J. U. Schlüter
X. Wu
S. Kim
S. Shankaran
J. J. Alonso
H. Pitsch

Center for Turbulence Research and Aerospace
Computing Lab,
Stanford University,
Stanford, CA 94305-3030

A Framework for Coupling Reynolds-Averaged With Large-Eddy Simulations for Gas Turbine Applications

Full-scale numerical prediction of the aerothermal flow in gas turbine engines are currently limited by high computational costs. The approach presented here intends the use of different specialized flow solvers based on the Reynolds-averaged Navier-Stokes equations as well as large-eddy simulations for different parts of the flow domain, running simultaneously and exchanging information at the interfaces. This study documents the development of the interface and proves its accuracy and efficiency with simple test cases. Furthermore, its application to a turbomachinery application is demonstrated.

[DOI: 10.1115/1.1994877]

1 Introduction

In the design of gas turbine engines computational fluid dynamics (CFD) is often used to predict the flow in single components of the engine, such as the compressor, the combustor, or the turbine. The simulation of the entire flow path of a gas turbine engine using high-fidelity CFD is deemed impossible by the enormous computational costs that it entails. However, the increasing availability of massively parallel computational resources and the improved algorithmic efficiency of future flow solvers puts the simulation of an entire engine within reach. In order for such a simulation to be useful in the design process it has to deliver accurate results with reasonable turnaround.

The goal of the advanced simulation and computing (ASC) program of the Department of Energy (DoE) at Stanford is to develop high-performance flow solvers that are able to use highly parallel supercomputers for the simulation of an entire engine. Although the development of new supercomputers is one of the main tasks in the overall ASC effort of the DoE, the physics part of the ASC project at Stanford investigates the development of flow solvers for gas turbine engines in order to improve efficiency, scalability, and modeling of physical effects. However, looking at the wide variety of flow phenomena, which have to be simulated in the flow path of an engine, it is clear that only the use of multiple specialized flow solvers (one for the turbomachinery portions and one for the combustor) can guarantee efficiency and accuracy of such a simulation: the flow regimes and the physical phenomena, which have to be modeled vary dramatically in these two components. Most flow solvers used nowadays in the design process are specialized for either of these two tasks.

The flow field in the turbomachinery portions of the domain is characterized by both high Reynolds and high Mach numbers. The accurate prediction of the flow requires the precise description of the turbulent boundary layers around the rotor and stator blades, including tip gaps and leakage flows. A number of flow solvers that have been developed to deal with this kind of problem have been in use in industry for many years. These flow solvers are typically based on the Reynolds-averaged Navier-Stokes (RANS) approach. Here, the unsteady flow field is ensemble averaged, removing all dependence on the details of the small-scale turbulence. A turbulence model becomes necessary to represent the portion of the physical stresses that has been removed during the averaging process. Because of the complexity of the flows in turbomachinery, various parameters in these turbulence models have to be adapted in order to deliver accurate solutions. Since this kind of flow has been the subject of a large number of investigations, these parameters are usually well known and, hence, the flow solvers deliver reasonably good results.

The flow in the combustor, on the other hand, is characterized by detached flows, chemical reactions, and heat release. The prediction of detached flows and free turbulence is greatly improved using flow solvers based on large-eddy simulations (LES). Although the use of LES increases the computational costs, it has been the only predictive tool that is able to simulate consistently these complex flows. LES resolves the large-scale turbulent motions in time and space and only the influence of smallest scales, which are generally more universal and hence, easier to represent has to be modeled [1,2]. Since the energy containing part of the turbulent scales is resolved, the modeling of turbulent combustion is facilitated by additional data that are provided by the LES solution [3]. LES flow solvers have been shown in the past to be able to model simple flames and are currently adapted for use in gas turbine combustors [4,5].

In order to predict multicomponent effects, such as compressor-combustor instabilities, combustor-turbine hot-streak migration, and combustion instabilities, the flow solvers that describe different components in the gas turbine have to run simultaneously, each computing its part of the domain, and periodically exchange flow information at the interface (Fig. 1). The simultaneous execution of multiple parallel flow solvers requires the definition of an interface that allows the exchange of flow information and a framework for well-posed boundary conditions in order to process the exchanged data.

The approach to couple multiple simulation codes has been used already in other areas of application, most notably in global climate simulations [6], and found recently more attention in other areas of mechanical engineering [7]. However, the idea to couple RANS and LES flow solvers is a very recent approach and a unique method to construct an LES-RANS hybrid.

Other LES-RANS hybrid approaches, such as detached-eddy simulations (DES) [8] and limited-numerical scales (LNS) [9] combine LES and RANS in a single flow solver. This requires a sensor that determines when to switch from one approach to the other. In our approach the domains are defined as zones by the computational domains of both codes. The use of two separate codes allows one to employ the optimal combination of a math-

Contributed by the Fluids Engineering Division for publication in the ASME Journal of Fluids Engineering. Manuscript received by the Fluids Engineering Division April 14, 2004. Final revision February 16, 2005. Associate Editor: Thomas B. Gatski.

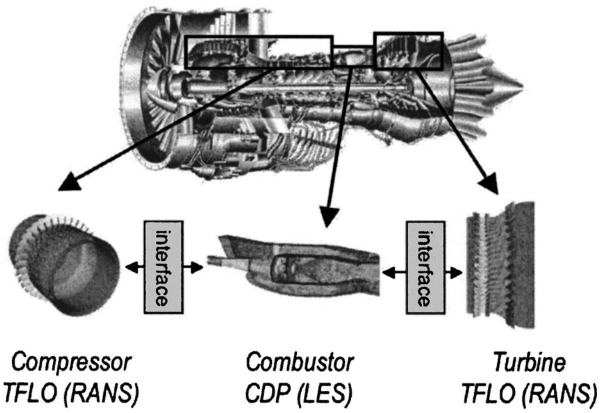


Fig. 1 Computation of the flow path of an entire gas turbine: decomposition of the engine. Compressor and turbine with RANS; combustor with LES (combustor and turbine images from [34,5])

ematical approach, numerical method, and models in each domain (e.g., the modeling of combustion requires the solution of additional transport equations). Here, we can limit the solution of these transport equations to the combustor. Furthermore, the time step in each domain can be chosen to fulfill the local requirements, which means that usually the RANS domain can be computed with a larger time step than the LES domain. The approach to couple two existing flow solvers also has the distinct advantage to build on the experience and validation that has been put into the individual codes during their development. Furthermore, once the procedures to couple independent simulation codes are in place, the extension of this concept to multiphysics simulations using other simulation tools can be done [10].

The current study describes the framework for the simultaneous execution of RANS and LES flow solvers and addresses the following points:

1. Description of the RANS and LES flow solvers (Sec. 2)
2. Presentation of the interface, which enables contact and information exchange between the simultaneously executed flow solvers (Sec. 3)
3. Description of the boundary conditions used by the flow solvers at the interfaces (Sec. 4)
4. Validation of the communication routines and the boundary conditions using simple test cases (Sec. 5)
5. Demonstration of coupled RANS-LES of complex geometries (Sec. 6)

2 Flow Solvers

This section describes the flow solvers used in the current study and emphasizes the differences between the RANS and LES approaches.

2.1 RANS Flow Solver. RANS flow solvers solve the classical Reynolds-averaged Navier-Stokes equations for turbulent flows. With this approach, the flow variables are split into mean and fluctuating portions $u_i=\bar{u}_i+u_i'$ and the Navier-Stokes equations are time averaged. This averaging process results in a set of equations for the mean flow quantities, but leaves an undetermined term $\overline{u_i'u_j'}$, which has to be modeled with a turbulence model. Turbulence models are commonly based on an eddy viscosity approach, which can be modeled with varying levels of complexity. The most commonly used models for RANS flow solvers are two-equation models, such as the $k-\epsilon$ or $k-\omega$ models, where two additional transport equations are solved in order to determine values and distribution of the eddy viscosity field. These models are typically accepted as a good compromise between efficiency and accuracy for turbomachinery applications.

The RANS flow solver used for this investigation is the TFLO code developed in the Aerospace Computing Lab (ACL) at Stanford. The flow solver computes the unsteady Reynolds-averaged Navier-Stokes equations using a cell-centered discretization on arbitrary multiblock meshes [11]. The convective terms are discretized using central differences (second-order accurate on smooth meshes). In order to maintain numerical stability artificial dissipation is added. The solution procedure is based on efficient explicit modified Runge-Kutta methods with several convergence acceleration techniques, such as multigrid, residual averaging, and local time stepping. These techniques, multigrid, in particular, provide excellent numerical convergence and fast solution turnaround. The turbulent viscosity is computed with the Wilcox $k-\omega$ two-equation turbulence model [12]. The dual-time stepping technique [13–15] is used for time-accurate simulations that account for the relative motion of moving parts as well as other sources of flow unsteadiness.

