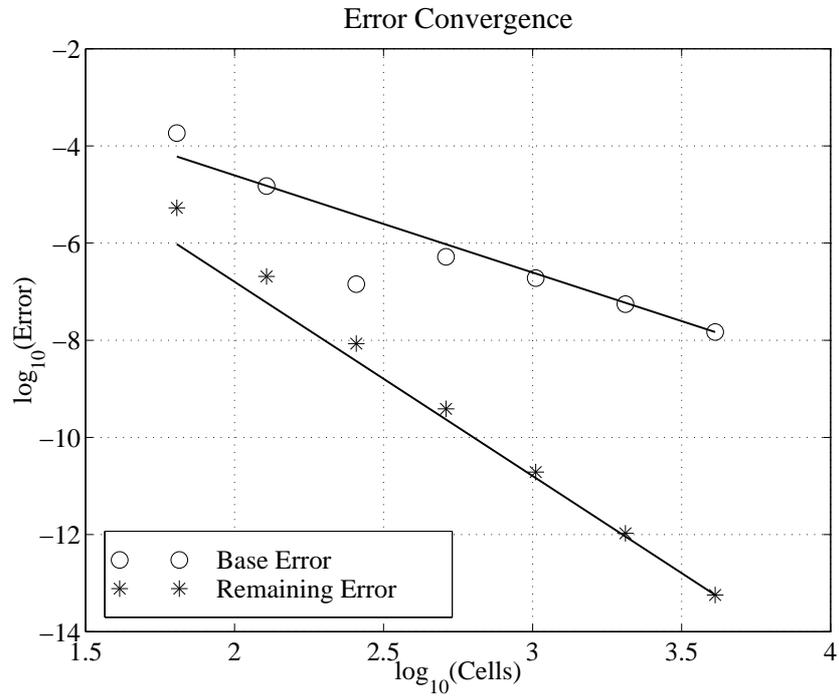


Figure 10: Error convergence for quasi-1D subsonic flow.

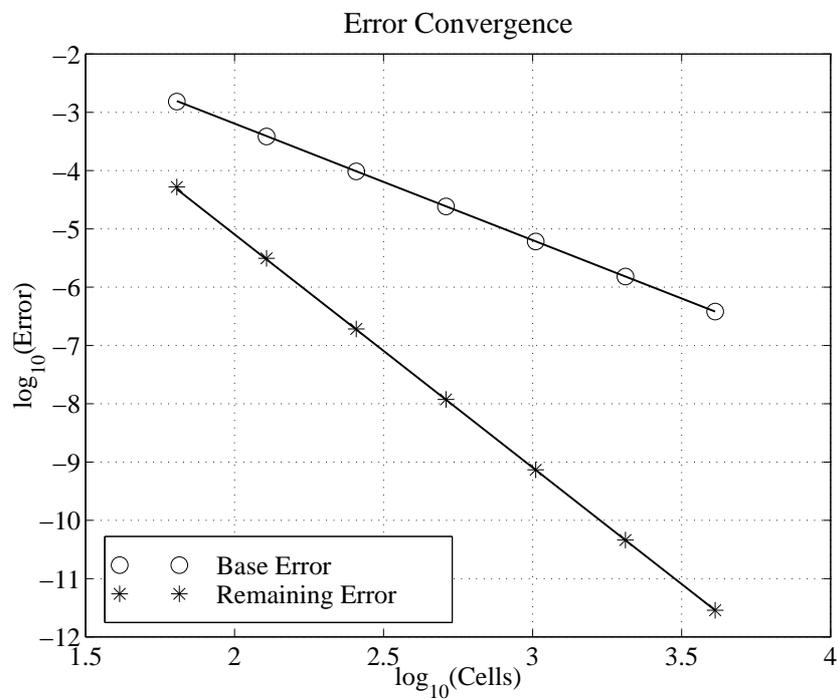


Figure 11: Error convergence for quasi-1D shock-free transonic flow.

4.4 Nonlinear thermal diffusion

The computational domain for this problem is the circular annulus $1 \leq r \leq 3$, and the p.d.e. is the nonlinear diffusion equation

$$\nabla \cdot (u \nabla u) = 0,$$

subject to the requirement that u is positive. Dirichlet boundary conditions are specified at the inner and outer boundaries so as to agree with the analytic solution

$$u(r, \theta) = \left(1 + \left(\frac{r}{4} - \frac{1}{r} \right) \cos \theta \right)^{1/2}.$$

The functional of interest is

$$J(u) = \int_0^{2\pi} \left. \frac{\partial u}{\partial n} \right|_{r=1} d\theta,$$

and the corresponding dual problem is

$$L_u^* v \equiv u \nabla^2 v = 0,$$

with Dirichlet boundary conditions of $1/u$ and 0 on the inner and outer boundaries, respectively.

