Nonlinear Acceleration of Coupled Fluid-Structure Transient Thermal Problems by Anderson Mixing

V. Ganine^{1*}, N. J. Hills¹ and B. L. Lapworth²

¹*Fluids Research Center, Faculty of Engineering and Physical Sciences,
University of Surrey, Guildford, Surrey GU2 7XH, UK*

²*Rolls-Royce plc, P.O. Box 31, Derby, DE24 8BJ, UK*

SUMMARY

Conjugate heat-transfer problems are typically solved using partitioned methods where fluid and solid subdomains are evaluated separately by dedicated solvers coupled through a common boundary. Strongly coupled schemes for transient analysis require fluid and solid problems to be solved many times each time step until convergence to a steady state. In many practical situations a fairly simple and frequently employed fixed-point iteration process is rather ineffective, it leads to a large number of iterations per time step and consequently to long simulation times. In this article Anderson mixing is proposed as a fixed-point convergence acceleration technique to reduce computational cost of thermal coupled fluid-solid problems. A number of other recently published methods with applications to similar fluid-structure interaction problems are also reviewed and analyzed. Numerical experiments are presented to illustrate relative performance of these methods on a test problem of rotating pre-swirl cavity air flow interacting with a turbine disk. It is observed that performance of Anderson mixing method is superior to that of other algorithms in terms of total iteration counts. Additional computational savings are demonstrated by reusing information from previously solved time steps.

KEY WORDS: Conjugate heat transfer; fluid-structure thermal coupling; multi-physics; fixed-point problems; Anderson mixing

1. INTRODUCTION

Multiphysics simulations of thermal interaction phenomena between fluid flows and solids play an increasingly important role in many engineering applications such as heating of vehicles in hypersonic flow, modeling of heat exchangers and cooling of turbine components in jet engines.

There are many mathematical and algorithmic challenges associated with the modeling of multiphysics systems that exhibit different phenomena at diverse space and time scales. Perhaps the most straightforward algorithmic approach is to represent all the relevant phenomena in multiple domains in a single larger mathematical model, and then to simultaneously solve it within a unique highly specialized solver. This is referred to as a monolithic or direct method [1, 2]. However, monolithic solution algorithms result in inflexible non-modular codes with high developmental costs restricting incorporation of existing single field solvers into a simulation environment.

This contrasts with the so-called partitioned or segregated approach. Similar to the domain decomposition methods, the global system of equations describing a physical problem is split into subdomain problems, e.g. into a fluid and solid problem. Partitioned methods proceed with alternating solution of the subdomains, subject to complementary coupling transmission conditions. It has become a more popular strategy since each physical model usually comes with its own well

*Correspondence to: V. Ganine, Fluids Research Center, Faculty of Engineering and Physical Sciences, University of Surrey, Guildford, Surrey GU2 7XH, UK. E-mail: v.ganine@surrey.ac.uk

developed theory and equations that are better treated separately by more efficient and reliable existing solvers. From the software engineering perspective partitioned methods allow reuse of the existing codes offering increased modularity and flexibility for potential evolution of each solver.

Among the partitioned schemes one can distinguish between strong and loose coupling strategies to enforce continuity of the transmission conditions at the interface after time discretization [3, 4]. With loosely or explicitly coupled systems the sub-problems are solved only once or just a few times per time step. As such, the interface conditions are not exactly satisfied, which may induce instabilities in the numerical scheme. For example, coupled heat transfer problems can be particularly difficult to treat with the explicit partitioned scheme for a certain ratio of thermal conductivity and diffusivity between the adjacent fluid and solid subdomains. Stability analysis of the partitioned schemes for thermally coupled problems can be found in [5, 6].

To circumvent the numerical instabilities of weakly coupled partitioned schemes, the interface conditions can be treated in an implicit fashion by enforcing them exactly at each time step. Those methods are referred to as strongly coupled or implicit. The price to pay is a relatively large number of subiterations at each time step resulting from the solution of a nonlinear algebraic system defined at the interface between subdomains. The ensuing high computational cost remains a critical issue for the partitioned strongly coupled methods.

Many practical coupled fluid-solid problems have the following constraint: the analytical form of the Jacobian matrix is not readily available nor is it rational to compute it. Moreover, the cost of an iteration is dominated by the fluid subdomain solve, which may be extremely high. Different methods for solving large scale nonlinear algebraic systems under the aforementioned constraints have been proposed and investigated recently, more often than not in connection with the actively investigated area of fluid-structure interaction (FSI).

The classical fixed-point iterative process (nonlinear block Gauss-Seidel) applied to FSI problems [7, 8] can be strikingly inefficient, suffering from slow convergence or even failing to converge. Aitken acceleration and other dynamic relaxation methods [9, 10] can be utilized to improve the convergence of the fixed point iterations. An extension of these relaxation methods leads to a class of vector extrapolation algorithms [11, 12, 13]. Alternatively, a time discretized coupled system can be linearized with Newton-like methods [14, 7, 8]. In inexact Newton-Krylov based approaches [15] the Jacobian matrix is not formed explicitly, which requires application of matrix free methods to resolve the linear Jacobian subproblems. Thus, the computational cost of each nonlinear iteration comprises a variable number of inner residual evaluations. Unlike the former, quasi-Newton algorithms do not require extra fluid solver calls, they approximate the Jacobian system directly from the terms of the generated nonlinear vector sequence [16].

Not coincidentally, the convergence behavior of these methods on typical FSI problems is well understood and reported, see e.g. [17], whereas very limited information is available on their performance on thermally coupled problems of mixed type such as fluid-solid systems. Reported examples include coupling of heat transfer models for melt crystal growth applications employing fixed-point iterations [18] and the Newton-Krylov method [19]. Performances of fixed-point iteration procedure and Newton-Krylov method are compared in [20] on a solid model interacting with external gas flow.

The main focus of the present work is on improving numerical efficiency of large scale strongly coupled thermal fluid-solid simulations. Here, the Anderson mixing method [21, 22] is proposed for solution of the nonlinear interface problem. Having its roots in the physics community (modeling of electronic structures), the Anderson family of acceleration methods has been largely overlooked by researchers in other fields. In this paper we show that the method provides an attractive alternative to both vector sequence acceleration methods and matrix-free Newton-Krylov algorithms as well as to the more traditional approach formulated on the estimates of heat transfer coefficient [23, 24]. Its major advantage is a significantly lower total iteration count with convergence rate comparable to the nested iteration Newton-Krylov methods. Since a large sequence of similar problems is typically solved during a time-integration process, it is shown that by incorporating data generated during antecedent time steps, the Anderson method may exhibit additional computational savings.

In the paper several Jacobian-free methods reported in the FSI body of literature are tested on a thermally coupled problem, namely the Aitken dynamic relaxation method, the Broyden rank-one update approach, the Interface-GMRES algorithm and the interface Newton-Krylov method. Their relative performance is compared to Anderson mixing method as well as to the procedure employing heat transfer coefficients.

Numerical experiments are conducted on an industrial scale model of rotating pre-swirl cavity air flow interacting with a turbine disk simulated through an engine flight cycle. The objective of such an analysis is a time-accurate prediction of transient metal temperature distributions. Due to the differences between characteristic scales of the “fast” fluid convection and the “slow” solid conduction processes, the coupled problem is greatly simplified by treating fluid state as quasi-stationary. Hence, fluid flow is simulated by solving steady state temperature dependent compressible Reynolds-averaged Navier-Stokes equations. The solids are modeled by solving the transient heat equation. The test case belongs to a computationally difficult class of problems frequently encountered in industry [25]. The results indicate that Anderson method with reuse of iterations from previous time steps is more efficient than other Jacobian-free techniques.

The rest of the article is organized as follows. Section 2 provides the description of the coupled physical system both in continuous and abstract forms. Jacobian-free techniques for solving the resulting nonlinear fixed-point problem are described in detail in Section 3. Section 4 presents a typical industrial test case with numerical results evaluating relative performance of the discussed methods. Finally, concluding remarks are given in Section 5.

2. PROBLEM STATEMENT

The conjugate heat transfer problem considered in this work constitutes a two field surface coupled multiphysics system. The fluid-structure domain Ω is divided into non-overlapping subdomains Ω_f and Ω_s sharing a common interface $\Sigma = \partial\Omega_f \cap \partial\Omega_s$. We define $\mathbf{n}_f$ as the outward normal to fluid domain Ω_f on Σ , whereas $\mathbf{n}_s = -\mathbf{n}_f$ is its solid domain counterpart.