2.2 LES Flow Solver. LES flow solvers solve a filtered version of the Navier-Stokes equations. The filter ensures that the large-scale turbulence is resolved in time and space, which results in a decomposition of the flow variables into a resolved and a subgrid portion, $u_i=\tilde{u}_i+u_i''$. For practical purposes, a mesh filter is applied, implying that the local cell size defines the filter at each point in the mesh. Applying the filter to the Navier-Stokes equation leaves an undetermined term, $\widetilde{u_i''u_j''}$, which defines the subgrid turbulence that must be modeled. As opposed to the Reynolds stress term $\overline{u_i'u_j'}$ in the RANS equations, which includes the turbulent motions of all scales, the LES term describes only the subgrid turbulence. With sufficiently high mesh resolution, the LES solution can be very robust with respect to the chosen subgrid model. Most models use an eddy viscosity approach for the description of the subgrid stresses. Typically the eddy viscosity is determined by algebraic models, such as the Smagorinsky model [16], or, as used in this study, by a dynamic procedure, where the solution of the high-frequency resolved flow field is used to determine the subgrid stresses [17].

For the initial development of the interface two separate LES flow solvers are used. The first one is a structured LES flow solver, which has the advantage of very fast execution speeds. The second LES flow solver used is the unstructured CDP code, which is used for its ability to resolve complex geometries.

The structured LES flow solver chosen for this work is a code developed at the Center for Turbulence Research (CTR) at Stanford by Pierce and Moin [18]. It solves the filtered momentum equations with a low-Mach-number assumption on an axisymmetric, structured, single-block mesh. A second-order finite-volume scheme on a staggered grid is used [19]. Centerline boundary conditions for the radial velocity and its gradients are obtained by averaging corresponding values across the centerline [20]. The approach is designed to allow radial flow communication through the centerline. The low-Mach-number approximation allows one to circumvent the acoustic Courant-Fredriechs-Lewey (CFL) condition for compressible flows and increases the permissible timestep by at least 1/Ma. In return, the pressure field has to be determined by solving the Poisson equation. The subgrid stresses are approximated with an eddy viscosity approach, where the eddy viscosity is determined by a dynamic procedure [17,21].

The unstructured LES flow solver CDP has been developed at the Center for Turbulence Research at Stanford [22]. The filtered momentum equations are solved on a cell-centered unstructured mesh with a second-order accurate central differences spatial discretization [23]. An implicit time-advancement procedure is applied. As in the structured flow solver, a low-Mach-number approximation is used and the Poisson equation is solved in order to

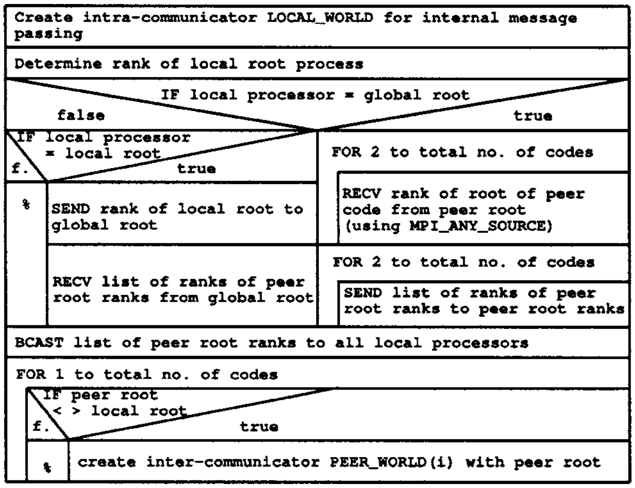

Create intra-communicator LOCAL_WORLD for internal message passing
Determine rank of local root process
IF local processor = global root
false
true
IF local processor = local root
f.
true
%
SEND rank of local root to global root
RECV list of ranks of peer root ranks from global root
FOR 2 to total no. of codes
RECV rank of root of peer code from peer root (using MPI_ANY_SOURCE)
FOR 2 to total no. of codes
SEND list of ranks of peer root ranks to peer root ranks
BCAST list of peer root ranks to all local processors
FOR 1 to total no. of codes
IF peer root < > local root
f.
true
%
create inter-communicator PEER_WORLD(i) with peer root

Fig. 2 Structure chart: exchange of root ranks needed for creation of intercommunicators

determine the pressure field. The subgrid stresses are modeled with a dynamic procedure.

3 Interface

The role of the interface is to establish the communication between two or more simultaneously executed flow solvers and to enable the efficient transfer of flow variables among all of them. In the following sections, the interface routines are described together with their implementation in the previously described RANS and LES flow solvers. Careful attention is paid to the fact that all flow solvers are parallelized using the message-passing interface (MPI) and that the execution of these codes is usually carried out on massively parallel supercomputers [24].

3.1 Peer-to-Peer Message Passing. The messages between two separate flow solvers (peer-to-peer message passing) is very similar to the information exchange between processors of a parallel computation. Many flow solvers are parallelized and use MPI for process-to-process message passing. MPI can be used for communication between different flow solvers as well.

Before establishing the contact between two flow solvers, one must make sure that the commands for message passing that are internal to each of the two codes do not interfere with the communication between codes. With MPI it is possible to define the scope of the message passing using communicators. The most commonly used communicator in MPI is the standard communicator MPI_COMM_WORLD, which includes all processors of all codes started from the same MPIRUN command. Using this communicator for internal message passing will inevitably result in confusion for communication between the two codes. Hence, each code creates its own local communicator (intracommunicator) to encapsulate the internal message passing. All codes have to use their own intracommunicator for all MPI commands concerning the internal parallelization of the code instead of MPI_COMM_WORLD.

In the next step, a communicator is created for the peer-to-peer message passing (intercommunicator). For example, assume a case with three flow solvers is to be run with a first instance of a RANS code using two processors (ranks 0 and 1, local root process 0), an LES code using four processors (ranks 2, 3, 4, and 5, local root process 2), and a second instance of a RANS code using three processors (ranks 6, 7, and 8, local root process 6). In order to create the intercommunicator, it is necessary that every processor knows the rank of the root processes of the other codes. A global root process is appointed (rank 0) that collects the ranks of the root processes of all codes (here: ranks 0, 2, and 6), compiles them into a list, and sends them back to the local root processes. A chart with the structure of this procedure is shown in Fig. 2. Since there is no intercommunicator available yet, this communication has to be done using the standard communicator MPI_COMM_WORLD. With the knowledge of the ranks of all root processes it is then possible to create the intercommunicators.

3.2 Handshake and Communication.

3.2.1 Handshake. The efficient parallelization of a flow solver seeks to limit the information exchange between parallel processes to a minimum, since the information exchange requires considerable time compared to the actual computation and can, therefore, limit parallel scalability. For similar reasons, it is desirable to minimize the communication between flow solvers running simultaneously. Since the flow solvers have to exchange flow information rather often, either after each iteration or after a chosen time step, the aim is to minimize the amount of information communicated at each synchronization point by including an initial handshake step, which serves to optimize the communication during the actual flow computation.

The simplest way to organize the information exchange between solvers would be to let only the root processes in each solver communicate. However, this would mean that prior to the peer-to-peer communication, the root processes would have to gather the flow information to transfer from their own processes, and after the peer-to-peer communication process is complete, they would have to distribute the obtained information back to their processes. This creates an obvious bottleneck in the communication pattern, which must be avoided. The solution reported here avoids this bottleneck by direct communication among the neighboring processors on the interface.