The primal and dual solutions shown in Figure 12 are obtained by a bi-linear Galerkin finite element formulation using 3×3 Gaussian quadrature to evaluate the mass and stiffness matrices. The nonlinear equations are solved using a full approximation scheme multigrid method. Bi-cubic spline interpolation and 3×3 Gaussian quadrature are then used to calculate the functional with and without the adjoint correction.

The error in the functional is shown in Figure 13 on a log-log plot versus the square root of the total number of cells; this is a measure of h^{-1} , the inverse of the average mesh spacing. The superimposed lines of slopes -2 and -4 show second order accuracy for the basic finite element solution and fourth order accuracy after the inclusion of the adjoint error correction. For a 128×32 mesh, the error decreases by a factor of more than 10^5 .

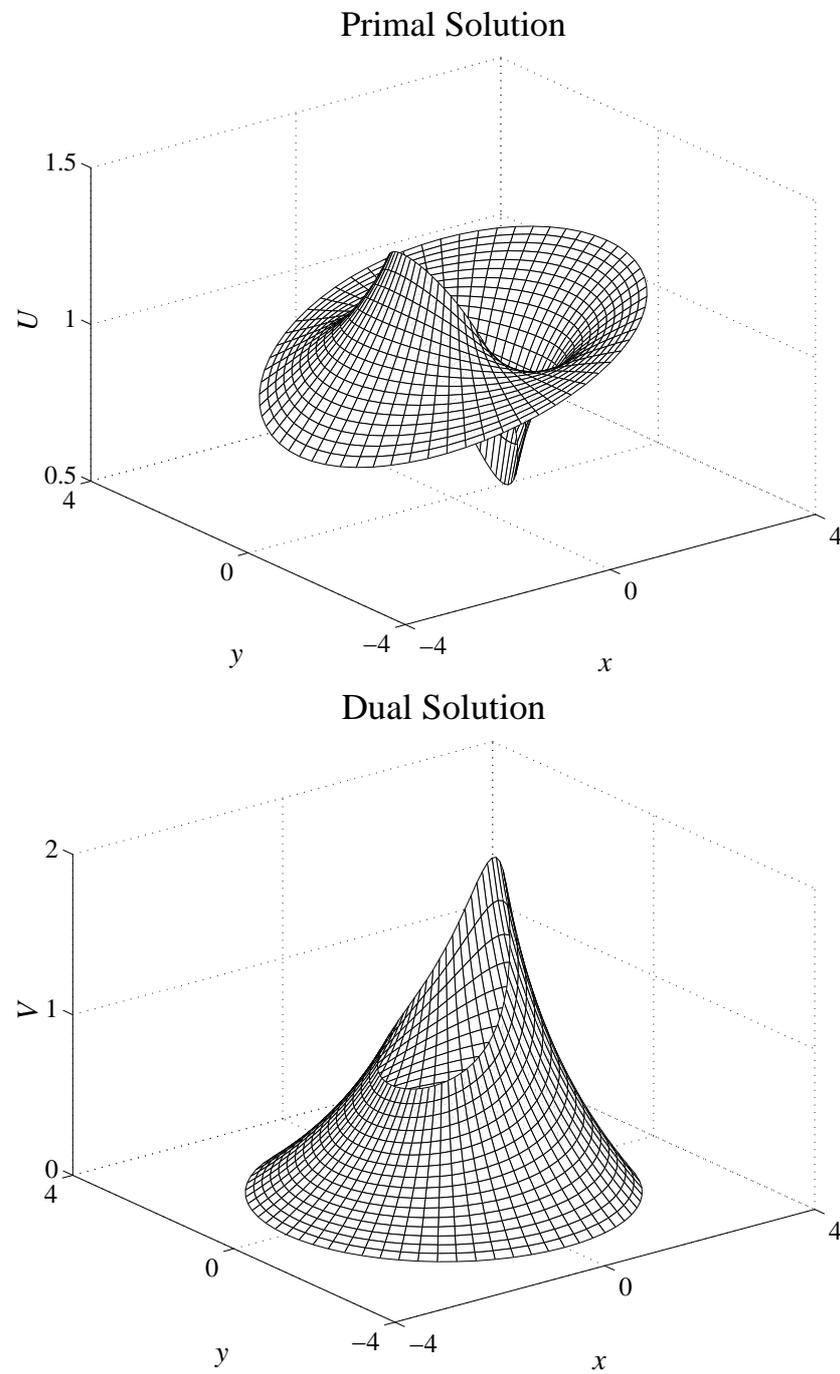


Figure 12: The primal and solutions for a 2D nonlinear thermal diffusion problem.

