Proceedings of ASME TURBO EXPO 2003
2003 ASME TURBO EXPO 2003 Land, Sea and Air
June 16-19, 2003, Atlanta USA

GT2003-38350

TOWARDS MULTI-COMPONENT ANALYSIS OF GAS TURBINES BY CFD: INTEGRATION OF RANS AND LES FLOW SOLVERS

Jörg Schlüter
Heinz Pitsch
Parviz Moin

Center for Turbulence Research
Building 500
Stanford, CA, 94305-3030, USA

Sriram Shankaran
Sangho Kim
Juan Alonso

Aerospace Computing Lab
Durand Building
Stanford, CA, 94305-3030, USA

NOMENCLATURE

Variable names:

x_i cartesian coordinates ($i = 1,2,3$)
u_i velocity components ($i = 1,2,3$) in cartesian coordinates
U_x axial velocity component in cylindrical coodinates
U_ϕ azimuthal velocity component in cylindrical coodinates
ρ density
k turbulent kinetic energy
ν_t turbulent eddy viscosity
t time using LES time-step Δt
τ time using RANS time-step $\Delta\tau$
τ_F body force time constant

Subscripts:

RANS RANS flow solver variables
LES LES flow solver variables
DB data base variables

ABSTRACT

The numerical prediction of the entire aero-thermal flow through an entire gas turbine is currently limited by its high computational costs. The approach presented here intends to use several specialized flow solvers based on the Reynolds-averaged Navier-Stokes equations (RANS) as well as Large Eddy Simulations (LES) running simultaneously and exchanging information at the interfaces. This study documents the development of the interface and proves its accuracy and efficiency on simple test-cases.

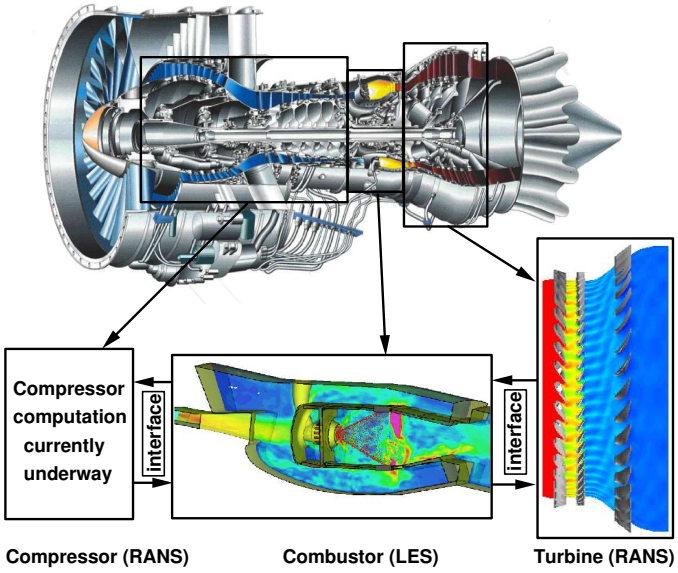


Figure 1. FLOW COMPUTATIONS IN A GAS TURBINE: COMPRESSOR ANS TURBINE [1] WITH RANS, COMBUSTOR WITH LES [2]. SO FAR, ALL COMPUTATIONS SEPARATE. INTERFACE WILL ALLOW FOR COUPLED COMPUTATIONS.

INTRODUCTION

Motivation

In the design process of a gas turbine, the usage of Computational Fluid Dynamics (CFD) is currently limited to ad-

dress design problems of single components, such as the compressor, the combustor, *or* the turbine. The optimization the entire system eludes currently CFD due to the high computational costs. Proper predictions of multi-component phenomena, such as compressor/combustor instabilities and combustor/turbine hot-streak migration, can not be presently performed.

So far, the computation of the entire flow path through a gas turbine, with all compressor and turbine stages and the reacting flow field in the combustor, is beyond the capabilities of existing flow solvers. The reason for that is, that the flow problems encountered on the way through the flow path of the gas turbine, vary strongly from one component to the other. In order to predict these flow problems efficiently, currently the flow solvers are adapted to certain flow conditions in a single component.

For instance, in the compressor and the turbine section, the flow solver has to be able to handle the moving blades, model the turbulence in wall bounded flows, and predict the pressure and density distribution properly. This can be done efficiently by a flow solver based on the Reynolds-Averaged Navier-Stokes (RANS) approach. Using high-speed computers, it is currently possible to compute entire turbine sections [1] and the computation of entire comressor sections is matter of current investigations.

On the other hand, the flow in the combustion chamber is governed by large scale turbulence, chemical reactions, and the presence of fuel spray. Experience shows that predictions of such phenomena are strongly improved by the use of Large Eddy Simulations (LES). Current LES flow solvers are able to predict cold flows accurately [2, 3] and are on the verge to compute reacting flow fields in a gas-turbine combustor [4, 5]

However, not a single flow solver exists, which is able to compute the entire flow through the gas turbine efficiently *and* accurately. A possible solution out of this dilemma is to use multiple flow solver, each appropriate to handle the flow problems encountered in its own domain. The here proposed strategy calls for a a RANS flow solver for the compressor sections, an LES flow solver for the combustor, and again a RANS flow solver for the turbine section (Fig. 1). In order to capture multi-component effects, these flow solvers have to run simultaneously and exchange information at the interfaces. With this strategy, it should be possible to render CFD available for the design process of the entire gas turbine.