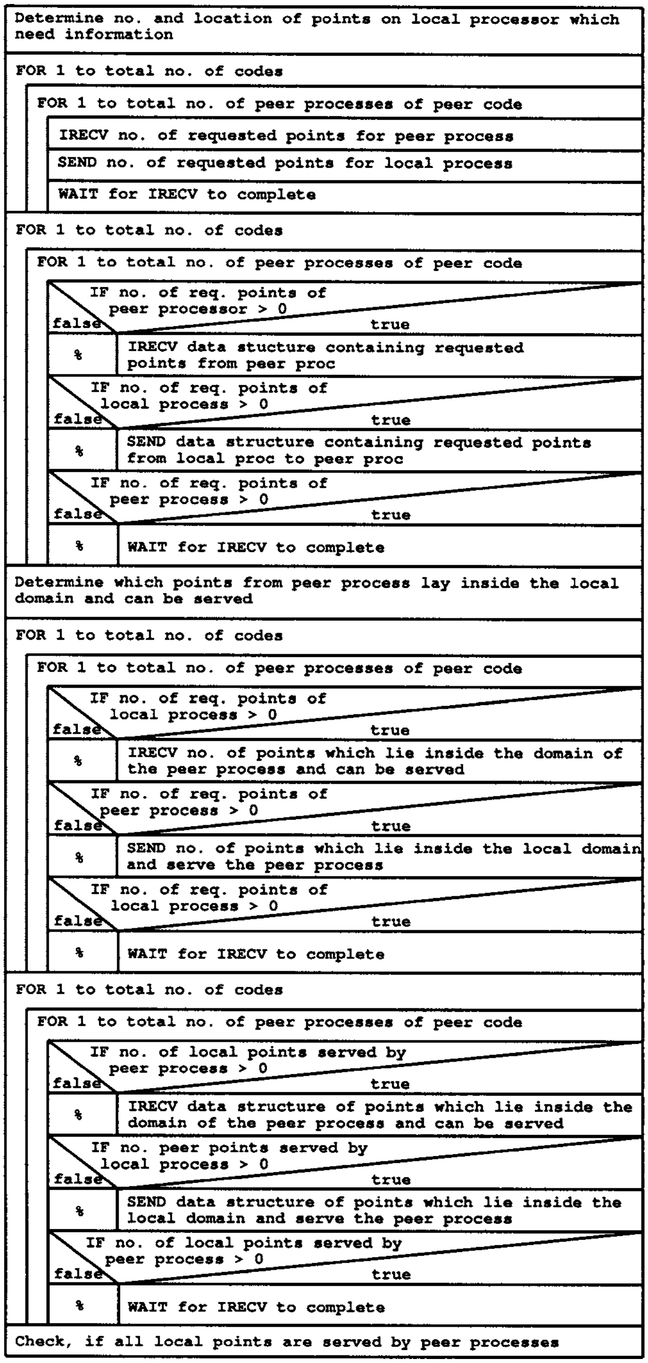
The initial handshake routine establishes the direct communication pattern described above (Fig. 3). First, each processor of each participating code must identify all the points for which it needs flow information from its peers to define its interface boundary conditions. The location of each of these points has to be stored in a data structure containing three integers and three double-precision values. The three integers are an *ip* number, which determines what kind of flow variables are requested for this point; an *id* number, which contains a unique identification number for each point; and a *flow solver* number denoting the flow solver requesting this point. The three double precision numbers contain the x-, y-, z-coordinates of the point in Cartesian coordinates using SI units.

The initial handshake takes place in four steps. First, each processor sends the number of points in its own domain for which flow data will be requested to each processor of the peer code. This allows each code to dynamically allocate memory to store the information received. In the second step, each processor receives information containing the location in space of the requested points from each of the peer processors that request a nonzero number of points.

In an intermediate step, each processor identifies whether a requested point lies within its own domain and data can be provided for it. During this step, the interpolation schemes required to obtain the data for this point are determined and stored for later use.

In the third communication step, each processor communicates the number of points found within its domain to all peer-processes requesting data. Again, this allows the dynamic allocation of arrays for the last step. In the fourth communication step, each processor sends out an array to each peer processor it can serve. The array consists of two integers containing the *ip* and *id* of the point. Finally, each processor determines whether all of its requested points can be served by peer processors. If not all points can be served, appropriate errors are flagged.

3.2.2 Communication. The communication of flow data between iterations is rather straight forward once the initial handshake is completed (Fig. 4). Since it is known to every processor what kind of data has to be provided to which peer processor, and from which peer processor data is expected, the data packages can

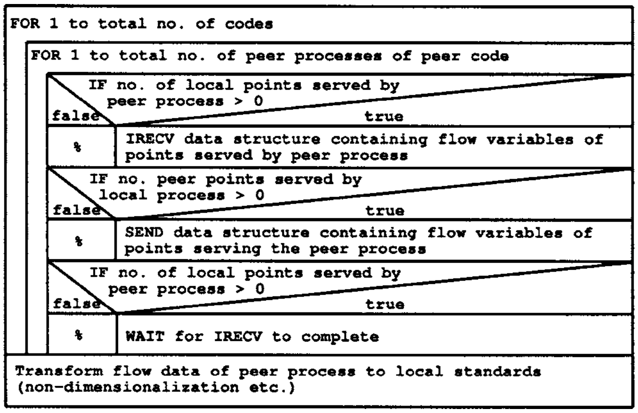


```
Determine no. and location of points on local processor which need information
FOR 1 to total no. of codes
    FOR 1 to total no. of peer processes of peer code
        IRECV no. of requested points for peer process
        SEND no. of requested points for local process
        WAIT for IRECV to complete
FOR 1 to total no. of codes
    FOR 1 to total no. of peer processes of peer code
        IF no. of req. points of peer processor > 0
            true
                IRECV data stucture containing requested points from peer proc
            false
                %
        IF no. of req. points of local process > 0
            true
                SEND data structure containing requested points from local proc to peer proc
            false
                %
        IF no. of req. points of peer process > 0
            true
                WAIT for IRECV to complete
            false
                %
Determine which points from peer process lay inside the local domain and can be served
FOR 1 to total no. of codes
    FOR 1 to total no. of peer processes of peer code
        IF no. of req. points of local process > 0
            true
                IRECV no. of points which lie inside the domain of the peer process and can be served
            false
                %
        IF no. of req. points of peer process > 0
            true
                SEND no. of points which lie inside the local domain and serve the peer process
            false
                %
        IF no. of req. points of local process > 0
            true
                WAIT for IRECV to complete
            false
                %
FOR 1 to total no. of codes
    FOR 1 to total no. of peer processes of peer code
        IF no. of local points served by peer process > 0
            true
                IRECV data structure of points which lie inside the domain of the peer process and can be served
            false
                %
        IF no. peer points served by local process > 0
            true
                SEND data structure of points which lie inside the local domain and serve the peer process
            false
                %
        IF no. of local points served by peer process > 0
            true
                WAIT for IRECV to complete
            false
                %
Check, if all local points are served by peer processes
```

Fig. 3 Structure chart: initial handshake to establish direct communication between interface processors

be sent directly without going through the root processes. After the initialization step, all communication is carried out in the most efficient possible pattern.

Each processor has to compile the data to be sent into a send buffer. The contents of this buffer may vary for different flow solvers and has to be defined beforehand. Although our communication procedure allows for flexible contents of the communication buffers, a standard data structure made up of seven variables has been established. These variables contain ρ, ρu_1, ρu_2, ρu_3, ρE, k, and ν_t, in this order, with ρ being the density, u_1, u_2, u_3 the Cartesian velocity components, E the total energy of the fluid, k the turbulent kinetic energy, and ν_t the eddy viscosity. This set of variables provides the freedom to chose between several RANS turbulence models without changing the interface routines, e.g. boundary conditions can be defined from this set of data for both the $k-\epsilon$ and $k-\omega$ turbulence models.



```
FOR 1 to total no. of codes
    FOR 1 to total no. of peer processes of peer code
        IF no. of local points served by peer process > 0
            true
                IRECV data structure containing flow variables of points served by peer process
            false
                %
        IF no. peer points served by local process > 0
            true
                SEND data structure containing flow variables of points serving the peer process
            false
                %
        IF no. of local points served by peer process > 0
            true
                WAIT for IRECV to complete
            false
                %
Transform flow data of peer process to local standards (non-dimensionalization etc.)
```

Fig. 4 Structure chart: communication of flow data during flow computations

4 Boundary Conditions

4.1 LES Boundary Conditions. The definition of the boundary conditions requires special attention on the LES side due to the different modeling approach used when compared to the RANS solver. Since on the LES side we resolve a part of the turbulent energy spectrum, the challenge is to regenerate and preserve the resolved turbulent motions at the boundaries.