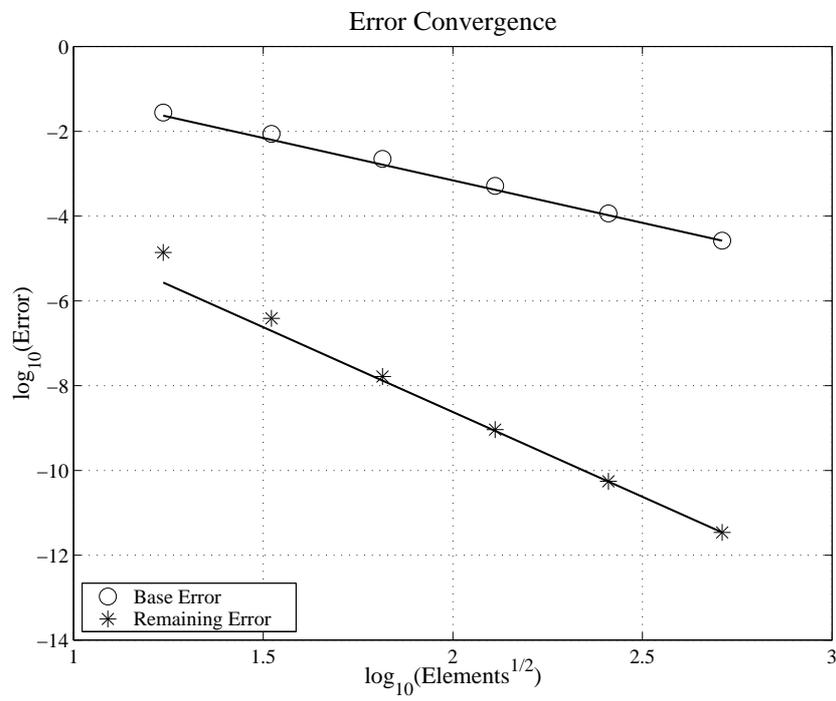


Figure 13: Error convergence of the boundary functional for a 2D nonlinear thermal diffusion problem.

5 Optimal grid adaptation

In this section we very briefly outline a number of possible strategies for grid adaptation based on *a posteriori* error estimates for output functionals. In each case we express reservations about the justification of using the strategy, while recognising that in the end an acceptable justification may be that it provides a robust refinement criterion in practice.

5.1 Option 1: magnitude of correction term

For nonlinear adjoint error correction, the dominant correction term is the interior integral inner product $(v_h, N(u_h))$, which can be expressed as a sum of contributions from each cell in the domain

$$(v_h, N(u_h)) \equiv \sum_{\alpha} (v_h, N(u_h))_{\alpha}.$$

One adaptive strategy is to subdivide those cells for which

$$(v_h, N(u_h))_{\alpha}$$

is bigger than some tolerance.

Müller and Giles have tried this approach for subsonic and transonic airfoil calculations [41], but the results are little better than using an *ad hoc* method based on pressure differences across each cell.

One questionable aspect of this strategy is that the purpose of the adjoint error correction is to evaluate and correct for this term, so what is the justification for trying to make it small? Is it not better to try to make the remaining error small?

5.2 Option 2: estimated remaining error term

After making the adjoint error correction, the main remaining error term can be expressed as

$$(v - v_h, N(u_h)).$$

The difficulty with this expression is that the analytic adjoint solution v is not known. One option therefore is to estimate it and then adapt those cells in which

$$(v - v_h, N(u_h))_{\alpha}$$

is greater than some tolerance.

The problem is how to estimate v . Rannacher *et al* [12, 15] use a quadratic reconstruction to estimate v , having used a piecewise linear finite element solution for v_h . However, if the quadratic reconstruction is a better approximation to v than the piecewise linear one, might it not be better to use the quadratic reconstruction as the approximate solution v_h and thereby get a more accurate adjoint error correction?

It might appear that another possible criticism of this approach is that it assumes that the solution error $v - v_h$ is primarily a local interpolation error, whereas it may be

due to truncation errors in an entirely different part of the grid. However, for a Galerkin finite element method, because of orthogonality the quantity

$$(v - v_h, N(u_h))$$

has the same value for *any* v_h in the appropriate finite element space. Therefore, it is permissible to consider a different v_h which is an interpolant of v , so $v - v_h$ is then an interpolation error which can be estimated using the computed adjoint solution.

The approach used by Venditti & Darmofal [61] is an extension of that used by Rannacher *et al.* An alternative, approximately equivalent form for the dominant part of the remaining error is

$$(R_h, u - u_h),$$

where $R_h \equiv L_{u_h}^* v_h - g(u_h)$ is the residual error in satisfying the adjoint p.d.e. Therefore, they adapt any cell in which the sum

$$|(v - v_h, N(u_h))|_\alpha + |(R_h, u - u_h)|_\alpha$$

is greater than some threshold. The analytic solutions u and v are again approximated by a higher order reconstruction. Because they use a finite volume method to calculate u_h and v_h , the replacement of $v - v_h$ by a local interpolation error does not have the same theoretical justification as with the Galerkin method. On the other hand, it does seem an excellent idea to take into account the residual errors of the adjoint problem, and they do use the more accurate reconstruction of the approximate solution to obtain the correction to the value of the functional. The numerical results they obtain are very good; Figure 14, taken from [61], illustrates the results they obtain for a three-element airfoil calculation. It shows the adapted grid obtained using their refinement criterion, as well as the improved accuracy of the lift prediction with and without adjoint correction, compared to a sequence of uniformly-refined grids.