2.1. Fluid Domain

The flow is modeled assuming ideal heat conducting gas governed by the Reynolds-averaged Navier-Stokes equations

$$\frac{\partial \mathbf{U}}{\partial t} + \nabla \cdot \mathcal{F}(\mathbf{U}, \nabla \mathbf{U}) = 0 \quad \text{in } \Omega_f \times (0, t) \quad (1)$$

where $\mathbf{U} = (\rho, \rho \mathbf{v}^T, \rho E)^T$ is the vector of conservative variables with ρ , $\mathbf{v}$ and E denoting density, velocity, and total internal energy correspondingly. $\mathcal{F}(\mathbf{U}, \nabla \mathbf{U})$ is the total flux vector given by

$$\mathcal{F}(\mathbf{U}, \nabla \mathbf{U}) = \begin{pmatrix} \rho \mathbf{v} \\ \rho \mathbf{v} \otimes \mathbf{v} + p \mathbf{I} \\ (\rho E + p) \mathbf{v} \end{pmatrix} - \begin{pmatrix} 0 \\ \boldsymbol{\tau} \\ \boldsymbol{\tau} \mathbf{v} + q \end{pmatrix} \quad (2)$$

where $\otimes$ denotes the Kronecker product and $\boldsymbol{\tau}$ is the shear stress tensor. The heat flux $\mathbf{q} = -\mathcal{K} \nabla T$ is assumed to obey Fourier’s law, where $\mathcal{K}$ is the thermal conductivity. Ideal gas pressure p and temperature T are linked to the conservative variables through Eqs. 3 and 4 correspondingly.

$$p = (\gamma - 1) \rho \left(E - \frac{\|\mathbf{v}\|_2^2}{2} \right) \quad (3)$$

$$T = \left(E - \frac{\|\mathbf{v}\|_2^2}{2} \right) / \mathcal{H} \quad (4)$$

where γ is the uniform specific heat ratio and $\mathcal{H}$ is specific heat.

2.2. Structural Domain

Heat transfer by conduction in the solid subdomain is modeled with the thermal equilibrium equation assuming Fourier's law

$$\rho\mathcal{C}\frac{\partial T}{\partial t} - \nabla \cdot (\mathcal{K}\nabla T) = 0 \quad \text{in } \Omega_s \times (0, t) \quad (5)$$

where T is temperature, $\mathcal{K}$ is solid thermal conductivity coefficient, ρ is the material density and $\mathcal{C}$ is heat capacity.

2.3. Boundary Conditions

The coupling between the fluid and the solid subdomains is provided by enforcing continuity of the temperature and of the normal component of the heat flux at interface Σ

$$\mathbf{T}_{\Sigma,f} = \mathbf{T}_{\Sigma,s} \quad \text{on } \Sigma \times (0, t) \quad (6)$$

$$\mathbf{n}_f \cdot \mathbf{q}_{\Sigma,f} = -\mathbf{n}_s \cdot \mathbf{q}_{\Sigma,s} \quad \text{on } \Sigma \times (0, t) \quad (7)$$

Both fluid and structure subdomain equations Eqs. 1 and 5 are equipped with appropriate boundary conditions on $\partial\Omega \setminus \Sigma$ and initial conditions on Ω . A classical partitioning scheme known as Dirichlet-Neumann is employed, where the fluid subproblem is supplemented with the Dirichlet boundary condition (temperature), and the Neumann condition (heat flux) is imposed on the solid subdomain.

2.4. Abstract Formulation

The weak formulation of the coupled problem is rather standard. Typically, the fluid field equations are discretized in space using finite volume method, while the solid equations are discretized with finite elements. Explicit and implicit finite difference schemes are utilized in time for fluid and solid subdomains accordingly. To express the coupled problem at a fixed time t in an abstract form we introduce mappings $\mathbb{F}$ and $\mathbb{S}$. Then the fluid problem

$$\mathbb{F}(\mathbf{T}, \mathbf{q}) = 0 \quad (8)$$

represents Eq. 1 with interface boundary conditions Eq. 6. The unknown is the heat flux grouped in the vector variable $\mathbf{q}$. The solid problem can be formally expressed as

$$\mathbb{S}(\mathbf{T}, \mathbf{q}) = 0 \quad (9)$$

symbolizing Eq. 5 with associated boundary conditions Eq. 7 and unknown temperature field $\mathbf{T}$. Thus, with the partitioned strongly coupled approach at any given time level t^{n+1} we can formulate an algebraic system of equations defined on the interface Σ

$$\begin{cases} \mathbf{q}_{\Sigma}^{n+1} = \mathcal{F}(\mathbf{T}_{\Sigma}^{n+1}) \\ \mathbf{T}_{\Sigma}^{n+1} = \mathcal{S}(\mathbf{q}_{\Sigma}^{n+1}) \end{cases} \quad (10)$$

where field operators $\mathcal{F}$ and $\mathcal{S}$ denote fluid and solid solvers. This can be alternatively viewed as a fixed-point problem

$$\mathbf{T}_{\Sigma}^{n+1} = \mathcal{S} \circ \mathcal{F}(\mathbf{T}_{\Sigma}^{n+1}) \quad (11)$$

equivalent to the following nonlinear system

$$\mathcal{G}(\mathbf{T}_{\Sigma}^{n+1}) = 0 \quad (12)$$

where the residual operator $\mathcal{G}$ is defined as

$$\mathcal{G}(\mathbf{T}_{\Sigma}^{n+1}) \equiv \mathcal{S} \circ \mathcal{F}(\mathbf{T}_{\Sigma}^{n+1}) - \mathbf{T}_{\Sigma}^{n+1} \quad (13)$$

3. SOLUTION STRATEGIES

In general, fluid-solid multiphysics problems impose certain restrictions on the algorithmic choices available for solution of the coupled problem Eq. 12. As already mentioned in the introduction the cost of evaluating $\mathcal{G}(\mathbf{T}_\Sigma^n)$ is very high, dominated by the fluid subdomain solver. Therefore a straightforward application of the fixed-point iteration process (repeated steps of a block Gauss-Seidel procedure) is not an option due to its slow convergence. It may result in not only too many calls to the fluid solver but also a failure to converge on many practical thermal coupled problems [23, 18, 26]. Another major difficulty that prevents us from using a standard Newton method, which explicitly requires the Jacobian each iteration, is that an analytic Jacobian is not available and a numerical one is not easily computable. Therefore, we must turn to either quasi-Newton methods, which are modified Newton methods where the actual Jacobian is replaced by an inexpensive approximation to it, or to the vector sequence acceleration methods. Apart from minimizing calls to the fluid solver, consideration should also be given to the amount of intervention required by existing codes. In this regard we favor methods defined on interface implementable as a wraparound of existing black box solvers. Different numerical schemes that meet the above criteria as well as the more established approach based on the estimation of heat transfer coefficient will be outlined in the following.

3.1. Relaxation using estimates of heat transfer coefficient

The coupling method reported in [23] is based on the convective formulation of the heat flux. According to Newton's law of cooling the normal component of heat flux is given by

$$q = h(T_s - T_f) \quad (14)$$

where h has a physical meaning of heat transfer coefficient, T_s and T_f are solid and fluid temperature levels. Assuming that the effect of changes in wall temperature T_s on heat transfer coefficient h due to fluid property variations is insignificant, (with an extra call to the fluid flow solver subject to perturbed wall temperature boundary condition $T_{s2} = T_{s1} + \delta$) one can obtain a reasonable estimate of heat transfer coefficient as

$$h = \frac{q_1 - q_2}{T_{s1} - T_{s2}} \quad (15)$$

Note that h can simply be viewed as a finite difference approximation of $\partial q / \partial T_s$. The derivative thus obtained is consequently employed to solve the solid problem in an implicit way

$$\mathbb{S}(\mathbf{T}_{k+1}^{n+1}, \mathbf{q}_{k+1}^{n+1}) = 0 \quad (16)$$

where subscript $k + 1$ denotes a subiteration. This can be illustrated using a standard expression for the solid domain nonlinear system of equations. After FE discretization and the application of the generalized trapezoidal finite difference rule for time integration, the solid subdomain nonlinear residual is given by

$$\frac{\mathbf{M}}{\Delta t}(\mathbf{T}^n - \mathbf{T}^{n+1}) - \mathbf{K}((1 - \theta)\mathbf{T}^n + \theta\mathbf{T}^{n+1}) + \mathbf{Q}^{n+1} = 0 \quad (17)$$

where $\mathbf{Q}^{n+1} = \mathcal{F}(\mathbf{T}^{n+1})$ denotes the vector of heat fluxes in the nodal space of solid solver. $\mathbf{K}$ and $\mathbf{M}$ are thermal conductivity and capacitance matrices (assuming no temperature dependent material properties) and θ is the time discretization control parameter. This nonlinear equation is then solved iteratively utilizing a quasi-Newton approach, where the Jacobian $\partial \mathbb{S} / \partial \mathbf{T}^{n+1}$ is evaluated only once at the beginning of a time step t^{n+1} as

$$\hat{\mathbf{J}}^{n+1} = \frac{\partial \mathbf{Q}^{n+1}}{\partial \mathbf{T}^{n+1}} - \frac{\mathbf{M}}{\Delta t} - \theta \mathbf{K} \quad (18)$$

with $\partial \mathbf{Q}^{n+1} / \partial \mathbf{T}^{n+1}$ approximated as the surface conductivity matrix formed using the estimates of heat transfer coefficient h . Note, that unlike in other strongly coupled partitioned procedures defined

on interface, the iterations here are carried out on the solid subdomain nonlinear residual by treating the fluid state as an internal variable of the solid subproblem.