4.1.1 LES Inflow Boundary Conditions. At the LES inflow boundary, the challenge is to prescribe transient turbulent velocity profiles from ensemble-averaged RANS data. A simple addition of random fluctuations to the RANS profiles misses the temporal and spatial correlations of real turbulence, and the fluctuations dissipate very quickly. Instead, we have chosen to create a database of turbulent fluctuations using an auxiliary LES computation of a periodic turbulent pipe flow. The LES inflow boundary condition can then be described as

$$u_{i,\mathrm{LES}}(t)=\underbrace{\bar{u}_{i,\mathrm{RANS}}(t)}_{I}+\underbrace{(u_{i,\mathrm{DB}}(t)-\bar{u}_{i,\mathrm{DB}})}_{II}\cdot\underbrace{\frac{\sqrt{\overline{u'^2_{(i)}}_{\mathrm{RANS}}(t)}}{\sqrt{\overline{u'^2_{(i)}}_{\mathrm{DB}}}}}_{III} \qquad (1)$$

where the subscript RANS denotes the solution obtained from the RANS computation, and quantities with subscript DB are provided from the database. Here, t is the time, u_i stands for the Cartesian velocity components, and $\bar{u}_i$ is the ensemble average of the velocity component u_i.

Term II of Eq. (1) is the velocity fluctuation of the database. This turbulent fluctuation is scaled to the desired value through the multiplication by term III, which ensures that the correct level of velocity fluctuation is recovered.

Since the RANS flow solver using a two-equation turbulence model cannot provide all Reynolds stresses, the normal stresses are approximated by

$$\overline{u'^2_{(i)}}_{\mathrm{RANS}}=\tfrac{2}{3}k \quad \text{with} \quad i=1,2,3 \qquad (2)$$

with (i) denoting that no summation of the components is made.

This boundary condition has been validated thoroughly in a previous study [25].

4.1.2 LES Outflow Boundary Conditions. In order to take into account upstream effects of the downstream flow development, the LES outflow conditions have to be defined so that the mean flow properties of the unsteady LES solution can be specified to match the statistical properties delivered by a downstream RANS simulation. A method that has been tested in the past employs virtual body forces in the momentum equations to drive the mean velocity field of the LES solution to a RANS target velocity field. The virtual body force is given by

$$F_i(\mathbf{x},\mathbf{t}) = \sigma[\bar{u}_{i,\mathrm{RANS}}(\mathbf{x},\mathbf{t}) - \bar{u}_{i,\mathrm{LES}}(\mathbf{x},\mathbf{t})] \tag{3}$$

where $\bar{u}_{i,\mathrm{RANS}}$ is the solution provided by the RANS flow solver, which is computed in an overlap region between the LES and RANS domains, and $\bar{u}_{i,\mathrm{LES}}$ is a time average of the LES solution over a trailing time window. The time variable t is the time according to the LES (changes with every LES time step), and τ is the time according to the RANS time step (changes with every RANS time step). The body force constant σ determines the strength of the body force. Its value can be estimated by a one-dimensional (1D) Euler analysis [26] and its minimum is given by

$$\sigma_{\min} = \frac{u_B}{l_F}\ln\left(\frac{|u_0 - u_t|}{\epsilon u_t}\right) \tag{4}$$

with u_B the bulk velocity, l_F the length of the forcing region, u_0 an estimate for the unforced solution, and u_t the target solution.

This body force ensures that the velocity profiles at the outlet of the LES domain fulfill the same statistical properties as the velocity profiles in the overlap region computed by a downstream RANS simulation. This makes it possible to take upstream effects of downstream flow alterations into account. This LES outflow condition has been validated in previous work [27].

The numerical outflow conditions at the LES outflow are determined by the so-called convective outflow condition

$$\frac{\partial \phi}{\partial t} + u_c \frac{\partial \phi}{\partial n} = 0 \tag{5}$$

where ϕ is any scalar or velocity component, u_c is the convective velocity, and n is the coordinate in the direction of the outward normal at the boundary. The pressure at the outlet adjusts, accordingly, to the velocity distribution determined by the Poisson equation and, hence, it cannot be prescribed. Instead, the proper pressure conditions are adjusted using Eq. (3).

4.2 RANS Boundary Conditions. The specification of RANS boundary conditions from LES data is essentially straightforward. The unsteady LES flow data are time averaged over the time step used by the RANS flow solver and can be employed directly as a boundary condition.

In the current study, the compressible formulation of the RANS flow solver and the low-Mach-number formulation of the LES code posed a challenge. Although the RANS code allows for acoustic waves to propagate within the limits of its domain, the density field of the LES solution is entirely defined by mixing and the combustion process and not by acoustics. This leads to the need for RANS inflow and outflow conditions that allow acoustic waves to leave the domain without spurious reflections. The construction of these boundary conditions is nontrivial, particularly for viscous flows, and must allow for variations of the flow variables at the interface locations. Currently, the local one-dimensional inviscid (LODI) relations [28] are applied.

In the case of the LES domain upstream of the RANS domain, the RANS flow solver has to define its inflow conditions from the LES data. For every point of the inlet plane the mass-flux vector $(\rho u, \rho v, \rho w)$ is imposed, delivered by the LES computation. This allows the density ρ to fluctuate to account for the passing of acoustic waves. The velocity components u, v, w are adjusted accordingly in order to conserve the mass flux. Variations of ρ are in the order of $<2\%$ for typical Mach numbers.

For the RANS turbulence model, k is delivered by the LES solution as the turbulent kinetic energy of the resolved turbulence. The second variable ω is currently set constant at the interfaces, since it was found that it is difficult to retrieve a meaningful approximation of this variable from the LES solution.

In the other case, where the LES domain is downstream of the RANS, at the outlet the static pressure is defined by the LES and applied as the outflow boundary condition to account for the influence of a downstream LES flow solution.

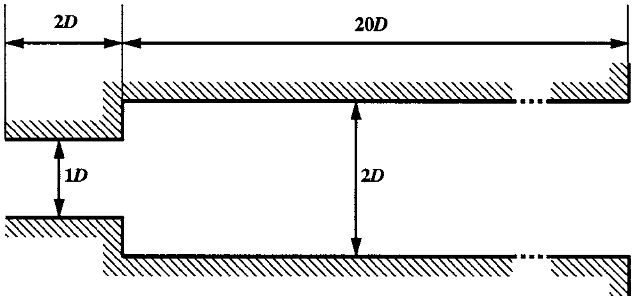


Fig. 5 Interface validation: geometry of the experimental test section [29]

5 Interface Validation

In order to validate the interface for gas turbine applications, two different scenarios have to be validated. The first one corresponds to the upstream interface between the compressor and the combustor. Then, the upstream flow solver is the RANS flow solver (for the compressor geometry), while the downstream domain (the combustor) is solved using an LES solver, which has to define its inflow boundary condition from the upstream RANS solution. The RANS flow solver obtains its outflow boundary condition from the LES solution. The second scenario corresponds to the downstream interface between the combustor and the turbine. Conversely, the LES flow solver is now upstream and has to define its outflow conditions according to the solution of the RANS solver downstream. The RANS flow solver obtains its inflow conditions from the upstream LES.

5.1 Upstream Interface: RANS-LES. In order to validate the upstream interface and the LES inflow boundary condition, a coupled RANS-LES computation of an axisymmetric expansion has been performed. The geometry is shown in Fig. 5 and has an expansion ratio of 1:2 leading to an area ratio of 1:4. The Reynolds number based on the upstream diameter and the bulk velocity is Re=30,000. This test case corresponds to a well-documented experimental configuration [29], and extensive experimental data are available upstream and downstream of the expansion. This allows for an accurate definition of the inflow flow parameters and an assessment of the simulated flow development.

Although this flow has been computed earlier using LES [25], here an additional level of complexity is added by using a coupled RANS-LES approach. While this has no particular advantage for the current test case, it allows one to validate the communication routines and the boundary conditions on both sides of the interface. A part of the flow domain upstream of the expansion is computed with the RANS code TFLO, whereas the flow at the expansion is computed by the LES flow solver CDP (Fig. 6).