5.3 Option 3: coarse grid error estimates

Using the residual errors from both the original and adjoint problems, the dominant remaining error can be expressed as

$$(L_{u_h}^{*-1} R_h, N(u_h)) \approx (R_h, L_{u_h}^{-1} N(u_h))$$

The problem with using this in an adaptive approach is that $L_{u_h}^{-1}$ is a global operator. However, it might be possible to use a coarse grid to approximately evaluate $L_{u_h}^{-1} N(u_h)$ and $L_{u_h}^{*-1} R_h$, and then adapt in any cell α for which

$$|(L_{u_h}^{*-1} R_h, N(u_h))|_\alpha + |(R_h, L_{u_h}^{-1} N(u_h))|_\alpha$$

is bigger than some tolerance. This approach may also give a useful *a posteriori* bound on the total remaining error.

The criticism that can be levelled at this idea is that it requires the calculation of the quantity $L_{u_h}^{-1} N(u_h)$ which is essentially a defect correction. In that case, is it not better

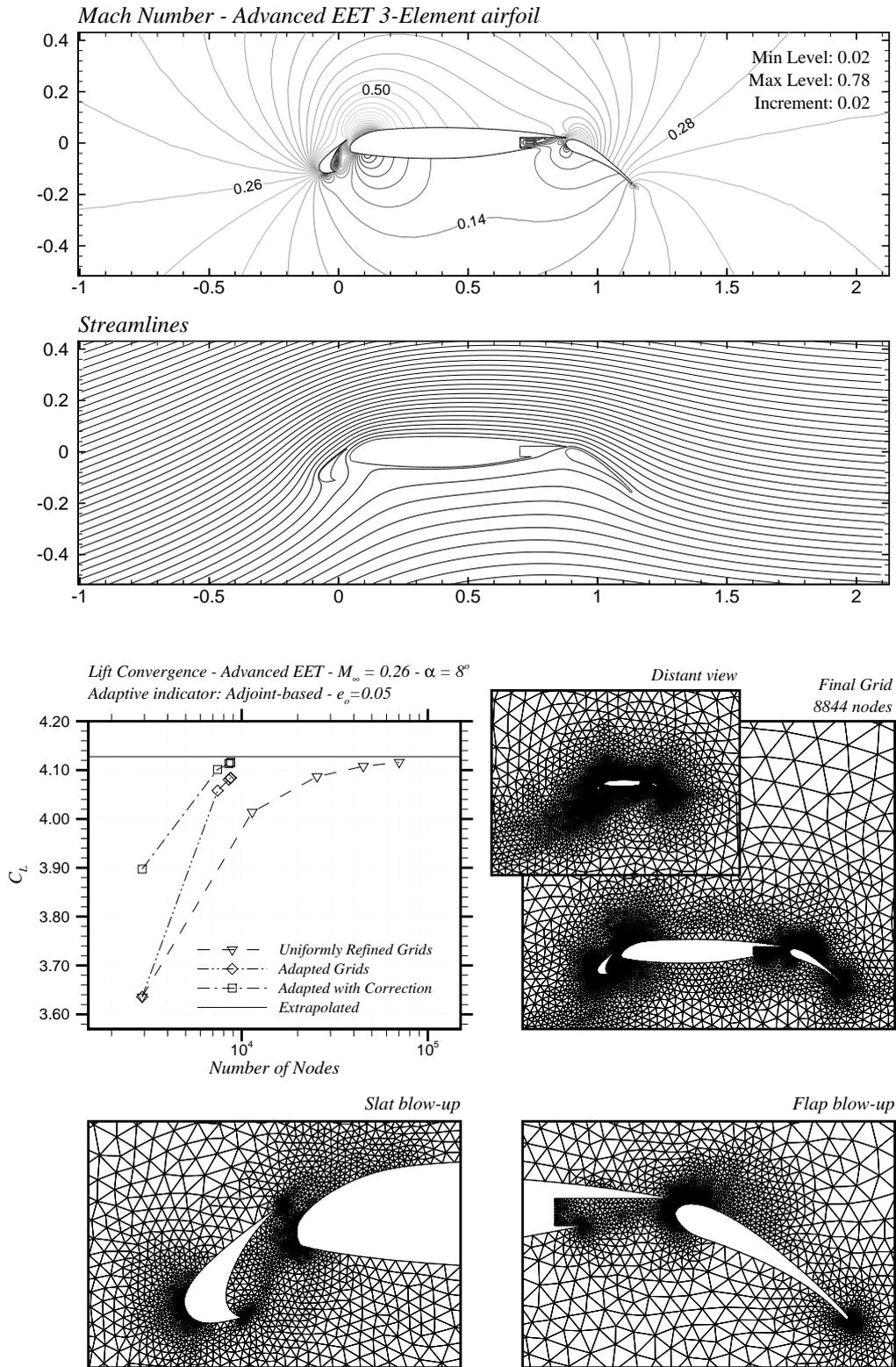


Figure 14: Venditti / Darmofal test case (Figures 2 and 3 from [61]).

to use this to further improve the accuracy of the solution and hence the functional, rather than bound the remaining error?