Adopting the notation where $\mathbf{f}$ denotes the solid problem nonlinear residual, $\mathbf{x}$ and $\mathbf{y}$ stand for temperature and heat flux vector variables, δ abbreviates a selected wall temperature perturbation factor, the superscript $n + 1$ marks current time step and the subscript k indicates inner iteration count, the solution strategy is summarized in Alg. 1.

Algorithm 1 Quasi-Newton method using estimates of heat transfer coefficient

```

1: Given  $\mathbf{x}_0^{n+1}$ 
2:  $\mathbf{y}_0^{n+1} = \mathcal{F}(\mathbf{x}_0^{n+1})$ ,  $\mathbf{y}_\delta^{n+1} = \mathcal{F}(\mathbf{x}_0^{n+1} + \delta)$ 
3:  $\mathbf{h}^{n+1} = (\mathbf{y}_\delta^{n+1} - \mathbf{y}_0^{n+1})/\delta$ 
4: Form  $\hat{\mathbf{J}}^{n+1}$  according to Eq. 18
5: Evaluate solid residual  $\mathbf{f}_0^{n+1} = \mathbb{S}(\mathbf{x}_0^{n+1}, \mathbf{y}_0^{n+1})$  according to Eq. 17
6:  $k = 0$ 
7: while  $\|\mathbf{f}_k^{n+1}\|_2 > \varepsilon_1$  do
8:   Solve  $\hat{\mathbf{J}}^{n+1} \Delta \mathbf{x}_k^{n+1} = -\mathbf{f}_k^{n+1}$ 
9:    $\mathbf{x}_{k+1}^{n+1} = \mathbf{x}_k^{n+1} + \Delta \mathbf{x}_k^{n+1}$ 
10:   $\mathbf{y}_{k+1}^{n+1} = \mathcal{F}(\mathbf{x}_{k+1}^{n+1})$ 
11:  Evaluate solid residual  $\mathbf{f}_{k+1}^{n+1} = \mathbb{S}(\mathbf{x}_{k+1}^{n+1}, \mathbf{y}_{k+1}^{n+1})$  according to Eq. 17
12:   $k = k + 1$ 
13: end while

```

3.2. Aitken dynamic relaxation

As we have already mentioned, direct application of the fixed-point procedure to thermal coupled problems may lead to oscillations and to a divergent iterative process. Therefore relaxation of each iteration is necessary to increase the stability of the numerical scheme. A simple relaxation strategy is given by

$$\mathbf{x}_k^{n+1} = \mathbf{x}_{k-1}^{n+1} + \beta \mathbf{f}_k^{n+1} \quad (19)$$

The notation follows the general iterative procedure defined in Section 2. At a time step $n + 1$ the nonlinear operator $\mathcal{S} \circ \mathcal{F}$ of Eq. 11 is applied to an input vector $\mathbf{x}_k^{n+1}$ generating an output vector $\tilde{\mathbf{x}}_k^{n+1}$ of the same size. The residual vector is defined as $\mathbf{f}_k^{n+1} = \tilde{\mathbf{x}}_k^{n+1} - \mathbf{x}_k^{n+1}$. Relaxation factor $\beta = 0$ means that the old vector is used for the next iteration, $\beta = 1$ corresponds to the case where new output vector is taken for the next iteration. The latter choice represents the fixed-point procedure. Typically β is set to a constant $0 < \beta < 1$.

Alternatively, an optimal value of the relaxation factor can be selected in each iteration. Different dynamic relaxation techniques have been reported in the FSI body of literature [27, 9, 12]. In this study we use the variant of vector Aitken Δ^2 method formulated in [27]. This algorithm starts with a known sequence of two input/output pairs of vectors and calculates an optimum relaxation parameter as

$$\beta_k = \beta_{k-1} + \beta_{k-1} \frac{\Delta \mathbf{f}_k^{n+1 T} \mathbf{f}_k^{n+1}}{\|\Delta \mathbf{f}_k^{n+1}\|_2^2} \quad (20)$$

with the residual difference vector $\Delta \mathbf{f}_k^{n+1} = \mathbf{f}_{k-1}^{n+1} - \mathbf{f}_k^{n+1}$. The first iteration is carried out using a pre-selected relaxation constant β_0 . This method is quite straightforward to implement, needs minimum processor and storage resources, while taking a single residual evaluation per iteration. The entire procedure is outlined in Alg. 2.

3.3. Broyden rank-one update

Broyden's method is a member of the quasi-Newton family of algorithms. Instead of computing the exact Jacobian, it employs an approximation thereof, which typically does not involve calculating

Algorithm 2 Aitken dynamic relaxation method

```

1: Given  $\mathbf{x}_0^{n+1}$  and  $\beta_0$ 
2:  $\tilde{\mathbf{x}}_0^{n+1} = \mathcal{S} \circ \mathcal{F}(\mathbf{x}_0^{n+1})$ 
3:  $\mathbf{f}_0^{n+1} = \tilde{\mathbf{x}}_0^{n+1} - \mathbf{x}_0^{n+1}$ 
4:  $\mathbf{x}_1^{n+1} = \mathbf{x}_0^{n+1} + \beta_0 \mathbf{f}_0^{n+1}$ 
5:  $k = 0$ 
6: while  $\|\mathbf{f}_k^{n+1}\|_2 > \varepsilon_1$  do
7:    $k = k + 1$ 
8:    $\tilde{\mathbf{x}}_k^{n+1} = \mathcal{S} \circ \mathcal{F}(\mathbf{x}_k^{n+1})$ 
9:    $\mathbf{f}_k^{n+1} = \tilde{\mathbf{x}}_k^{n+1} - \mathbf{x}_k^{n+1}$ 
10:   $\Delta \mathbf{f}_k^{n+1} = \mathbf{f}_k^{n+1} - \mathbf{f}_{k-1}^{n+1}$ 
11:   $\beta_k = \beta_{k-1} + \beta_{k-1} \frac{\Delta \mathbf{f}_k^{n+1 T} \mathbf{f}_k^{n+1}}{\|\Delta \mathbf{f}_k^{n+1}\|_2^2}$ 
12:   $\mathbf{x}_{k+1}^{n+1} = \mathbf{x}_k^{n+1} + \beta_k \mathbf{f}_k^{n+1}$ 
13: end while

```

the derivatives. The latter is updated each iteration [28]. An iteration of Broyden's second type method is described by

$$\mathbf{x}_k^{n+1} = \mathbf{x}_{k-1}^{n+1} + \mathbf{B}_k^{-1} \mathbf{f}_k^{n+1} \quad (21)$$

where $\mathbf{B}_k^{-1}$ is intended to approximate the inverse of the exact Jacobian of $\mathbf{f}^{n+1}$ at iteration k . The update formula for $\mathbf{B}_k^{-1}$ that satisfy both the secant condition

$$\mathbf{B}_k^{-1} \Delta \mathbf{f}_k^{n+1} = \Delta \mathbf{x}_k^{n+1} \quad (22)$$

and the minimal modification to $\mathbf{B}_k^{-1}$ condition

$$\mathbf{B}_{k+1}^{-1} \mathbf{q} = \mathbf{B}_k^{-1} \mathbf{q} \quad (23)$$

for all $\mathbf{q}$ orthogonal to $\Delta \mathbf{f}_k^{n+1}$ is given by

$$\mathbf{B}_{k+1}^{-1} = \mathbf{B}_k^{-1} + \frac{(\Delta \mathbf{x}_k^{n+1} + \mathbf{B}_k^{-1} \Delta \mathbf{f}_k^{n+1}) \Delta \mathbf{f}_k^{n+1 T}}{\Delta \mathbf{f}_k^{n+1 T} \Delta \mathbf{f}_k^{n+1}} \quad (24)$$

with $\Delta \mathbf{x}_k^{n+1} = \mathbf{x}_{k-1}^{n+1} - \mathbf{x}_k^{n+1}$ and $\Delta \mathbf{f}_k^{n+1} = \mathbf{f}_{k-1}^{n+1} - \mathbf{f}_k^{n+1}$. In the absence of any information on the exact Jacobian, the first iteration initial guess is $\mathbf{B}_0^{-1} = \beta I$.