With the origin of the coordinate system at the center of the expansion, the inlet velocity profiles in the RANS section are

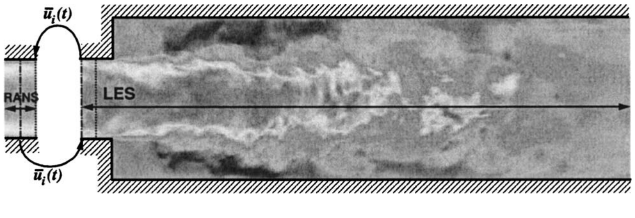


Fig. 6 Interface validation: integrated RANS-LES of a confined jet

specified at $x/D=-0.75D$ according to the experimental data interpolated between $x/D=-2.0$ and $x/D=-0.5$. The variation of these two measured velocity profiles is minimal. The RANS flow solver TFLO computes the flow through the upstream pipe and transfers the data at its outlet to the subsequent LES flow solver. The RANS domain is relatively short $(0.5D$, with D being the diameter of the pipe upstream of the expansion) in order to reproduce the experimental flow field at the inlet of the LES domain as closely as possible. At the RANS outlet the static pressure delivered by the LES solution is imposed.

The LES domain starts at $x/D=-0.5D$, which results in an overlap region between the RANS and LES of $\Delta x/D=0.25$. The LES flow solver CDP obtains its inflow velocity profiles from the RANS flow solver and specifies its LES inflow boundary conditions according to Eq. (1). An overlap region between the two domains was used in order to ensure that the data collection for the communication of flow variables is far enough from the boundary of the computed domain, where the results are influenced by the convective boundary condition.

The RANS mesh contains 350,000 mesh points and is refined near the wall. The LES mesh contains 1.1 million mesh points with the mesh points concentrated near the spreading region of the jet. The mesh has an H–O topology over the pipe cross section: an O mesh near the pipe walls allows for the proper resolution of the wall, while an H-mesh in the center is used to ensure the resolution of the centerline. The far field of the jet is intentionally left relatively coarse in order to decrease computational costs.

The simulation was run for approximately five flow-through times before collecting flow statistics for another five flow-through times. The integrated simulation was computed using five processors for the RANS domain and 24 processors for the LES domain. The exchange of flow information between the two flow solvers was performed with the interface frequency $f_{\text{INTERFACE}}=1/\Delta t_{\text{RANS}}$ using a RANS time step of $\Delta t_{\text{RANS}}=0.1\times D/U_{\text{BULK}}$. The LES time step was determined by the convective CFL condition and was set to $\Delta t_{\text{LES}}=0.01\times D/U_{\text{BULK}}$. This results in a time-step ratio of $\Delta t_{\text{RANS}}/\Delta t_{\text{LES}}=10$. The computation of a single flow through time on an IBM SP3 using 29 processors requires ~2.5 h of wall-clock time, adding up to a total wall-clock time of 25 h for the entire simulation.

The quality of the results of the integrated RANS-LES computation has been assessed by two means. First, the numerical results have been compared against the experimental data. Second, a LES of the entire domain using only the LES code has been performed, where the inflow parameters have been specified according to the measurements at $x=-0.5$. This allows a comparison of the integrated RANS-LES results with a LES computation and the assessment, whether errors have been introduced by uncertainties of the LES approach in the region of the jet or by the coupled RANS-LES approach.

Figure 7 shows the velocity profiles obtained from this computation. The mean velocity distribution of the integrated RANS-LES computation agrees very well with the experimental data and the LES computation. The spreading of the jet and the reattachment of the flow are well predicted by both simulations.

The axial velocity fluctuations are also well predicted in the near field of the jet by both simulations. The far field of the jet is underresolved, and hence the numerical predictions underpredict the turbulence levels. Since the error appears in both simulations, this error can be associated to the mesh resolution and is not caused by the coupling of the RANS and LES flow solver.

This computation validates the coupled RANS-LES approach for the case, where the RANS domain is upstream of the LES calculation.

5.2 Downstream Interface: LES-RANS.

In order to verify the interface for this second scenario where the LES domain is upstream of the RANS domain, a swirl flow is considered. The computation of a swirl flow presents a challenging test case for the validation of the interface and the boundary conditions because of the complexity of the flow and its sensitivity to inflow and outflow parameters. The sensitivity of the upstream flow field allows for a fair assessment of the interface because it ensures that the downstream RANS solution is of relevance for the upstream flow development computed by LES. Yet, this test case is simple enough to perform a LES computation of the entire domain in order to obtain a solution, which serves as a reference to assess the accuracy of integrated computations. Here, the structured LES flow solver and RANS flow solver TFLO are used.

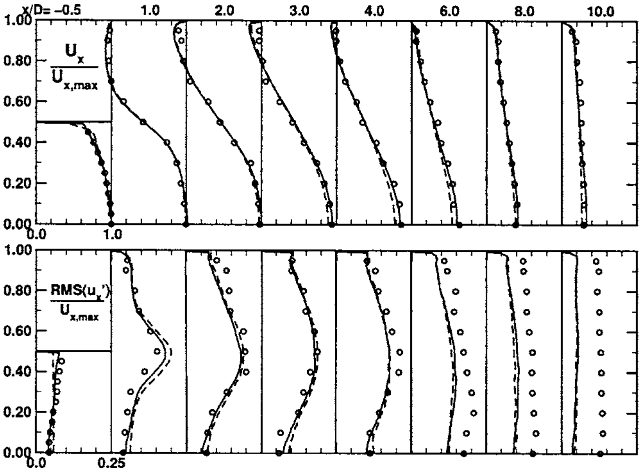


Fig. 7 Results of interface validation. Above: axial velocity profiles. Below: axial velocity fluctuations. Circles: experiments. Solid lines: LES with inflow from experimental data. Dashed lines: integrated RANS-LES, RANS with inflow from experimental data, LES inflow derived from simultaneously running RANS solver.

A swirl flow at an expansion with a subsequent contraction three diameters D downstream of the expansion is considered (Fig. 8(*a*)). The inlet velocity profiles are taken from an the experiment described in the previous test case [29,30]. The swirl number of the flow is $S=0.3$, which is just supercritical, meaning that vortex breakdown takes place and a recirculation zone develops. The Reynolds number for this flow configuration was set to Re=20,000. The flow conditions were chosen for maximum sensitivity to boundary conditions in order to create a challenging test case for the validation of the interface and the LES outflow boundary conditions.

The extension and strength of this recirculation zone is strongly influenced by the presence of the downstream contraction, which is to be resolved by the RANS flow solver. Unless the effect of the RANS solution is correctly transferred by the interface boundary conditions, the details of the recirculation region in the LES do-

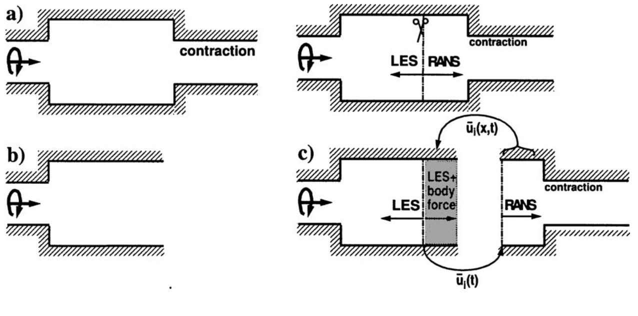


Fig. 8 Geometry for integrated LES/RANS computations: (*a*) full geometry, (*b*) reduced LES domain, and (*c*) schematic splitting of domain to two computational domains

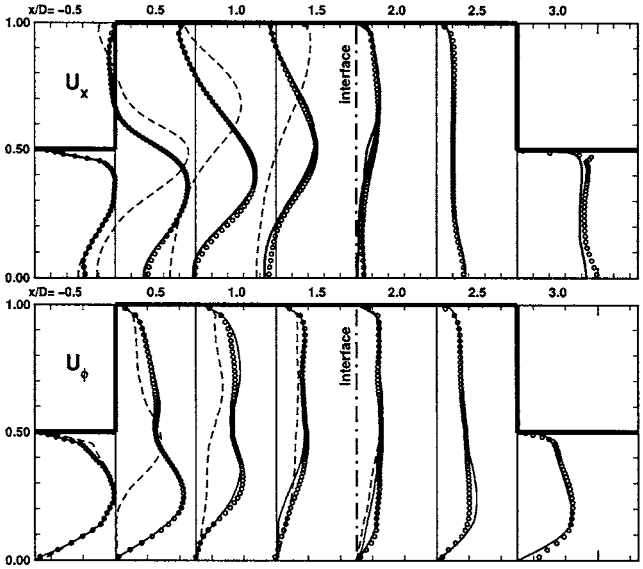


Fig. 9 Integrated LES/RANS computations. Velocity components for different downstream positions. Circles: LES of full geometry (Fig. 8(*a*)) dashed line: LES of expansion (Fig. 8(*b*)) solid line: integrated LES-RANS computation (Fig. 8(*c*)).

main will not be computed correctly.

