Computers and Fluids 148 (2017) 26–38

Contents lists available at ScienceDirect

Computers and Fluids

journal homepage: www.elsevier.com/locate/compfluid

Robust and efficient adjoint solver for complex flow conditions

Shenren Xu*, Sebastian Timme

School of Engineering, University of Liverpool, Liverpool L69 3GH, United Kingdom

ARTICLE INFO

Article history:
Received 24 May 2016
Revised 7 February 2017
Accepted 13 February 2017
Available online 20 February 2017

Keywords:
Adjoint method
Krylov solvers
Subspace recycling
GCRO–DR
RANS equations

ABSTRACT

A key step in gradient-based aerodynamic shape optimisation using the Reynolds-averaged Navier–Stokes equations is to compute the adjoint solution. Adjoint equations inherit the linear stability and the stiffness of the nonlinear flow equations. Therefore for industrial cases with complex geometries at off-design flow conditions, solving the resulting stiff adjoint equation can be challenging. In this paper, Krylov subspace solvers enhanced by subspace recycling and preconditioned with incomplete lower-upper factorisation are used to solve the stiff adjoint equations arising from typical design and off-design conditions. Compared to the baseline matrix-forming adjoint solver based on the generalized minimal residual method, the proposed algorithm achieved memory reduction of up to a factor of two and convergence speedup of up to a factor of three, on industry-relevant cases. These test cases include the DLR-F6 and DLR-F11 configurations, a wing-body configuration in pre-shock buffet and a large civil aircraft with mesh sizes ranging from 3 to 30 million. The proposed method seems to be particularly effective for the more difficult flow conditions.

1. Introduction

Over the past few decades, adjoint-based aerodynamic shape optimisation using computational fluid dynamics (CFD) has been widely used for the design of automobiles [1], aeroplanes [2–5] and turbomachines [6–8]. It was first proposed in [9] to use the adjoint equations to efficiently compute the design gradient for aerodynamic shape optimisation. The method was later extended to configurations of increasing complexity such as the redesign of the wing of a transonic business jet using Euler equations on multiblock structured meshes [2] as well as for Navier–Stokes equations on unstructured meshes to capture the viscous effect on complex shapes [10,11]. A comprehensive strategy for developing and implementing discrete adjoint methods for aerodynamic shape optimisation problems is presented in [12] and demonstrated in a three-dimensional unstructured Reynolds-averaged Navier–Stokes (RANS) adjoint solver on several cases including a high-lift configuration and a modern transport configuration. The methodology was later extended in [13,14] to include multigrid in the line-implicit adjoint solver for better convergence and applied to the drag-reduction optimisation of a wing body configuration.

With the maturing of the adjoint method, applications nowadays are more focused on realistic configurations under both design and off-design conditions. The increased complexity in both geometry and flow conditions can pose significant computational challenges for the adjoint solver. Flow and adjoint solvers using well-established fixed-point iterations, either explicit or implicit, could have difficulty converging. One such example is reported in [15] for a transonic viscous case with a mesh consisting of 69,000 points with stretched cells in the boundary layer. Similar issues are reported for more realistic cases in [16], where the DLR–TAU adjoint solver is used to optimise the DLR-F6 wing body configuration and the DLR-F11 high-lift configuration. For the DLR-F6 case, side-of-body separation near the trailing edge destabilises fixed-point iteration and recursive projection method (RPM) [17] is applied to stabilise the adjoint. However, RPM fails to stabilise the adjoint for DLR-F11 [16] because the unstable fixed point iteration diverged too fast, and generalised minimal residual method (GMRES) [18] was used to successfully converge the case.

The numerical stiffness discussed above is mainly due to the ill-conditioned coefficient matrix in the adjoint equations. The issue could be alleviated to some extent by using an approximate instead of exact flow Jacobian matrix. Essentially, one is trading accuracy for solver efficiency and robustness. One typical remedy is to use the frozen turbulence assumption when solving the adjoint RANS equations as it is well known that coupling the turbulence equation with the mean flow equation significantly increases the numerical stiffness and sometimes it even destabilises the time marching scheme. The effect of various other approximations of the Jacobian matrix on the gradient accuracy and the optimisation results is investigated in detail in [19].

* Corresponding author.
E-mail addresses: shenren.xu@liverpool.ac.uk, shenren@liv.ac.uk (S. Xu).

http://dx.doi.org/10.1016/j.compfluid.2017.02.012

Alternatively, one could adhere to the exact Jacobian matrix and solve the stiff adjoint equations more efficiently so that the resulting adjoint solution, and consequently the design gradient, remains accurate. To avoid the linear instability issue of any fixed-point iterative solver, Krylov solvers are usually preferred for solving the stiff and marginally stable adjoint equations. It is proposed that the Jacobian-free Newton–Krylov method is preferred for critical aerodynamic simulations and shape optimisation applications where numerical stiffness constantly causes convergence difficulties [20,21]. A few key aspects on the efficient implementation of the method are also highlighted to show that once properly implemented, superior efficiency and reliability can be achieved, compared with other more well-established solution methods such as multi-stage explicit schemes, point or block implicit procedure and implicit factorisation methods. The method has been successfully applied to solve adjoint equations arising from aerodynamic shape optimisaion [22] and error estimation [23].