The Broyden update in this form has the disadvantage of storing the full matrix inverse of the approximate Jacobian. This problem can be circumvented by not forming the matrix explicitly, since only its action on a vector is needed. Instead, each iteration we store the factors of dyadic products as individual vectors [16]

$$\mathbf{B}_{k+1}^{-1} = (I + \mathbf{u}_k \mathbf{v}_k^T)(I + \mathbf{u}_{k-1} \mathbf{v}_{k-1}^T) \dots (I + \mathbf{u}_0 \mathbf{v}_0^T) \mathbf{B}_0^{-1} \quad (25)$$

where $\mathbf{u}_k = (\Delta \mathbf{x}_k^{n+1} + \mathbf{B}_k^{-1} \Delta \mathbf{f}_k^{n+1}) / \|\Delta \mathbf{f}_k^{n+1}\|_2$ and $\mathbf{v}_k = \Delta \mathbf{f}_k^{n+1} / \|\Delta \mathbf{f}_k^{n+1}\|_2$.

The advantage of Broyden class of methods is that they only require the residuals, not the derivatives, taking a single residual evaluation per iteration. The entire procedure is given in Alg. 3.

3.4. Interface-GMRES

Interface-GMRES method was first introduced in [11], and its numerical performance on FSI problems and links to vector extrapolation algorithms are discussed in [12, 13]. This approach is a nested iteration Newton method. The primary level is iterating over a sequence of linear Newton correction problems

$$\mathbf{x}_k^{n+1} = \mathbf{x}_{k-1}^{n+1} + \mathbf{J}_k^{-1} \mathbf{f}_k^{n+1} \quad (26)$$

Algorithm 3 Broyden rank-one update method of second type

```

1: Given  $\mathbf{x}_0^{n+1}$  and  $\beta$ 
2:  $\tilde{\mathbf{x}}_0^{n+1} = \mathcal{S} \circ \mathcal{F}(\mathbf{x}_0^{n+1})$ 
3:  $\mathbf{f}_0^{n+1} = \tilde{\mathbf{x}}_0^{n+1} - \mathbf{x}_0^{n+1}$ 
4:  $\Delta \mathbf{x}_1^{n+1} = \mathbf{f}_0^{n+1}$ 
5:  $\mathbf{x}_1^{n+1} = \mathbf{x}_0^{n+1} + \beta \Delta \mathbf{x}_1^{n+1}$ 
6:  $k = 0$ 
7: while  $\|\mathbf{f}_k^{n+1}\|_2 > \varepsilon_1$  do
8:    $k = k + 1$ 
9:    $\tilde{\mathbf{x}}_k^{n+1} = \mathcal{S} \circ \mathcal{F}(\mathbf{x}_k^{n+1})$ 
10:   $\mathbf{f}_k^{n+1} = \tilde{\mathbf{x}}_k^{n+1} - \mathbf{x}_k^{n+1}$ 
11:   $\Delta \mathbf{f}_k^{n+1} = \mathbf{f}_k^{n+1} - \mathbf{f}_{k-1}^{n+1}$ 
12:   $\mathbf{v}_k = \frac{\Delta \mathbf{f}_k^{n+1}}{\|\Delta \mathbf{f}_k^{n+1}\|_2}$ 
13:   $\mathbf{u}_k = \beta \Delta \mathbf{f}_k^{n+1}$ 
14:   $\Delta \mathbf{x}_{k+1}^{n+1} = \beta \mathbf{f}_k^{n+1}$ 
15:  for  $i = 1, 2, \dots, k-1$  do
16:     $\mathbf{u}_k = \mathbf{u}_k + \mathbf{u}_i \mathbf{v}_i^T \mathbf{u}_k$ 
17:     $\Delta \mathbf{x}_{k+1}^{n+1} = \Delta \mathbf{x}_{k+1}^{n+1} + \mathbf{u}_i \mathbf{v}_i^T \Delta \mathbf{x}_{k+1}^{n+1}$ 
18:  end for
19:   $\mathbf{u}_k = \frac{\Delta \mathbf{x}_{k+1}^{n+1} + \mathbf{u}_k}{\|\Delta \mathbf{f}_k^{n+1}\|_2}$ 
20:   $\Delta \mathbf{x}_{k+1}^{n+1} = \Delta \mathbf{x}_{k+1}^{n+1} + \mathbf{u}_k \mathbf{v}_k^T \Delta \mathbf{x}_{k+1}^{n+1}$ 
21:   $\mathbf{x}_{k+1}^{n+1} = \mathbf{x}_k^{n+1} + \Delta \mathbf{x}_{k+1}^{n+1}$ 
22: end while

```

where $\mathbf{f}_k^{n+1} = \tilde{\mathbf{x}}_k^{n+1} - \mathbf{x}_k^{n+1}$ is the nonlinear residual vector, $\mathbf{x}_k^{n+1}$ and $\tilde{\mathbf{x}}_k^{n+1}$ are input and output vectors of the nonlinear operator $\mathcal{S} \circ \mathcal{F}$, $\mathbf{J}_k$ denotes the Jacobian matrix of $\mathbf{f}^{n+1}$.

Each linear subproblem $\mathbf{J}_k^{-1} \mathbf{f}_k^{n+1}$ in turn is solved using a matrix-free GMRES method without explicitly forming the Jacobian. It is argued that instead of building a Krylov subspace $\mathbf{K}_j = \text{span}\{\mathbf{f}_k^{n+1}, \mathbf{J}_k \mathbf{f}_k^{n+1}, \mathbf{J}_k^2 \mathbf{f}_k^{n+1}, \dots, \mathbf{J}_k^j \mathbf{f}_k^{n+1}\}$ spanned by true Jacobian matrix-vector products, a solution to the linear Newton correction problem can be extracted from the subspace $\mathbf{K}_j = \text{span}\{\tilde{\mathbf{x}}_1^{n+1} - \mathbf{x}_k^{n+1}, \dots, \tilde{\mathbf{x}}_j^{n+1} - \mathbf{x}_k^{n+1}\}$ spanned by a sequence of differences between vectors generated by the fixed-point iteration process on $\mathbf{x}_k^{n+1}$ with itself. An important feature of this algorithm is the possibility to reuse Krylov vectors from previous Newton iterations. The version with data reuse is given in Alg. 4. Note that with a simple modifications to lines 13 and 16 of the algorithm, one can incorporate data generated from j antecedent time steps into $\Delta \mathbf{X}_k^{n+1} = (\Delta \tilde{\mathbf{x}}_{1,1}^{n+1-j}, \dots, \Delta \tilde{\mathbf{x}}_{k,i}^{n+1})$ and $\Delta \mathbf{F}_k^{n+1} = (\Delta \tilde{\mathbf{f}}_{1,1}^{n+1-j}, \dots, \Delta \tilde{\mathbf{f}}_{k,i}^{n+1})$ correspondingly.