The first calculation of this study is carried out using the LES solver for the entire domain, including the expansion and the contraction. This computation is considered the reference solution, and the accuracy of all subsequent computations is measured against these results. The following computations consider that this domain is to be computed by two or more separate flow solvers. The geometry is divided into two computational domains with a short overlap region. The expansion is computed with the LES code, whereas the contraction is computed by the RANS solver (Fig. 8(*c*)). If the coupling of the two codes is done appropriately, then this coupled simulation should recover the solution of the LES performed for the entire domain.

The mesh of the LES domain computing the entire domain contains $386\times64\times64$ ($\approx$1.5 million) cells and is refined near the walls and shear layers of the swirl flow. The mesh for the LES domain for the coupled LES-RANS computations consists of $256\times64\times64$ ($\approx$1.0 million) cells and closely resembles the LES mesh for the entire domain. The flow was computed for ten flow-through times before collecting flow statistics for another five flow-through times. The LES of the entire domain computes one flow-through time in 2 h wall-clock time on 12 processors on an SGI Origin 2000. The coupled LES-RANS computation computes the same physical time span in 1.4 h using eight processors for the LES domain and five processors for the RANS domain.

The RANS time step was chosen to $\Delta t_{\mathrm{RANS}}=0.1\times D/U_{\mathrm{BULK}}$, which defines the interface frequency. The LES time step in this computation was varying between different iterations in order to allow for a maximum time step according to the CFL condition. In order to ensure an accurate synchronization of the two flow solvers, the LES time step preceding the communication was adjusted to match the RANS time. The ratio of the time steps was approximately $\Delta t_{\mathrm{RANS}}/\Delta t_{\mathrm{LES}}\approx 12-15$.

Figure 9 shows the velocity profiles for three different computations. The velocity profiles denoted by the circles represent the LES computation of the entire domain (Fig. 8(*a*)) and hence, the target for the integrated computations.

In order to demonstrate the influence of the contraction on the swirl flow at the expansion, the dashed lines show the velocity profiles of a LES computation of the expansion without the computation of the contraction (Fig. 8(*b*)). Convective outflow conditions (Eq. (5)) are used without a body-force treatment. It can be seen that the obtained velocity field differs substantially from the first simulation, and hence the influence of the downstream contraction cannot be neglected.

The solid lines in Fig. 9 show the integrated LES-RANS computation using two flow solvers for the two domains (Fig. 8(*c*)). The location of the interface is denoted with a dotted-dashed line. The velocity profiles on the left-hand side of the interface are computed with LES and the profiles on the right-hand side with RANS. The RANS computation of the subsequent contraction delivers a mean flow field, which is used to correct the outflow conditions of the upstream LES. As a result, the velocity profiles of the integrated LES-RANS computation tend toward the velocity profiles of the LES of the entire domain. Because of the large RANS time step, in the RANS domain the energy of resolved turbulence is negligible compared to the energy of the modeled turbulence.

This test case demonstrates that the downstream development of the flow can have a substantial influence on the upstream flow development. The coupled RANS-LES approach is able to predict the downstream flow development with a RANS flow solver and transfer its effect to the upstream LES. Although in this particular test case a LES of the entire domain was feasible, the value of coupled LES-RANS is apparent in more complex applications such as gas turbines, where LES is not always feasible and some components might have to be computed with RANS.

6 Demonstration of Integrated RANS-LES in Complex Geometries

In order to demonstrate the applicability of coupled RANS-LES computations in realistic gas turbine geometries, a turbomachinery case has been investigated [31]. The goal of this study is to test the interface routines for the flow between the compressor and the combustor and to study the influence of possible unsteady interactions between the compressor and the combustor inlet diffuser. The test case consists of a compressor geometry computed by a RANS flow solver and a prediffuser (the component upstream of the injector to the combustor) computed by a LES flow solver.

The computational study of such cases is relevant and important, since typically these two components are developed in isolation and combined tests are typically done only in a final prototype assembly. Yet, the upstream compressor has a substantial influence on the diffuser performance [32,33]. The numerical prediction of this flow configuration will allow for an assessment of the interactions of the components during the design phase. One of the most important questions for compressor-prediffuser flows is whether separation in the diffuser takes place. Since the inflow of the prediffuser is inhomogeneous and periodically perturbed by blade passings, the integrated computation of this geometry can offer insight into how to modify the geometry in order to develop a more compact, attached diffuser.

The drawback of the choice of this configuration is that no experimental data are available to validate the computation. The quality of the computed results can only be guaranteed on the basis of the separate validation process that the component codes have undergone and the detailed testing of the interface routines that has been presented in this and previous work. Some validation studies of the individual flow solvers are given in Yao et al. [11] and Davis et al. [34,35] for the TFLO code and in Constantinescu et al. [5], Ham et al. [36] and Moin and Apte [22] for the CDP code. The interface has been developed and tested, in detail, in previous sections. Although many of the techniques necessary for the coupling of these two flow solvers are still under development, all necessary elements, such as the coupling procedure and the boundary conditions on both sides, are currently in place for the chosen test case.

The goal of this computation is to demonstrate the feasibility of integrated RANS-LES computations in a turbomachinery environment and to identify practical issues involved in these calculations.

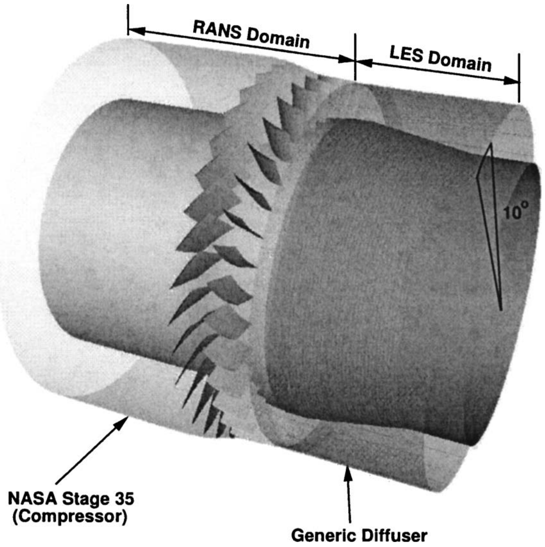


Fig. 10 Geometry of coupled NASA stage 35/prediffuser. RANS domain includes inflow channel, one rotor, and one stator. LES domain includes the diffuser. A 10 deg axisymmetric sector is computed.

6.1 Geometry. The compressor geometry for this test case corresponds to that of a modified version of the NASA Stage 35 experimental rig. The one-stage experimental rig consists of 46 rotors followed by 36 stators. In order to simplify this geometry, the rotor row has been rescaled to have only 36 blades, which allows us to compute an axisymmetric segment of 10 deg using periodic boundary conditions at the corresponding azimuthal planes (Fig. 10).

For this integrated computation, the rotor tip gap has been closed in order to decrease the overall computational cost. The inclusion of the tip gap can be addressed in the TFLO flow solver and poses no additional problems from the integration point of view. The RANS time step was chosen to resolve one blade passing with 50 intervals.

The RANS mesh is a structured multiblock mesh consisting of ~1.5 million control volumes. The speed of the rotor was set to a relatively low 5000 rpm in order to keep the flow at the interface within the low-Mach-number regime that the LES solver is able to handle. This decrease in rotational speed had to be done for the current case. In a real engine, the compressor consists of multiple stages resulting in a higher pressure and higher temperature at the compressor exit. The high temperature of the air in this section of the flow path will ensure that the low-Mach-number approximation is not violated, even when the engine is at full load.

For the RANS domain, the flow solver TFLO has been used. On the LES side, the LES flow solver CDP has been applied.

The diffuser expansion begins one stator chord length behind the stator. The LES domain starts $\frac{1}{3}$ chord behind the stator. The RANS domain reaches $\frac{2}{3}$ of the chord length into the LES domain, which essentially means that the RANS outlet plane is right at the beginning of the expansion of the diffuser.

The diffuser geometry has been chosen with a relatively wide opening such that separation may occur. The diffuser opens toward the centerline of the compressor. Over three chord lengths, the diffuser opens up 0.5 chord lengths. The outer wall of the diffuser is straight.

The LES mesh for the CDP flow solver consists of 500,000 control volumes and is concentrated near the walls. The cell size near the wall is approximately $y^{+}=30$, and while recent studies show that this resolution may not be enough to characterize a diffuser flow [37], we consider it sufficient for the purpose of the demonstration of the RANS-LES approach.