Krylov solvers are also affected by the conditioning of the system matrix. For example, restarted GMRES could suffer from convergence stagnation for challenging problems unless m is sufficiently large, which would then result in prohibitively high memory overhead. An obvious remedy to alleviate the memory bottleneck of the Krylov solver for difficult cases is to use a stronger preconditioner. For example, a clean wing geometry for the common research model at cruise condition is studied in [24] using a mesh with 28 million points. For this case, incomplete lower-upper (ILU) factorisation with fill-in level of two, i.e., ILU(2), is necessary to effectively precondition GMRES, which is then able to converge with m of 200. Had a weaker preconditioner such as ILU(0) been used, the Krylov solver would have required many more vectors to converge.

The fundamental reason for the convergence stagnation of GMRES(m) is that the restarted subspace is often close to the previous subspace. Generalised conjugate residual with optimal truncation (GCROT) [25], its simplified and flexible variant [26] and generalised conjugate residual with deflated restarting (GCRO-DR) [27] have been proposed to address this shortcoming by recycling a selected subspace from one cycle to the next. The subspace recycling technique allows the solvers to converge without stagnation with much lower memory requirement. GCRO-DR was shown to be effective in both lowering the stagnation memory threshold and accelerating the convergence for large scale linearised aerodynamics analysis [28].

In this paper, we replace the baseline GMRES solver within the DLR-TAU adjoint solver with GCRO-DR. The proposed method is applied to solve the adjoint equations to demonstrate its effectiveness in both reducing memory overhead and accelerating convergence for solving the adjoint equations arising from industry-relevant cases with complex geometries under both design and off-design flow conditions.

The remainder of the paper is organized as follows. The mathematical formulation of the flow and adjoint equations is explained in Section 2. The details of the Krylov solvers are given in Section 3 and the preconditioning technique is discussed in Section 4. The application of the proposed method to five test cases is presented in Section 5. A comprehensive comparison between GMRES, GCROT and GCRO-DR is first given for a small, yet stiff, two-dimensional aerofoil case for a parameter study. Both GCRO-DR and GMRES are then applied to several more realistic three-dimensional industry-relevant cases.

2. Nonlinear flow and adjoint solvers

2.1. Nonlinear flow solver

The DLR–TAU code is a CFD software package widely used as production code in the European aerospace industry as well as a research code for method development [29,30]. The RANS equations are solved with a finite-volume discretisation on unstructured grids with various options of spatial and temporal discretisation schemes and turbulence models. In this paper, the mean flow is by default discretised with the Jameson–Schmidt–Turkel (JST) scheme [31] with matrix dissipation [32], unless stated otherwise. The Spalart–Allmaras model [33] is discretised using first-order accurate Roe scheme [34]. The nonlinear flow equations are pseudo time marched using the first-order backward Euler implicit scheme. At each pseudo time step, agglomeration multigrid is used to accelerate the convergence with lower-upper symmetric-Gauss–Seidel [35] as the multigrid smoother.

2.2. Adjoint solver

The cost function for optimisation $\mathbf{J} := (J_1, J_2, \ldots, J_N)^T$ is a function of the flow solution $\mathbf{U}$, the coordinates of the computational mesh points $\mathbf{X}$ and the design variable $\boldsymbol{\alpha} := (\alpha_1, \alpha_2, \ldots, \alpha_M)^T$. To evaluate the design gradient, the cost function is linearised as

$$\frac{d\mathbf{J}}{d\boldsymbol{\alpha}} = \frac{\partial \mathbf{J}}{\partial \boldsymbol{\alpha}} + \mathbf{v}^T \mathbf{f}$$

where $\mathbf{v}$ is the solution to the adjoint equation

$$\left(\frac{\partial \mathbf{R}}{\partial \mathbf{U}}\right)^T \mathbf{v} = \mathbf{g} \qquad (1)$$

with $\mathbf{f}$ and $\mathbf{g}$ defined as

$$\mathbf{f} := -\frac{\partial \mathbf{R}}{\partial \boldsymbol{\alpha}} \quad \text{and} \quad \mathbf{g}^T := \frac{\partial \mathbf{J}}{\partial \mathbf{U}}$$

and $\mathbf{R}$ is the nonlinear residual vector. Note that the design variables do not appear in the adjoint equation thus Eq. (1) needs to be solved only as many times as the number of cost functions. For aerodynamic applications, the cost functions are usually limited to a handful, such as lift, drag and moment, while the design variables could be many more. The adjoint approach is therefore very efficient.

The adjoint equation is solved using a Jacobian-forming Newton–Krylov approach. The exact flow Jacobian matrix corresponding to the second-order accurate spatial discretisation is computed using the hand-differentiated nonlinear residual subroutine. The Jacobian matrix is stored in block compressed sparse row format, with each block containing a 6-by-6 dense matrix. The Jacobian matrix is then transposed to obtain the coefficient matrix for the adjoint equation. Computing the Jacobian matrix and its transpose are done in parallel with negligible computational time compared to the adjoint solution time for all the cases considered in this work. The right-hand side for each cost function is computed using the linearised subroutine that computes the cost function. No simplification such as frozen turbulence is used in this work so that an exact dual adjoint solution is solved. Once the coefficient matrix and the right-hand side are formed, the resulting large sparse linear system of equations is then solved using ILU preconditioned Krylov solvers, which are explained in detail in the following two sections.