3.5. Newton-Krylov

The Newton-Krylov method is a nested iteration Newton approach, where each Newton step is found with a matrix-free Krylov subspace solver [28, 15]. The primary iteration loop is carried out solving a sequence of linear Newton step subproblems

$$\mathbf{x}_k^{n+1} = \mathbf{x}_{k-1}^{n+1} + \mathbf{J}_k^{-1} \mathbf{f}_k^{n+1} \quad (27)$$

Each of the subproblems is solved using GMRES method that only requires the action of the Jacobian in a form of a matrix-vector product, which is approximated as

$$\mathbf{J}_k \mathbf{v} = \frac{(\mathcal{G}(\mathbf{x}_k^{n+1} + \delta \mathbf{v}) - \mathcal{G}(\mathbf{x}_k^{n+1}))}{\delta} \quad (28)$$

for a small perturbation δ . The Newton-Krylov method applied to our interface problem is shown in Alg. 5. In our study we adopt the inexact version of the algorithm, the forcing term η_k is changing

Algorithm 4 Interface-GMRES method

```

1: Given  $\mathbf{x}_1^{n+1}$ 
2:  $\tilde{\mathbf{x}}_1^{n+1} = \mathcal{S} \circ \mathcal{F}(\mathbf{x}_1^{n+1})$ 
3:  $\mathbf{f}_1^{n+1} = \tilde{\mathbf{x}}_0^{n+1} - \mathbf{x}_1^{n+1}$ 
4:  $k = 0, i = 0$ 
5: while  $\|\mathbf{f}_k^{n+1}\|_2 > \varepsilon_1$  do
6:    $k = k + 1$ 
7:    $\zeta = \|\mathbf{f}_k^{n+1}\|_2$ 
8:    $\tilde{\mathbf{x}}_{k,i} = \tilde{\mathbf{x}}_k^{n+1}$ 
9:   while  $\zeta > \varepsilon_2$  do
10:     $i = i + 1$ 
11:     $\Delta\tilde{\mathbf{x}}_{k,i}^{n+1} = \tilde{\mathbf{x}}_{k,i-1}^{n+1} - \mathbf{x}_k^{n+1}$ 
12:     $\tilde{\mathbf{x}}_{k,i}^{n+1} = \mathcal{S} \circ \mathcal{F}(\tilde{\mathbf{x}}_{k,i-1}^{n+1})$ 
13:     $\Delta\mathbf{X}_k^{n+1} = (\Delta\tilde{\mathbf{x}}_{1,1}^{n+1}, \dots, \Delta\tilde{\mathbf{x}}_{k,i}^{n+1})$ 
14:     $\mathbf{f}_{k,i}^{n+1} = \tilde{\mathbf{x}}_{k,i} - \tilde{\mathbf{x}}_{k,i-1}$ 
15:     $\Delta\mathbf{f}_{k,i} = \mathbf{f}_{k,i}^{n+1} - \mathbf{f}_k^n$ 
16:     $\Delta\mathbf{F}_k^{n+1} = (\Delta\mathbf{f}_{1,1}^{n+1}, \dots, \Delta\mathbf{f}_{k,i}^{n+1})$ 
17:    Solve  $\arg\min_{\gamma} \|\Delta\mathbf{F}_k^{n+1}\gamma + \mathbf{f}_k^{n+1}\|_2$ 
18:     $\zeta = \|\Delta\mathbf{F}_k^{n+1}\gamma + \mathbf{f}_k^{n+1}\|_2$ 
19:  end while
20:  $\mathbf{x}_{k+1}^{n+1} = \mathbf{x}_k^{n+1} + \Delta\mathbf{X}_k^{n+1}\gamma$ 
21:  $\tilde{\mathbf{x}}_{k+1}^{n+1} = \mathcal{S} \circ \mathcal{F}(\mathbf{x}_{k+1}^{n+1})$ 
22:  $\mathbf{f}_{k+1}^{n+1} = \tilde{\mathbf{x}}_{k+1}^{n+1} - \mathbf{x}_{k+1}^{n+1}$ 
23: end while

```

according to the policy given in [29]

$$\eta_k = \chi \left(\frac{\|\mathbf{f}_k^{n+1}\|_2}{\|\mathbf{f}_{k-1}^{n+1}\|_2} \right)^\psi \quad (29)$$

for selected parameters $\chi \in (0, 1]$ and $\psi \in (1, 2]$.

3.6. Anderson mixing

Anderson mixing method was originally developed by Anderson [30] to solve certain nonlinear integral equations with applications in physics. It follows a general iterative procedure defined by a nonlinear operator, which if applied to an input vector generates an output vector. Input and output vectors generated during a number of previous iterations are “mixed” or combined in an optimal way to produce a more accurate prediction of the next iterate. The procedure appears to be a valuable general tool for accelerating fixed-point iterations that can be viewed as a member of multiseant quasi-Newton class of methods [21, 22]. From that perspective, the Anderson mixing method calculates a Newton step as

$$\mathbf{x}_k^{n+1} = \mathbf{x}_{k-1}^{n+1} + \mathbf{B}_k^{-1} \mathbf{f}_k^{n+1} \quad (30)$$

where $\mathbf{B}_k^{-1}$ is an approximation of the true Jacobian inverse at iteration k . The generalized Broyden method of second type with flexible rank m update imposes on $\mathbf{B}_k^{-1}$ two conditions. First, it must satisfy a set of m secant equations

$$\mathbf{B}_k^{-1} \Delta\mathbf{F}_k^{n+1} = \Delta\mathbf{X}_k^{n+1} \quad (31)$$

Algorithm 5 Newton-Krylov method

```

1: Given  $\mathbf{x}_1^{n+1}$ 
2:  $\tilde{\mathbf{x}}_1^{n+1} = \mathcal{S} \circ \mathcal{F}(\mathbf{x}_1^{n+1})$ 
3:  $\mathbf{f}_1^{n+1} = \tilde{\mathbf{x}}_1^{n+1} - \mathbf{x}_1^{n+1}$ 
4:  $k = 0$ 
5: while  $\|\mathbf{f}_k^{n+1}\|_2 > \varepsilon$  do
6:    $k = k + 1$ 
7:    $\mathbf{v}_0 = \frac{\mathbf{f}_k^{n+1}}{\|\mathbf{f}_k^{n+1}\|_2}$ 
8:    $\zeta = \|\mathbf{f}_k^{n+1}\|_2, \alpha = \zeta, i = 0$ 
9:    $\eta_k = \chi(\|\mathbf{f}_k^{n+1}\|_2 / \|\mathbf{f}_{k-1}^{n+1}\|_2)^\psi$ 
10:  while  $\zeta > \eta_k \|\mathbf{f}_k^{n+1}\|_2$  and  $i \leq i_{max}$  do
11:     $i = i + 1$ 
12:     $\tilde{\mathbf{x}}_i = \mathbf{x}_k^{n+1} + \delta \mathbf{v}_{i-1}$ 
13:     $\mathbf{f}_i = \tilde{\mathbf{x}}_i - \mathcal{S} \circ \mathcal{F}(\tilde{\mathbf{x}}_i)$ 
14:     $\mathbf{v}_i = \frac{\mathbf{f}_i - \mathbf{f}_k^{n+1}}{\delta}$ 
15:    for  $j = 1, \dots, i - 1$  do
16:       $h_{j,i-1} = \mathbf{v}_j^T \mathbf{v}_{i-1}$ 
17:       $\mathbf{v}_i = \mathbf{v}_i - h_{j,i-1} \mathbf{v}_j$ 
18:    end for
19:     $h_{i,i-1} = \|\mathbf{v}_i\|_2$ 
20:     $\mathbf{h}_{i-1} = (h_{0,i-1}, \dots, h_{i,i-1})^T$ 
21:     $\mathbf{H} = (\mathbf{h}_0, \dots, \mathbf{h}_{i-1})$ 
22:     $\mathbf{v}_i = \frac{\mathbf{v}_i}{\|\mathbf{v}_i\|_2}$ 
23:     $\mathbf{V} = (\mathbf{v}_0, \dots, \mathbf{v}_i)$ 
24:     $\mathbf{e}_1 = (1, 0, \dots, 0)^T$ 
25:    Solve  $\argmin_{\gamma} \|\alpha \mathbf{e}_1 - \mathbf{H} \gamma\|_2$ 
26:     $\zeta = \|\alpha \mathbf{e}_1 - \mathbf{H} \gamma\|_2$ 
27:  end while
28:   $\mathbf{x}_{k+1}^{n+1} = \mathbf{x}_k^{n+1} + \mathbf{V} \gamma$ 
29:   $\tilde{\mathbf{x}}_{k+1}^{n+1} = \mathcal{S} \circ \mathcal{F}(\mathbf{x}_{k+1}^{n+1})$ 
30:   $\mathbf{f}_{k+1}^{n+1} = \tilde{\mathbf{x}}_{k+1}^{n+1} - \mathbf{x}_{k+1}^{n+1}$ 
31: end while