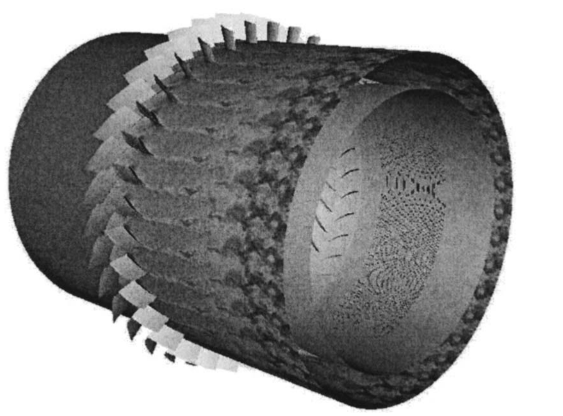


Fig. 11 Integrated RANS-LES of compressor/prediffuser: velocity distribution at the 50% plane

LES inflow boundary conditions were defined corresponding to Eq. (1). The turbulence database needed for this inflow boundary condition was created by a periodic annular pipe flow using the same mean flow characteristics as estimated for this flow configuration.

The load balancing between the two flow solvers has to be done manually. A variety of factors play a role in the efficient allocation of available resources to the two flow solvers. These factors are the number of control volumes in each domain, the ratio of time steps (here: $\Delta t_{\mathrm{RANS}}/\Delta t_{\mathrm{LES}}=7$), the convergence speeds, the numerical methods, parallel efficiencies, and partitioning limits. The variety of factors requires a practical approach to address the load balancing. Here, we performed simulations of the separate domains and assessed the computational needs for each flow solver. As it turned out, for the current case an equal amount of computational resources were necessary for both domains. This allows one to minimize the idle times at the synchronization points.

6.2 Results. The computations using the unstructured LES flow solver CDP and the multiblock structured RANS solver TFLO were carried out using 64 processors for TFLO and 64 processors for CDP. Here, eight blade passings were computed in 60 h of wall-clock time using an IBM Power3 system.

The actual Mach number at the interface was Ma=0.1 ensuring the validity of the low-Mach-number approximation in the LES domain. With these flow conditions, the Reynolds number in the compressor, based on the inlet velocity and the chord length, is approximately Re=60,000. The mass flux over the interface was conserved with an error of ≈0.5%.

Since no experimental data are available for this test case, the interpretation of the results has to be done qualitatively. Figures 11 and 12 show the axial velocity distributions at 50% of the span of the compressor blades for an instantaneous snapshot of the computation. The upstream RANS solution corresponds to a phase-averaged solution, whereas the downstream LES solution is truly unsteady.

The wakes of the stators can clearly be identified in the RANS domain downstream of the stators. The communication of the flow solvers at the interface ensures that the full three-dimensional (3D) flow features are transferred from the upstream flow solver to the downstream domain. The boundary conditions of the LES flow solver are defined according to these data. Hence, the wake of the stator correctly propagates across the interface and can still be found far downstream in the diffuser. It can also be seen that the turbulence, which is resolved in the LES domain, creates more evenly distributed velocity profiles.

The differences in the description of turbulence are more apparent in Fig. 13, which shows the vorticity distribution at 50% of the span of the stator. Here the magnitude of the vorticity is depicted

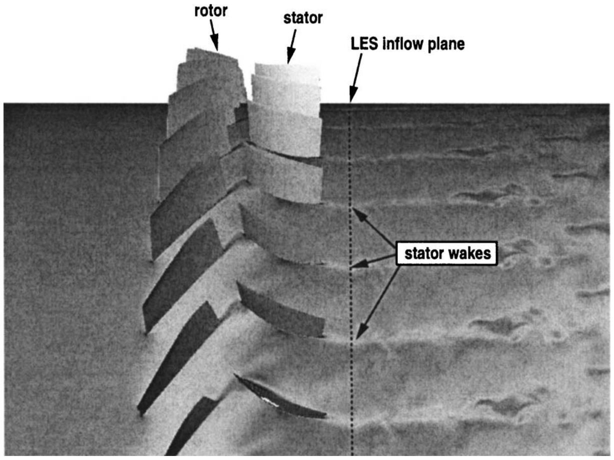


Fig. 12 Integrated RANS-LES of compressor/prediffuser: velocity distribution at the 50% plane. Close-up of the interface.

and is computed according to the unsteady flow field of both domains. In the RANS domain, the vorticity is mainly created because of the mean flow features, such as the wall boundary layers, and secondary flows and vortices. The stator creates two vorticity sheets, one on the extrado and one on the intrado. Both vorticity sheets propagate downstream across the interface.

The vorticity distribution in the LES domain is characterized by small-scale turbulence. Turbulence, present in the upstream RANS domain and modeled by a RANS turbulence model, has to be regenerated. The small-scale turbulence has been reconstructed at the interface using the LES inflow boundary condition (Eq. (1)). It can be seen that the small-scale turbulence interferes with the stator wakes. The turbulent diffusion of the stator wakes in the RANS domain is modeled with an eddy viscosity model, which gives these a very smooth appearance. In the LES domain, the turbulent transport is given by the resolved turbulence, and hence vortical turbulent structures can be identified.

7 Conclusions

The current study describes an approach to combine RANS and LES flow solvers for integrated simulations. Here, for gas turbine applications, a framework has been established that allows one to simulate the multicomponent effects between the turbomachinery and the combustor. The RANS flow solver is used to compute turbomachinery portions, whereas the LES is intended for the combustor. The main motivation is to reduce the overall computational cost of this kind of simulations by using the appropriate models for each of the portions of the flow path.

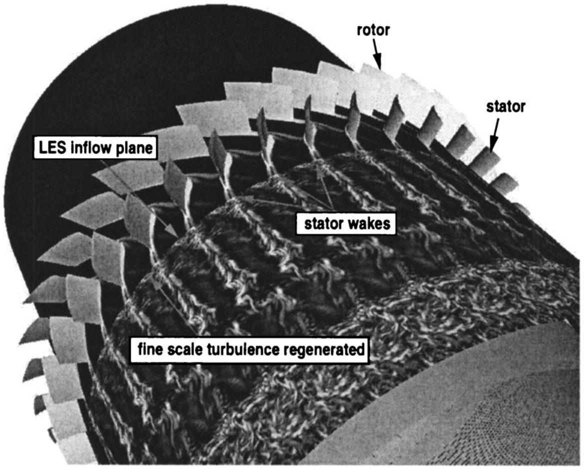


Fig. 13 Integrated RANS-LES of compressor/prediffuser: vorticity magnitude distribution at the 50% plane. Vorticity created on the surfaces of the stators can be found in the LES domain.

Part of the efforts to integrate RANS and LES flow solvers was devoted to the setup of an efficient communication pattern between the flow solvers in a parallel environment. Algorithms have been developed and implemented in several flow solvers that allow for an arbitrary number of flow solvers to be run simultaneously and exchange flow information at the interfaces of their domains.

Furthermore, it has to be ensured that the received flow information is used in meaningful boundary conditions to take the flow physics computed by the peer flow solvers into account. For this reason, appropriate boundary conditions have been developed and implemented.

The interface and boundary conditions have been validated on two test cases, one where the RANS flow solver is upstream of the LES, and one where the RANS flow solver is downstream of the LES. Both validation studies show very good results. Additionally, the computation of the coupled modified NASA stage 35/ prediffuser geometry demonstrates the concept of integrated RANS-LES computations in complex geometries.

The current study renders the integrated RANS-LES approach available to study multicomponent effects in gas turbines. Recent studies showed that the present approach can be applied to realistic gas turbine geometries with reasonable computational costs [29]. Ultimately, this approach will allow one to simulate the flow through entire gas turbines using RANS for the turbomachinery and LES for the combustor.

Acknowledgments

The support by the US Department of Energy within the ASC program is gratefully acknowledged.