Yet again, it appears there is a choice to be made between improved accuracy or a tight bound. If we choose the improved accuracy then we can still use the suggested measure as a refinement criterion, but we are refining based on the estimated magnitude of one part of the corrected error, rather than on the estimated remaining error. It may still prove to be a useful adaptation criterion, but its rational basis has been compromised.

5.4 Multiple functionals

A last comment concerns the situation in which there are several different functionals of concern, such as lift, drag and pitching moment. How should all of these be incorporated into the adaptation criterion?

One could perform a separate adjoint calculation for each. This would be necessary if one wished to perform an adjoint error correction to improve the accuracy of each of the output functionals. Alternatively, following options 1 and 2 above, one could adapt whenever

$$V \int_{\alpha} |N(u_h)|_{\alpha} dA$$

exceeds some threshold, with V representing the magnitude of the typical adjoint solution (big near the airfoil, tending to zero far away) if following option 1, or a corresponding interpolation error estimate for option 2.

6 Future research directions

In this concluding section, we offer some thoughts about directions for future research. Some of these are specific to adjoint error correction, but most apply more generally to the subject of *a posteriori* error analysis and grid adaptation for functional outputs.

6.1 Grid adaptation

As indicated in the previous section, grid adaptation remains a topic for future research. Being practical, there is no need for a “perfect” adaptive strategy, even if one could define what that would be. The important criterion for success is that the strategy should be robust and produce consistently good results for a wide variety of problems. It is also important that for problems with singularities, for which it is known that the standard order of accuracy (expressed in terms of the number of nodes/cells used) can be recovered with the appropriate degree of local grid resolution, the adaptive strategy should automatically accomplish this.

Two other adaptation issues to be addressed are anisotropic refinement and grid redistribution. Currently, most adaptive strategies use grid refinement, adding additional nodes/cells through an isotropic refinement process that locally refines equally in each direction, giving cells with a bounded aspect ratio. This is good for many applications, but far from ideal for others.

One example is the inviscid flow around a wing. Here the grid resolution normal to the leading edge needs to be much finer than the spanwise resolution. In this case, anisotropic refinement is probably the best solution. This means adding nodes in such a way that the resolution normal to the leading edge is greater than in the spanwise resolution. The question this poses is how to decide which direction requires additional resolution? There are already *ad hoc* refinement methods that address this. The challenge will be to extend the *a posteriori* adjoint-based refinement indicators to give a more quantitative answer to this question.

Another more extreme example of the need for anisotropic resolution is a boundary layer on a wing, where there is clearly a need for much better resolution across the boundary layer than in the other two directions. In this case, the best solution may well be grid redistribution, moving existing grid nodes to provide the resolution where it is needed. Again there are good existing *ad hoc* methods for doing this and the challenge is how to develop new methods using *a posteriori* error estimates.

6.2 Asymptotic error bounds

Ideally, we would like to have our cake and eat it too! We would like to use smooth reconstruction and adjoint error correction to generate extremely accurate functional values, and at the same time still be able to bound the remaining error with bounds that are at least asymptotically valid and fairly tight.

As suggested in the previous section, this may be unachievable. There may be a choice to be made between using smooth reconstructed solutions for defect and/or adjoint error correction, or for tight error bounds. If so, then our preference would be for the increased accuracy. However, this definitely merits further research.

6.3 Smooth reconstruction on unstructured grids

One key issue is going to be the smooth reconstruction of approximate solutions in multiple dimensions on unstructured grids. On a structured grid, cubic spline interpolation can be used in each direction, but on an unstructured grid there is a need for a suitable generalisation of cubic spline interpolation to produce a reconstructed solution of sufficient smoothness.

Venditti & Darmofal have achieved some success with a piecewise quadratic reconstruction using least squares minimization in the H_1 Sobolev norm [61]. In unpublished research, this local approximation has been sufficient for them to obtain results doubling the order of accuracy of functional outputs from a Galerkin approximation of a convection-diffusion equation.

However, there is reason to believe that in general the solution at the nodes of an unstructured grid may not be very smooth. In particular, the solution error,

$$e_i = u_i - u(x_i)$$

may not be very smooth, and therefore even if $e_i = O(h^2)$, any interpolation may lead to a reconstruction error

$$e(x) = u_h(x) - u(x)$$

for which $\nabla e = O(h)$. This loss of accuracy because of the loss of smoothness would completely negate the ability of the adjoint error correction to improve upon the accuracy of finite element solutions.