```

where $\Delta \mathbf{F}_k^{n+1} = (\mathbf{f}_{k-m}^{n+1} - \mathbf{f}_{k-m+1}^{n+1}, \dots, \mathbf{f}_{k-1}^{n+1} - \mathbf{f}_k^{n+1})$ and $\Delta \mathbf{X}_k^{n+1} = (\mathbf{x}_{k-m}^{n+1} - \mathbf{x}_{k-m+1}^{n+1}, \dots, \mathbf{x}_{k-1}^{n+1} - \mathbf{x}_k^{n+1})$. Second, it should comply with the no-change condition

$$(\mathbf{B}_k^{-1} - \mathbf{B}_{k-m}^{-1})\mathbf{q} = 0 \quad (32)$$

for all vectors $\mathbf{q}$ orthogonal to the subspace spanned by the columns of $\Delta \mathbf{F}_k^{n+1}$. An update formula that satisfy both conditions takes the form

$$\mathbf{B}_{k+1}^{-1} = \mathbf{B}_{k-m}^{-1} + (\Delta \mathbf{X}_k^{n+1} + \mathbf{B}_{k-m}^{-1} \Delta \mathbf{F}_k^{n+1})(\Delta \mathbf{F}_k^{n+1 T} \Delta \mathbf{F}_k^{n+1})^{-1} \Delta \mathbf{F}_k^{n+1 T} \quad (33)$$

Selecting $\mathbf{B}_{k-m}^{-1} = \beta I$ corresponds to a classical Anderson acceleration method with a mixing parameter β . The procedure is summarized in Alg. 6. Note that unlike the dynamic relaxation and Broyden rank-one update methods specified earlier, the Anderson approach finds a next Newton direction as an optimal linear combination of input and output vectors within a subspace spanned by m previous iterations. On the other hand, in contrast to Interface-GMRES or Newton-Krylov methods it makes no use of inner iterations requiring a single residual evaluation per Newton iteration. To reuse data from j foregoing time steps lines 10 and 11 have to be modified as $\Delta \mathbf{X}_k^{n+1} = (\mathbf{x}_0^{n+1-j} - \mathbf{x}_1^{n+1-j}, \dots, \mathbf{x}_{k-1}^{n+1} - \mathbf{x}_k^{n+1})$ and $\Delta \mathbf{F}_k^{n+1} = (\mathbf{f}_0^{n+1-j} - \mathbf{f}_1^{n+1-j}, \dots, \mathbf{f}_{k-1}^{n+1} - \mathbf{f}_k^{n+1})$ correspondingly.

Algorithm 6 Anderson mixing method

```

1: Given  $\mathbf{x}_0^{n+1}$  and  $\beta$ 
2:  $\tilde{\mathbf{x}}_0^{n+1} = \mathcal{S} \circ \mathcal{F}(\mathbf{x}_0^{n+1})$ 
3:  $\mathbf{f}_0^{n+1} = \tilde{\mathbf{x}}_0^{n+1} - \mathbf{x}_0^{n+1}$ 
4:  $\mathbf{x}_1^{n+1} = \mathbf{x}_0^{n+1} + \beta \mathbf{f}_0^{n+1}$ 
5:  $k = 0$ 
6: while  $\|\mathbf{f}_k^{n+1}\|_2 > \varepsilon_1$  do
7:    $k = k + 1$ 
8:    $\tilde{\mathbf{x}}_k^{n+1} = \mathcal{S} \circ \mathcal{F}(\mathbf{x}_k^{n+1})$ 
9:    $\mathbf{f}_k^{n+1} = \tilde{\mathbf{x}}_k^{n+1} - \mathbf{x}_k^{n+1}$ 
10:   $\Delta \mathbf{X}_k^{n+1} = (\mathbf{x}_0^{n+1} - \mathbf{x}_1^{n+1}, \dots, \mathbf{x}_{k-1}^{n+1} - \mathbf{x}_k^{n+1})$ 
11:   $\Delta \mathbf{F}_k^{n+1} = (\mathbf{f}_0^{n+1} - \mathbf{f}_1^{n+1}, \dots, \mathbf{f}_{k-1}^{n+1} - \mathbf{f}_k^{n+1})$ 
12:  Solve  $\arg\min_{\gamma} \|\Delta \mathbf{F}_k^{n+1} \gamma + \mathbf{f}_k^{n+1}\|_2$ 
13:   $\mathbf{x}_{k+1}^{n+1} = \mathbf{x}_k^{n+1} + \beta \mathbf{f}_k^{n+1} + (\Delta \mathbf{X}_k^{n+1} + \beta \Delta \mathbf{F}_k^{n+1}) \gamma$ 
14: end while

```

4. NUMERICAL EXPERIMENTS

In this section relative performance of the algorithms described in the foregoing is compared on a test problem of a rotating cavity cooling flow interacting with a gas turbine disk component. This example is thought to be realistic enough to exhibit the characteristics of practical industrial scale transient heat transfer problems. Convergence behavior, total iteration counts and the effect of previously generated data reuse are investigated.

4.1. Simulation setup

The structural subdomain is a part of a larger industrial model of a jet engine representing a turbine disk component. It is discretized with 2D triangular axisymmetric 6-node finite elements as depicted in Fig. 1(b), which results in a system of 3459 degrees of freedom. Neumann boundary conditions are imposed at the interior surface of the cavity marked by an arrow. For simplicity, the remaining boundaries are assumed to be insulated. Implicit time integration scheme is employed throughout simulation, where the time stepping sequence is generated automatically by controlling a quadratic variation of temperature through three most recent solution points. The structure subdomain problem is discretized and solved in Matlab using FAESOR toolbox [31].

The geometry of the fluid subdomain is shown in Fig. 1(a). It is modeled in a form of 3D axisymmetric 0.4° periodic sector discretized with block-structured mesh. The cavity filled with cooling air is bounded by a rotating disk and a stator. Flow enters through two inlets, side leakage and inner labyrinth seal, marked in Fig. 2 by 2 and 3 correspondingly. It leaves the cavity through the outer labyrinth seal marked by 1. The fluid subdomain is solved using the $\kappa - \epsilon$ turbulence model with wall functions. The Rolls-Royce in-house CFD code HYDRA [32] with finite-volume discretization, edge-based data formulation and explicit time-marching five step Runge-Kutta technique with multigrid smoothing is used throughout the simulation.

Following the standard industrial practice, thermal transient analysis is carried out with the steady-state fluid subdomain solutions. This is usually justified by the fact that many multi-scale problems in time contain physical variables that represent processes operating at different speeds, which is the case with a relatively slow response of metal heat conduction compared to much faster flow convection. Consequently, such systems admit the standard quasi-steady-state approximation. As a result of neglecting fluid transient effects the computational effort is considerably reduced.

A transient cycle is defined in time by key points that represent change in operating conditions. The environment parameters such as rotational speed, inlet mass flow rate and cooling flow temperature as well as outlet pressure are defined at those ramp points. Following the standard procedure detailed in [24], major differences in the inlet mass flow rates and outlet pressures at low

Figure 1. Fluid (a) and solid (b) subdomain grids. The location of fluid-structure coupled interface is marked by an arrow.

Figure 2. Flow stream function (a) and location of the inlets and the outlet relative to the solid subdomain (b).

and maximum power operating conditions are reflected by creating two distinct grids for the fluid subdomain featuring 44,735 and 83,760 nodes. Two grids are developed in an attempt to keep the size of near-wall cells within the range recommended for standard wall functions while maintaining a proper level of mesh quality and resolution. Adiabatic flow solutions are pre-computed off-line for each grid. First fluid-solid cycle of an operating condition starts by running fluid solver with the

Table I. Transient cycle flow parameters.

		Low Power	Maximum Power
Outlet 1	Rotation speed, Ω/Ω_r	0.6270	1.0
	Pressure, P/P_r	0.1021	1.0
Inlet 2	Mass flow rate, $\dot{m}/\dot{m}_r$	0.0032	0.0268
	Swirl ratio	0	0
Inlet 3	Flow temperature, T/T_r	0.3285	1.0
	Mass flow rate, $\dot{m}/\dot{m}_r$	0.2382	1.0
	Swirl ratio	0.2931	0.3060
	Flow temperature, T/T_r	0.3285	1.0

initial flow solution prepared off-line, the following cycles take it from the previous sub-iterations. During the short transition zones between low and high power points two heat flux predictions are obtained using both CFD models with the boundary temperature supplied by solid solver. The heat flux conveyed back to solid subdomain is approximated as a linear interpolation in time between those two predictions. Fig. 3 shows the transient cycle in terms of the normalized rotational speed, cooling flow temperature and associated CFD models. Tab. I provides further details on flow

Figure 3. Transient cycle definition showing evolution of relative rotational speed, relative cooling flow temperature and associated fluid domain models versus operating condition points in time.

parameters at two operating conditions of the transient cycle.