References

[1] Ferziger, J. H., 1996, *New Tools in Turbulence Modelling*, Springer, New York, *Les edition physique*, Chap. 2, pp. 29–47.
[2] Sagaut, P., 2002, *Large Eddy Simulation for Incompressible Flows*, 2nd Edition, Springer, Berlin.
[3] Veynante, D., and Poinsot, T., 1996, *New Tools in Turbulence Modelling*, Les edition physique, Chap. 5, pp. 105–140.
[4] Poinsot, T., Schlüter, J., Lartigue, G., Selle, L., Krebs, W., and Hoffmann, S., 2001, "Using Large Eddy Simulations to Understand Combustion Instabilities in Gas Turbines," *IUTAM Symposium on Turbulent Mixing and Combustion*, Kingston, Canada pp. 1–8.
[5] Constantinescu, G., Mahesh, K., Apte, S., Iaccarino, G., Ham, F., and Moin, P., 2003, "A New Paradigm for Simulation of Turbulent Combustion in Realistic Gas Turbine Combustors Using LES," *ASME Turbo Expo 2003*, ASME Paper No. GT2003-38356.
[6] Trenberth, K. E., 1992, *Climate System Modeling*, Cambridge, New York, Chap. 9.
[7] Adamidis, P., Beck, A., Becker-Lemgau, U., Ding, Y., Franzke, M., Holthoff, H., Laux, M., Müller, A., Münch, M., Reuter, A., Steckel, B., and Tilch, R., 1998, "Steel Strip Production—A Pilot Application for Coupled Simulation With Several Calculation Systems," J. Mater. Process. Technol., **80–81**, pp. 330–336.
[8] Spalart, P. R., 2000, "Trends in Turbulence Treatments," *Fluids 2000*, June, Denver, AIAA Paper No. 2000-2306.
[9] Batten, P., Goldberg, U., and Chakravarthy, S., 2002, "LNS—An Approach Towards Embedded LES," *40th AIAA Aerospace Sciences Meeting and Exhibit*, Jan., Reno, AIAA Paper No. AIAA 2002-0427.
[10] Doi, H., and Alonso, J. J., 2002, "Fluid/Structure Couples aeroelastic computations for Transsonic flows in Turbomachinery," *ASME Turbo Expo 2002*, ASME Paper No. GT-2002-30313.
[11] Yao, J., Jameson, A., Alonso, J. J., and Liu, F., 2000, "Development and Validation of a Massively Parallel Flow Solver for Turbomachinery Flows," AIAA Paper No. 00-0882.
[12] Wilcox, D. C., 1998, *Turbulence Modeling in CFD*, 2nd Edition, DCW Industries, La Canada, CA.
[13] Jameson, A., 1991, "Time Dependent Calculations Using Multigrid, With Applications to Unsteady Flows Past Airfoils and Wings," *AIAA 10th Computational Fluid Dynamics Conference*, Honolulu, AIAA Paper No. 91-1596.
[14] Alonso, J. J., Martinelli, L., and Jameson, A., 1995, "Multigrid Unsteady Navier-Stokes Calculations With Aeroelastic Applications," *AIAA Paper, AIAA 33rd Aerospace Sciences Meeting and Exhibit*, Reno, AIAA Paper No. 95-0048.

[15] Belov, A., Martinelli, L., and Jameson, A., 1996, "Three-Dimensional Computations of Time-Dependent Incompressible Flows With an Implicit Multigrid-Driven Algorithm on Parallel Computers," *Proc. of 15th Int. Conference on Numerical Methods in Fluid Dynamics, Monterey, CA*, Springer-Verlag, Berlin, Lecture Notes in Physics, No. 490.

[16] Smagorinsky, J., 1963, "General Circulation Experiments With the Primitive Equations, I. The Basic Experiment," Mon. Weather Rev., **91**(3), pp. 99–152.

[17] Germano, M., Piomelli, U., Moin, P., and Cabot, W., 1991, "A Dynamic Subgrid-Scale Eddy Viscosity Model," Phys. Fluids A, **3**(7), pp. 1760–1765.

[18] Pierce, C., and Moin, P., 1998, "Large Eddy Simulation of a Confined Coaxial Jet With Swirl and Heat Release," AIAA Paper No. 98-2892, June.

[19] Akselvoll, K., and Moin, P., 1996, "Large-Eddy Simulation of Turbulent Confined Coannular Jets," J. Fluid Mech., **315**, pp. 387–411.

[20] Pierce, C. D., 2001, "Progress-Variable Approach for Large-Eddy Simulation of Turbulent Combustion," Ph.D. thesis, Stanford University, Stanford, June.

[21] Moin, P., Squires, K., Cabot, W., and Lee, S., 1991, "A Dynamic Subgrid-Scale Model for Compressible Turbulence and Scalar Transport," Phys. Fluids A, **3**(11), pp. 2746–2757.

[22] Moin, P., and Apte, S., 2004, "Large-Eddy Simulation of Realistic Gas Turbine Combustors," AIAA Paper No. 2004-0330, Jan.

[23] Mahesh, K., Constantinescu, G., and Moin, P., 2004, "A Numerical Method for Large-Eddy Simulation in Complex Geometries," J. Comput. Phys., **197**(1), pp. 215–240.

[24] Schlüter, J. U., Shankaran, S., Kim, S., Pitsch, H., Alonso, J. J., and Moin, P., 2003, "Towards Multi-Component Analysis of Gas Turbines by CFD: Integration of RANS and LES Flow Solvers," *ASME Turbo Expo 2003*, June 16–19, Atlanta, ASME Paper No. GT2003-38350.

[25] Schlüter, J. U., Pitsch, H., and Moin, P., 2004, "Large Eddy Simulation In-Flow Conditions for Coupling With Reynolds-Averaged Flow Solvers," AIAA J., **42**(3), pp. 478–484.

[26] Schlüter, J. U., Pitsch, H., and Moin, P., 2004, "Outflow Conditions for Integrated Large Eddy Simulation/Reynolds-Averaged Navier-Stokes Simulations," AIAA J., **43**(1), pp. 156–164.

[27] Schlüter, J. U., Wu, X., Kim, S., Alonso, J. J., and Pitsch, H., 2004, "Coupled RANS-LES Computation of a Compressor and Combustor in a Gas Turbine Engine," *40th AIAA/ASME/SAE/ASEE Joint Propulsion Conference and Exhibit*, July, AIAA Paper No. 2004-3417.

[28] Poinsot, T. J., and Lele, S. K., 1992, "Boundary Conditions for Direct Simulations of Compressible Viscous Reacting Flows," J. Comput. Phys., **101**, pp. 104–129.

[29] Dellenback, P. A., Metzger, D. E., and Neitzel, G. P., 1988, "Measurements in Turbulent Swirling Flow Through an Aprupt Axisymmetric Expansion," AIAA J., **26**(6), pp. 669–681.

[30] Dellenback, P. A., 1986, "Heat Transfer and Velocity Measurements in Turbulent Swirling Flows Through an Aprupt Axisymmetric Expansion," Ph.D. thesis, Arizona State University, Dec.

[31] Schlüter, J. U., Wu, X., Kim, S., Alonso, J. J., and Pitsch, H., 2004, "Integrated RANS-LES Computations of Gas Turbines: Compressor-Diffuser," Jan. AIAA Paper No. 2004-0369.

[32] Barker, A. G., and Carrotte, J. F., 2001, "Influence of Compressor Exit Conditions on Combustor Annular Diffusers, Part 1: Diffuser Performance," J. Propul. Power, **17**(3), pp. 678–686.

[33] Barker, A. G., and Carrotte, J. F., 2001, "Influence of Compressor Exit Conditions on Combustor Annular Diffusers, Part 2: Flow Redistribution," J. Propul. Power, **17**(3), pp. 687–694.

[34] Davis, R., Yao, J., Clark, J. P., Stetson, G., Alonso, J. J., Jameson, A., Haldeman, C., and Dunn, M., 2002, "Unsteady Interaction Between a Transsonic Turbine Stage and Downstream Components," *ASME Turbo Expo 2002*, ASME Paper No. GT-2002-30364.

[35] Davis, R., Yao, J., Alonso, J. J., Paolillo, R., and Sharma, O. P., 2003, "Prediction of Main/Secondary-Air System Flow Interaction in a High Pressure Turbine," AIAA Paper No. AIAA-2003-4833.

[36] Ham, F., Apte, S., Iaccarino, G., Wu, X., Herrmann, M., Constantinescu, G., Mahesh, K., and Moin, P., 2004, "Unstructured LES of Reacting Multiphase Flows in Realistic Gas Turbine Combustors," *CTR Annual Research Briefs*, Center for Turbulence Research, Stanford, pp. 139–160.

[37] Wu, X., Schlüter, J. U., Moin, P., Pitsch, H., Iaccarino, G., and Ham, F., "Identification of an Internal Layer in a Diffuser," Stanford University/NASA Ames, Center for Turbulence Research, Annual Research Briefs 2004, pp. 169–182.