To avoid this, it seems likely that it will be necessary to use some form of smoothed spline reconstruction, in which the spline does not interpolate the nodal values, but instead compromises between the twin objectives of matching the computed data and maintaining smoothness. For example, if U_h is a second order accurate piecewise linear finite element solution, then the reconstruction u_h could be defined by some suitable approximation to the equation

$$h^2 \nabla^2(\nabla^2 u_h) + u_h - U_h = 0.$$

The purpose of the bi-harmonic term is to ensure the smoothness of the solution. The h^2 weighting ensures that this is not achieved at the expense of sacrificing the second order accuracy of the underlying solution.

6.4 Shocks

One last challenge we wish to highlight is the problem of shocks and other discontinuities.

With the quasi-1D Euler equations, it can be proved that with an appropriate conservative formulation, and a numerical discretisation that is second order accurate when the solution is smooth, the accuracy of output functionals such as the integrated pressure is also second order [21]. However, numerical evidence suggests this is not the case in multiple dimensions, and instead there is a error in quantities such as the lift on a transonic airfoil that is proportional to the local grid spacing at the shock. Thus, to get even second order accuracy would require anisotropic grid adaptation so that the grid spacing at the shock is $O(h^2)$, with h here being the average grid spacing in the rest of the grid.

Application of adjoint error correction ideas raises another problem. The reconstructed solution will be continuous, whereas the true solution is discontinuous. Therefore, it is unavoidable that there is an $O(1)$ error in the approximate solution at the shock. This violates the whole basis for the adjoint error correction since it relies on a linearisation of the nonlinear equations that is valid only for small perturbations.

We are currently working on this problem. Our approach is to numerically approximate a “viscous” shock with the level of viscosity being $O(h^2)$. The adjoint error correction then has to correct for the numerical error in approximating the viscous shock, plus the analytic error in using the viscous shock problem to approximate the inviscid shock problem. This latter part requires the use of matched asymptotic expansions to understand that to leading order there is a linear dependence of integral functionals on the level of viscosity. This error can be compensated for by using the viscous adjoint to give the sensitivity of the lift to a change in the level of the viscosity.

Acknowledgements

In the course of the research discussed in these notes, we have benefited from stimulating discussions with many people, including in particular Endre Süli, Paul Houston, David Darmofal and Rolf Rannacher.

This research was supported by EPSRC grants GR/K91149 and GR/L95700, and by NASA/Ames Cooperative Agreement No. NCC2-5431.

References

- [1] W.K. Anderson and D.L. Bonhaus. Airfoil design on unstructured grids for turbulent flows. *AIAA Journal*, 37(2):185–191, 1999.
- [2] W.K. Anderson and V. Venkatakrisnan. Aerodynamic design optimization on unstructured grids with a continuous adjoint formulation. *Computers & Fluids*, 28(4-5):443–480, 1999.
- [3] C.H. Bischof, A. Carle, P.D. Hovland, P. Khademi, and A. Mauer. ADI-FOR 2.0 User’s Guide (Revision D). Technical Report 192, Mathematics and Computer Science Division, Argonne National Laboratory, 1998.
- [4] J.W. Barrett and C.M. Elliott. Total flux estimates for a finite-element approximation of elliptic equations. *IMA Journal of Numerical Analysis*, 7:129–148, 1987.
- [5] O. Baysal and M. Eleshaky. Aerodynamic design optimization using sensitivity analysis and computational fluid dynamics. *AIAA Journal*, 30(3):718–725, 1992.
- [6] R. Becker, H. Kapp, and R. Rannacher. Adaptive finite element methods for optimal control of partial differential equations: basic concepts. *SIAM Journal of Control and Optimization*, 39:113–132, 2000.
- [7] I. Babuška and A. Miller. The post-processing approach in the finite element method – Part 1: calculation of displacements, stresses and other higher derivatives of the displacements. *International Journal for Numerical Methods in Engineering*, 20:1085–1109, 1984.
- [8] I. Babuška and A. Miller. The post-processing approach in the finite element method – part 2: the calculation of stress intensity factors. *International Journal for Numerical Methods in Engineering*, 20:1111–1129, 1984.
- [9] J.W. Barrett, G. Moore, and K.W. Morton. Optimal recovery in the finite element method, part 2: Defect correction for ordinary differential equations. *IMA Journal of Numerical Analysis*, 8:527–540, 1988.
- [10] I. Babuška and W.C. Rheinboldt. *A posteriori* error estimates for the finite element method. *International Journal for Numerical Methods in Engineering*, 12:1597–1615, 1978.
- [11] I. Babuška and W.C. Rheinboldt. Error estimates for adaptive finite element computations. *SIAM Journal of Numerical Analysis*, 15(4):736–754, 1978.
- [12] R. Becker and R. Rannacher. A feed-back approach to error control in finite element methods: basic analysis and examples. *East-West Journal of Numerical Mathematics*, 4:237–264, 1996.