4.2. Results and discussion

In this section, we carry out some numerical tests comparing the performances of the Anderson algorithm with respect to Broyden, Newton-Krylov, Interface-GMRES, Aitken and the quasi-Newton schemes. During the experiments each calculation the nonlinear interface residuals are measured in l^2 norm scaled by $1/\sqrt{N}$, where N is the number of interface variables. A nonlinear interface iteration is accepted whenever the residual norm becomes smaller than a combination of the relative and the absolute errors. The former tolerance is set to 10^{-2} and the latter is set to 10, which matches a typical industrial accuracy level. The accuracy is selected to provide a smooth enough coupled solution. Since the thermal code controls local time integration error, a non-smooth solution may trigger unnecessary time step reduction thus lengthening the simulation. The time stepping sequences are generated automatically at each trial and do not necessarily match.

Failure to converge within the maximum number of iterations, set to 20, causes automatic time step adjustment. The first iteration relaxation parameter is set to $\beta = 0.4$ for the Aitken, Broyden and Anderson methods. Inner residual tolerance ε_2 is set to $0.5\|\zeta_0\|_2$ for the Interface-GMRES method, whereas for the Newton-Krylov method it varies according to the forcing term strategy given in [29] with parameters $\chi = 0.9$ and $\psi = 2$.

Solid subdomain temperature contours obtained at ramp points 3, 5 and 7 of the transient cycle are shown in Fig. 4. We can see that during the transient period between low and high power operating conditions the structure temperature grows locally in the areas surrounding the cavity, then the heat energy is gradually dissipated towards the end of the cycle. The metal temperature

Figure 4. Metal temperature field at ramp points 3 (a), 5 (b) and 7 (c).

profiles for these three ramp points taken from the turbine disk wall shown in Fig. 5 and predicted by different methods are compared in Fig. 6. We note that the methods provide nearly identical (within the selected tolerance level) boundary temperature predictions, consistent with the physics of the problem. If the coupled solution appears not to be affected by the algorithmic choices, there are some physical constraints that may influence the performance of these methods. First, there is a common restriction on the iterates $\mathbf{x}_k^{n+1}$. Having a physical meaning of temperature measured in $^{\circ}K$ they cannot assume any negative values. If taken too far from the current flow conditions, they may also induce convergence difficulties of the fluid solver. For example, in the Newton-Krylov method evaluation of $\mathcal{F}(\mathbf{x}_k^{n+1} + \delta\mathbf{v})$ while approximating a Jacobian-vector product may have no physical meaning for some $\delta\mathbf{v}$ or may not be computable at all. Again, the initial flow solution that we reuse between time steps and subiterations may be corrupted by the inner iterations of the nested methods. Therefore, different flow solutions have been used during inner and outer iterations in the Newton-Krylov and the Interface-GMRES methods.

Figure 5. Boundary profile location on the turbine disk.

Figure 6. Comparison of boundary temperature profiles obtained using different methods at ramp point 3 (a)-(c), 5 (d)-(f) and 7 (g)-(i).

Figs. 7 and 8 show relative residual convergence rates for two time steps. One is taken from a relatively high temperature gradient area between ramp points 3 and 4, while the other from the steady state zone between ramp points 6 and 7. Residual reduction is presented against both the number of subiterations and fluid-solid cycles. The latter refers to the number of calls to the fluid subdomain solver that must be carried out sequentially. We note that the Newton-Krylov

Figure 7. Relative residual reduction for 55,377 s time step versus nonlinear iterations (a) and fluid-solid cycles (b). For Newton-Krylov and Interface-GMRES nested iteration methods corresponding linear residual is indicated by a dashed line.

Figure 8. Relative residual reduction for 57,050 s time step versus nonlinear iterations (a) and fluid-solid cycles (b). For Newton-Krylov and Interface-GMRES nested iteration methods corresponding linear residual is indicated by a dashed line.

and Interface-GMRES nested iteration methods converge monotonically and exhibit the highest convergence rates. A better Jacobian approximation leads to less numerous but more expensive Newton iterations. These methods require significantly more fluid solver calls per Newton iteration, as shown in Figs. 7(b) and 8(b). Their linear residual reduction rates are also displayed for reference. They express a reduction in the norm of local linearized models of $\mathcal{G}$. The overall objective is to solve them inexpensively to just enough precision to make a good progress towards the solution while being far from it, at the same time to obtain a nearly quadratic convergence rate if getting closer. In fact, the two methods are almost equivalent showing similar performance. The advantage of Interface-GMRES over Newton-Krylov method becomes clear in Fig. 7(b), where the Newton-Krylov method oversolves the linear subproblem during second iteration resulting from an inappropriately small forcing term. Typically, performance of the inexact Newton-Krylov schemes is strongly dependent on fine tuning of the control parameters, which is a non-trivial task.

Increased temperature gradient may cause a failure of convergence for certain methods. Thus, Broyden rank-1 method exhibits the slowest rate in the high gradient time step having noticeable difficulties to converge. A mild reduction in number of iterations is observed moving to the low

gradient time step. Aitken and quasi-Newton htc schemes both have comparable non-monotonous convergence behavior in the high zone. On the other hand, the rate of convergence of Aitken method approaches those of the nested iteration methods when temperature gradient goes lower. The results show that Anderson mixing method enjoys the most optimal convergence behavior among the studied coupling approaches. It exhibits comparable to Newton-Krylov and Interface-GMRES convergence rates with monotonous behavior in both time steps at significantly lower cost. Only a slight increase in number of iterations is observed moving towards the high temperature gradient zone.

The results of this experiment displaying the number of coupling iterations needed by each algorithm to satisfy the stopping criterion with selected tolerance at each time step are shown in Fig. 9. Clearly, the convergence difficulties of the Broyden rank-1 method, especially in the high

Figure 9. Comparison of the number of fluid-solid cycles per time step.

gradient area, translate into higher number of smaller size time steps taken and ultimately into lowest efficiency. The performances of the Newton-Krylov and the Interface-GMRES methods are quite similar. In most cases they converge in 6 to 8 iterations per time step. The difference in total computational cost between those methods and the Quasi-Newton htc scheme appears to be not too big. The number of iterations of the latter increases dramatically as we progress into high temperature gradient areas or take larger time steps in the low gradient zones. This can be explained by the fact that the Jacobian approximation is computed only once using poor estimates

of the earliest iteration, while it is not continuously updated taking advantage of the nonlinear convergence. The Anderson method demonstrates similar performance to the Aitken method taking from 4 to 6 iterations per time step in most cases.

In the second set of experiments we investigate the effect of time step data reuse in the Interface-GMRES and the Anderson mixing methods. The number of coupling iterations without reuse as well as with reuse of several consecutive time steps are shown in Fig. 10 for the Interface-GMRES method and in Fig. 11 for the Anderson acceleration approach. The reuse of one time step data in the

Figure 10. The effect of time step data reuse in the Interface-GMRES method on the number of fluid-solid cycles.

Figure 11. The effect of time step data reuse in the Anderson mixing method on the number of fluid-solid cycles.

Interface-GMRES method brings about some minor gain in performance. This is not surprising since the time step reuse significantly slows down convergence in at least 7 time steps, as is clear from Fig. 10. Incorporating data from two consecutive time steps does not appear to yield any benefits. On the other hand, including data from one time step in the Anderson approach provides substantial reduction in the number of iterations per time step, bringing that number down in most cases to 3

Table II. Comparison of the total number of nonlinear iterations, coupled iterations, fluid solver calls and cpu time.

Method	Subiterations	Fluid-Solid Cycles	Total Fluid Solver Calls	CPU time
Broyden rank 1	336	336	436	22h 26m
Newton-Krylov	141	280	381	18h 24m
Interface-GMRES	156	272	347	17h 34m
Interface-GMRES(1)	144	260	333	16h 58m
Quasi-Newton htc	255	255	381	16h 40m
Aitken	202	202	259	13h 11m
Anderson	194	194	247	12h 49m
Anderson(1)	145	145	188	9h 27m

to 4, although a small number of time steps converge more slowly. This results in immediate 25 % boost in efficiency relative to the case with no data reuse. However, “mixing” in the vectors from two or more antecedent time steps causes gradual increase of the computational cost per iteration. This reflects the fact that the vector differences formed earlier than one time step ago in most cases become outdated. Hence, the optimum performance for this problem in the sense of minimizing fluid-solid iterations is achieved when reusing the data from one previous time step.

Finally, the cumulative number of iterations relative to the time step size is reported in Fig. 12. In the figures we also include performance of Interface-GMRES and Anderson methods run with the optimal number of time steps to reuse. Tab. II reports performance of each method during the trials

Figure 12. Cumulative number of fluid-solid cycles as a function of time.

giving a total number of nonlinear iterations, a total number of fluid-solid cycles, a total number of fluid solver invocations and cpu time. The number of cfd code invocations includes extra fluid solver calls needed to estimate heat transfer coefficient for the quasi-Newton method as well as the calls to two fluid subdomain models when interpolation between two ramp points of the transient cycle is needed. The results demonstrate a clear advantage of the Anderson mixing method on the selected thermally coupled example.