- [13] R. Becker and R. Rannacher. Weighted *a posteriori* error control in finite element methods. In *et al* H.G. Block, editor, *Proceedings of ENUMATH-97*, pages 621–637. World Scientific Publishing, 1998.
- [14] R. Braack and R. Rannacher. Adaptive finite element methods for low Mach number flows with chemical reactions. VKI Lecture Series 1999-03, 1999.
- [15] R. Becker and R. Rannacher. An optimal control approach to error control and mesh adaptation. In A. Iserles, editor, *Acta Numerica 2001*. Cambridge University Press, 2001.
- [16] A. Carle, M. Fagan, and L.L. Green. Preliminary results from the application of automated code generation to CFL3D. AIAA Paper 98-4807, 1998.
- [17] J. Elliott. *Aerodynamic optimization based on the Euler and Navier-Stokes equations using unstructured grids*. PhD thesis, MIT Dept. of Aero. and Astro., 1998.
- [18] J. Elliott and J. Peraire. Practical 3D aerodynamic design and optimization using unstructured meshes. *AIAA Journal*, 35(9):1479–1485, 1997.
- [19] C. Faure and Y. Papegay. Odyssee User’s Guide version 1.7. Technical Report RT-0224, INRIA, Sophia-Antipolis, 1998.
- [20] M.B. Giles, M.C. Duta, and J.-D. Müller. Adjoint code developments using the exact discrete approach. AIAA Paper 2001-2596, 2001.
- [21] M.B. Giles. Analysis of the accuracy of shock-capturing in the steady quasi-1D Euler equations. *International Journal of Computational Fluid Dynamics*, 5(2):247–258, 1996.
- [22] M.B. Giles. Aerospace design: a complex task. Lecture notes for VKI Lecture Course on Inverse Design. Technical Report NA97/07, Oxford University Computing Laboratory, Wolfson Building, Parks Road, Oxford, OX1 3QD, 1997.
- [23] M.B. Giles. Defect and adjoint error correction. Invited paper at the ICCFD conference, Kyoto, 2000.
- [24] M.B. Giles. On the iterative solution of adjoint equations. In G. Corliss, C. Faure, A. Griewank, L. Hascoët, and U. Naumann, editors, *Automatic Differentiation: From Simulation to Optimization*. Springer, 2001.
- [25] R. Giering and T. Kaminski. Recipes for adjoint code construction. *ACM Transactions on Mathematical Software*, 24(4):437–474, 1998.

- [26] M.B. Giles, M.G. Larson, M. Levenstam, and E. Süli. Adaptive error control for finite element approximations of the lift and drag in a viscous flow. Technical Report NA 97/06, Oxford University Computing Laboratory, Wolfson Building, Parks Road, Oxford, OX1 3QD, 1997.
- [27] M.B. Giles and N.A. Pierce. Adjoint equations in CFD: duality, boundary conditions and solution behaviour. AIAA Paper 97-1850, 1997.
- [28] M.B. Giles and N.A. Pierce. Superconvergent lift estimates using the adjoint Euler equations. In *Proceedings of Third Asian CFD Conference*, 1998.
- [29] M.B. Giles and N.A. Pierce. Improved lift and drag estimates using adjoint Euler equations. AIAA Paper 99-3293, 1999.
- [30] M.B. Giles and N.A. Pierce. Analytic adjoint solutions for the quasi-one-dimensional Euler equations. *Journal of Fluid Mechanics*, 426:327–345, 2001.
- [31] M.B. Giles and N.A. Pierce. An introduction to the adjoint approach to design. *Flow, Turbulence and Control*, to appear, 2001.
- [32] A. Griewank. *Evaluating derivatives : principles and techniques of algorithmic differentiation*. SIAM, 2000.
- [33] P. Houston, R. Rannacher, and E. Süli. A posteriori error analysis for stabilised finite element approximations of transport problems. *Computer Methods in Applied Mechanics and Engineering*, 190(11-12):1483–1508, 2000.
- [34] A. Jameson. Aerodynamic design via control theory. *Journal of Scientific Computing*, 3:233–260, 1988.
- [35] A. Jameson. Optimum aerodynamic design using control theory. In M. Hafez and K. Oshima, editors, *Computational Fluid Dynamics Review 1995*, pages 495–528. John Wiley & Sons, 1995.
- [36] C. Johnson. On computability and error control in CFD. *International Journal for Numerical Methods in Fluids*, 20(8-9):777–788, 1995.
- [37] A. Jameson, N. Pierce, and L. Martinelli. Optimum aerodynamic design using the Navier-Stokes equations. *Journal of Theoretical and Computational Fluid Mechanics*, 10:213–237, 1998.
- [38] C. Johnson, R. Rannacher, and M. Boman. Numerics and hydrodynamic stability – toward error control in computational fluid dynamics. *SIAM Journal of Numerical Analysis*, 32(4):1058–1079, 1995.
- [39] V.M. Korivi, A.C. Taylor III, and G.W. Hou. Sensitivity analysis, approximate analysis and design optimization for internal and external viscous flows. AIAA Paper 91-3083, 1991.

- [40] B. Koren. Defect correction and multigrid for an efficient and accurate computation of airfoil flows. *Journal of Computational Physics*, 76, 1988.
- [41] J.-D. Müller and M.B. Giles. Solution adaptive mesh refinement using adjoint error analysis. AIAA Paper 2001-2550, 2001.
- [42] B. Mohammadi. Optimal shape design, reverse mode of automatic differentiation and turbulence. AIAA Paper 97-0099, 1997.
- [43] B. Mohammadi and O. Pironneau. Mesh adaptation and automatic differentiation in a CAD-free framework for optimal shape design. *International Journal for Numerical Methods in Fluids*, 30(2):127–136, 1999.
- [44] P. Monk and E. Süli. The adaptive computation of far field patterns by a posteriori error estimates of linear functionals. *SIAM Journal of Numerical Analysis*, 36(1):251–274, 1998.
- [45] E. Nielsen and W.K. Anderson. Aerodynamic design optimization on unstructured meshes using the Navier-Stokes equations. *AIAA Journal*, 37(11):957–964, 1999.
- [46] J.C. Newman, A.C. Taylor, R.W. Barnwell, P.A. Newman, and G. J.-W. Hou. Overview of sensitivity analysis and shape optimization for complex aerodynamic configurations. *Journal of Aircraft*, 36(1):87–96, 1999.
- [47] J.T. Oden and S. Prudhomme. New approaches to error estimation and adaptivity for the Stokes and Oseen equations. *International Journal for Numerical Methods in Fluids*, 31(1):3–15, 1999.
- [48] J.T. Oden and S. Prudhomme. Goal-oriented error estimation and adaptivity for the finite element method. *Computers and Mathematical Applications*, 41(5-6):735–756, 2000.
- [49] N.A. Pierce and M.B. Giles. Adjoint recovery of superconvergent functionals from approximate solutions of partial differential equations. Technical Report NA98/18, Oxford University Computing Laboratory, 1998.
- [50] N.A. Pierce and M.B. Giles. Adjoint recovery of superconvergent functionals from PDE approximations. *SIAM Review*, 42(2):247–264, 2000.
- [51] J. Peraire and A.T. Patera. Bounds for linear-functional outputs of coercive partial differential equations: local indicators and adaptive refinement. In P. Ladeveze and J.T. Oden, editors, *New Advances in Adaptive Computational Methods in Mechanics*. Elsevier, 1997.
- [52] P. Peraire and A. Patera. Asymptotic a posteriori finite element bounds for the outputs of noncoercive problems: the Helmholtz and Burgers equations. *Computer Methods in Applied Mechanics and Engineering*, 171(1-2):77–86, 1999.

- [53] M. Paraschivoiu, J. Peraire, and A. Patera. A posteriori finite element bounds for linear-functional outputs of elliptic partial differential equations. *Computer Methods in Applied Mechanics and Engineering*, 150(1-4):289–312, 1997.
- [54] R. Rannacher. Adaptive Galerkin finite element methods for partial differential equations. *Journal of Computational and Applied Mathematics*, 1-2:205–233, 2000.
- [55] G. Strang and G. Fix. *An Analysis of the Finite Element Method*. Prentice-Hall, 1973.
- [56] R.D. Skeel. A theoretical framework for proving accuracy results for deferred corrections. *SIAM Journal of Numerical Analysis*, 19:171–196, 1981.
- [57] H.J. Stetter. The defect correction principle and discretization methods. *Numerische Mathematik*, 29:425–443, 1978.
- [58] E. Süli. A *posteriori* error analysis and adaptivity for finite element approximations of hyperbolic problems. In D. Kröner, M. Ohlberger, and C. Rohde, editors, *An Introduction to Recent Developments in Theory and Numerics for Conservation Laws*, volume 5 of *Lecture Notes in Computational Science and Engineering*, pages 123–194. Springer-Verlag, 1998.
- [59] S. Ta’asan, G. Kuruwila, and M.D. Salas. Aerodynamic design and optimization in one shot. AIAA Paper 92-0025, 1992.
- [60] D. Venditti and D. Darmofal. Adjoint error estimation and grid adaptation for functional outputs: application to quasi-one-dimensional flow. *Journal of Computational Physics*, 164:204–227, 2000.
- [61] D. Venditti and D. Darmofal. A grid adaptive methodology for functional outputs of compressible flow simulations. AIAA Paper 01-2659, 2001.