5. CONCLUDING REMARKS

In this paper we investigate six different nonlinear iterative techniques to solve thermal fluid-solid strongly coupled problems, i.e., Broyden rank-1 update, Newton-Krylov, Interface-GMRES, quasi-Newton with estimation of heat transfer coefficients, Aitken dynamic relaxation and Anderson

mixing methods. Their relative efficiency is compared on a test problem of rotating cooling air flow interacting with a jet engine component. The test case features subdomains with complicated realistic geometry under a wide range of operating conditions. The comparisons show an evident advantage in using the Anderson mixing scheme. It is demonstrated that the method can substantially reduce the total number of fluid solver invocations leading to shorter simulation times. The algorithm exhibits reasonably fast convergence rates, robustness and overall efficiency under a wide range of conditions. Another appealing characteristic of the Anderson approach is its ability to recycle iterations from previous time steps. Significant decrease in number of fluid solver calls is observed on the test case after incorporating data from one preceding time step. These results indicate that the method will be one of the first choices to solve practical large-scale fluid-solid thermal interaction problems. Further research is still needed to suggest some tentative guidelines leading to a more flexible strategy of time step data reuse to guarantee optimal performance.

6. ACKNOWLEDGEMENTS

The authors would like to acknowledge Rolls-Royce plc with Engineering and Physical Sciences Research Council for providing financial and technical support for this work.

REFERENCES

1. Patankar S. *Numerical Heat Transfer and Fluid Flow*. McGraw Hill: New York NY, 1980.
2. Hübner B, Dinkler D. A simultaneous solution procedure for strong interactions of generalized Newtonian fluids and viscoelastic solids at large strains. *International Journal for Numerical Methods in Engineering* 2005; **64**(7):920–939, doi:10.1002/nme.1396.
3. Piperno S, Farhat C, Laroutou B. Partitioned procedures for the transient solution of coupled aeroelastic problems. part i: Model problem, theory and two-dimensional application. *Computer Methods in Applied Mechanics and Engineering* 1995; **124**:79–112, doi:10.1016/0045-7825(95)92707-9.
4. Felippa C, Park K, Farhat C. Partitioned analysis of coupled mechanical systems. *Computer Methods in Applied Mechanics and Engineering* 2001; **190**:3247–3270, doi:10.1016/S0045-7825(00)00391-1.
5. Giles M. Stability analysis of numerical interface conditions in fluid-structure thermal analysis. *International Journal for Numerical Methods in Fluids* 1997; **25**:421–436, doi:10.1002/(SICI)1097-0363(19970830)25:4<421::AID-FLD557>3.0.CO;2-J.
6. Roe B, Haselbacher A, Geubelle P. Stability of fluid-structure thermal simulations on moving grids. *International Journal for Numerical Methods in Fluids* 2007; **54**:1097–1117, doi:10.1002/flid.1416.
7. Matthies H, Niekamp R, Steindorf J. Algorithms for strong coupling procedures. *Computer Methods in Applied Mechanics and Engineering* 2006; **195**:2028–2049, doi:10.1016/j.cma.2004.11.032.
8. von Scheven M, Ramm E. Strong coupling schemes for interaction of thin-walled structures and incompressible walls. *International Journal for Numerical Methods in Engineering* 2011; **87**(1–5):214–231, doi:10.1002/nme.3033.
9. Küttler U, Wall W. Fixed-point fluid–structure interaction solvers with dynamic relaxation. *Computational Mechanics* 2008; **43**(1):61–72, doi:10.1007/s00466-008-0255-5.
10. van Brummelen E. Partitioned iterative solution methods for fluid-structure interaction. *International Journal for Numerical Methods in Fluids* 2011; **65**:3–27, doi:10.1002/flid.2465.
11. Michler C, van Brummelen E, de Borst R. An interface Newton–Krylov solver for fluid–structure interaction. *International Journal for Numerical Methods in Fluids* 2005; **47**:1189–1195, doi:10.1002/flid.850.
12. Küttler U, Wall W. Vector extrapolation for strong coupling fluid–structure interaction solvers. *Journal of Applied Mechanics* 2009; **76**(6):021 205, doi:10.1115/1.3057468.
13. Michler C, van Brummelen E, de Borst R. An investigation of interface–GMRES(R) for fluid-structure interaction problems with flutter and divergence. *Computational Mechanics* 2011; **47**:17–29, doi:10.1007/s00466-010-0519-8.
14. Fernandez M, Moubachir M. A Newton method using exact Jacobians for solving fluid-structure coupling. *Computers and Structures* 2005; **83**(2–3):127–142, doi:10.1016/j.compstruc.2004.04.021.
15. Knoll D, Keyes D. Jacobian-free Newton–Krylov methods: A survey of approaches and applications. *Journal of Computational Physics* 2004; **193**:357–397, doi:10.1016/j.jcp.2003.08.010.
16. Martinez J. Practical quasi-Newton methods for solving nonlinear systems. *Journal of Computational and Applied Mathematics* 2000; **124**(1–2):97–121, doi:10.1016/S0377-0427(00)00434-9.
17. Minami S, Yoshimura S. Performance evaluation of nonlinear algorithms with line-search for partitioned coupling techniques for fluid-structure interactions. *International Journal for Numerical Methods in Fluids* 2010; **64**:1129–1147, doi:10.1002/flid.2274.
18. Yeckel A, Pandey A, Derby J. Fixed-point convergence of modular, steady-state heat transfer models coupling multiple scales and phenomena for melt crystal growth. *International Journal for Numerical Methods in Engineering* 2006; **67**:1768–1789, doi:10.1002/nme.1685.

19. Yeckel A, Lun L, Derby J. An approximate block Newton method for coupled iterations of nonlinear solvers: Theory and conjugate heat transfer applications. *Journal of Computational Physics* 2009; **228**:8566–8588, doi:10.1016/j.jcp.2009.08.003.
20. Hooper R, Smith T, Ober C. Enabling fluid–structure strong thermal coupling within a multi–physics environment. *44th AIAA Aerospace Sciences Meeting*, Reno, Nevada, US, 2006.
21. Fang H, Saad Y. Two classes of multiseccant methods for nonlinear acceleration. *Numerical Linear Algebra with Applications* 2009; **16**:197–221, doi:10.1002/nla.617.
22. Walker H, Ni P. Anderson acceleration for fixed–point iterations. *Technical Report MS-9-21-45*, Worcester Polytechnic Institute, Mathematical Sciences Department 2009. Submitted to *SIAM Journal of Numerical Analysis*.
23. Verdicchio J, Chew J, Hills N. Coupled fluid/solid heat transfer computation for turbine discs. *ASME Turbo Expo Paper 2001-GT-0205*, New Orleans, Louisiana, US, 2001.
24. Illingworth J, Hills N, Barnes C. 3D fluid–solid heat transfer coupling of an aero–engine preswirl system. *ASME Turbo Expo Paper 2005-GT-68939*, Reno, Nevada, US, 2005.
25. Chew J, Hills N. Computational fluid dynamics for turbomachinery internal air systems. *Philosophical Transactions of the Royal Society* 2007; **365**:2587–2611, doi:10.1098/rsta.2007.2022.
26. Verstraete T, Alsalihi Z, van den Braembussche R. Numerical study of the heat transfer in micro gas turbines. *Journal of Turbomachinery* 2007; **29**:835–841, doi:10.1115/1.2720874.
27. Mok D, Wall W. Partitioned analysis schemes for the transient interaction of incompressible flows and nonlinear flexible structures. *Trends in Computational Structural Mechanics*, Barcelona, Spain, 2001; 689–698.
28. Kelley C. *Solving Nonlinear Equations with Newton's Method*. SIAM: Philadelphia PA, 2003.
29. Eisenstat S, Walker H. Choosing the forcing terms in an inexact Newton method. *SIAM Journal on Scientific Computing* 1996; **17**(1):16–32, doi:10.1137/0917003.
30. Anderson D. Iterative procedures for nonlinear integral equations. *Journal of the Association of Computing Machinery* 1965; **12**:547–560, doi:10.1145/321296.321305.
31. Krysl P. *A Pragmatic Introduction to the Finite Element Method for Thermal and Stress Analysis: With the MATLAB Toolkit Sofea*. World Scientific: Hackensack NJ, 2006.
32. Moinier P. Algorithm development for an unstructured viscous flow solver. PhD Thesis, University of Oxford 1999.