The usage of entirely separate flow solvers allows, for a given flow problem, to choose the best combination of a variety of existing flow solvers, which have been developed, optimized, and validated separately. Once these have been equipped with a generic interface, it is possible to continue the development of the flow solvers separately without compromising compatibility. The implementation of this interface into several flow solvers allows their modular exchange, which results in a high degree of flexibility.

The current investigation describes the development of an interface, which is able to couple different flow solvers. Ultimately, this will lead to multi-component computations of a gas turbine, but the test-cases reported here are limited to simple flow configurations in order to prove the feasibility and advantages of coupled computations. Due to the variety of challenges encountered in the coupling process, simple and robust approaches were chosen to prove the concept and refinements left for later work.

Challenges

The implementation of an interface for the simultaneous flow computation using separate flow solvers faces a number of challenges. These can be described as follows:

1. *Establishing a contact between the flow solvers for information exchange:* The first obvious obstacle is to establish a real-time connection between two or more simultaneous running flow solvers over which the information can be exchanged. Most flow solvers are already parallelized using MPI (Message Passing Interface). Here, MPI will be used for peer-to-peer message passing as well.
2. *Ensuring that each flow solver obtains the information needed on the boundaries:* A general procedure has to be defined, which handles the logistics of information fluxes and ensures, that each flow solver knows, what information has to be sent where.
3. *Processing of the obtained information to boundary conditions:* Finally, the physical problem of defining meaningful boundary conditions from the obtained data has to be addressed. This can be especially challenging, when two different modeling approaches, such as LES and RANS, are used.

The current investigation deals with these tasks and describes a successful coupling of RANS and LES flow solvers.

PEER-TO-PEER MESSAGE PASSING

The message passing 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 a contact between two flow solvers, it has to be ensured, that the commands for the internal message passing due to the parallelization of the two codes do not interfere with each other. With MPI it is possible to define the range of the message passing with communicators. The most commonly used communicator of MPI is the standard communicator `MPI_COMM_WORLD` which includes all processors of all codes. Using this communicator for internal message passing will inevitably result in confusion between the two codes.

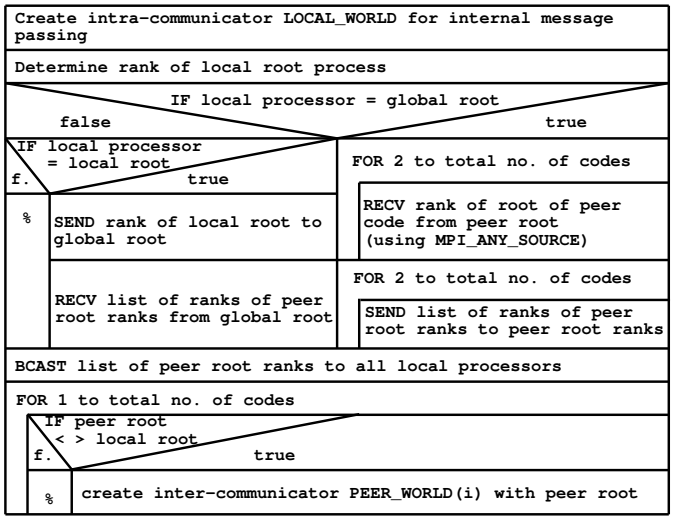


Figure 2. STRUCTURE CHART: EXCHANGE OF ROOT RANKS NEEDED FOR CREATION OF INTER-COMMUNICATORS

Hence, each code has to create its own local communicator (*intra*-communicator) to encapsulate the internal message passing. All codes have to use their own intra-communicator for all MPI commands concerning the parallelization of the code instead of `MPI_COMM_WORLD`.

In the next step, a communicator is created for the peer-to-peer message passing (*inter*-communicator). Say, a case with three flow solvers is to be run with a first 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 RANS code using three processors (ranks 6, 7, and 8, local root process 6). In order to create the inter-communicator, 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) which 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 structure chart for this procedure is shown in Fig. 2. Since there is no inter-communicator available yet, this communication has to be done with the standard communicator `MPI_COMM_WORLD`. With the knowledge of the ranks of all root processes it is possible to create the inter-communicators.

HANDSHAKE AND COMMUNICATION

Handshake

Efficient parallelizing of a flow solver seeks to limit the information exchange between parallel processes to a minimum, since the information exchange requires a large amount of time compared to the actual computation. Similarly, it is favorable to minimize the communication between several parallelly running flow solvers. 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 communication efforts by an initial handshake, which optimizes the communication during the actual flow computation.

The most simple way to organize the information exchange would be to let only the root processes communicate. However, this would mean that prior to the peer-to-peer communication the root processes would have to gather the flow information to hand over from their own processes, and after the peer-to-peer communication would have to broadcast the obtained information back to their processes. The solution reported here avoids this additional communication by direct communication of the neighboring processors on the interface.

The initial handshake routine establishes the direct communication (Fig. 3). First, for each code each processor has to identify all the points, which need flow information from the peers to define its interface boundary condition. The location of each of these points has to be stored in a data structure containing three integers and three double precisions. 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 real 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 communicates the number of points in its own domain requesting flow data to each processor of a peer code. This allows each code to dynamically allocate arrays to store received information. In the second step each processor receives a data package containing the location of the requested points from each peer processor that request a non-zero number of points.

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

In the third communication step each processor communicates to all peer-processes requesting data the number of points found. This allows again to dynamically allocate arrays for the following fourth 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 ip and id of the point. Finally, each processor determines whether all of its requested point can be served by peer processors.

Communication

The communication of flow data between iterations is rather straight forward once the 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

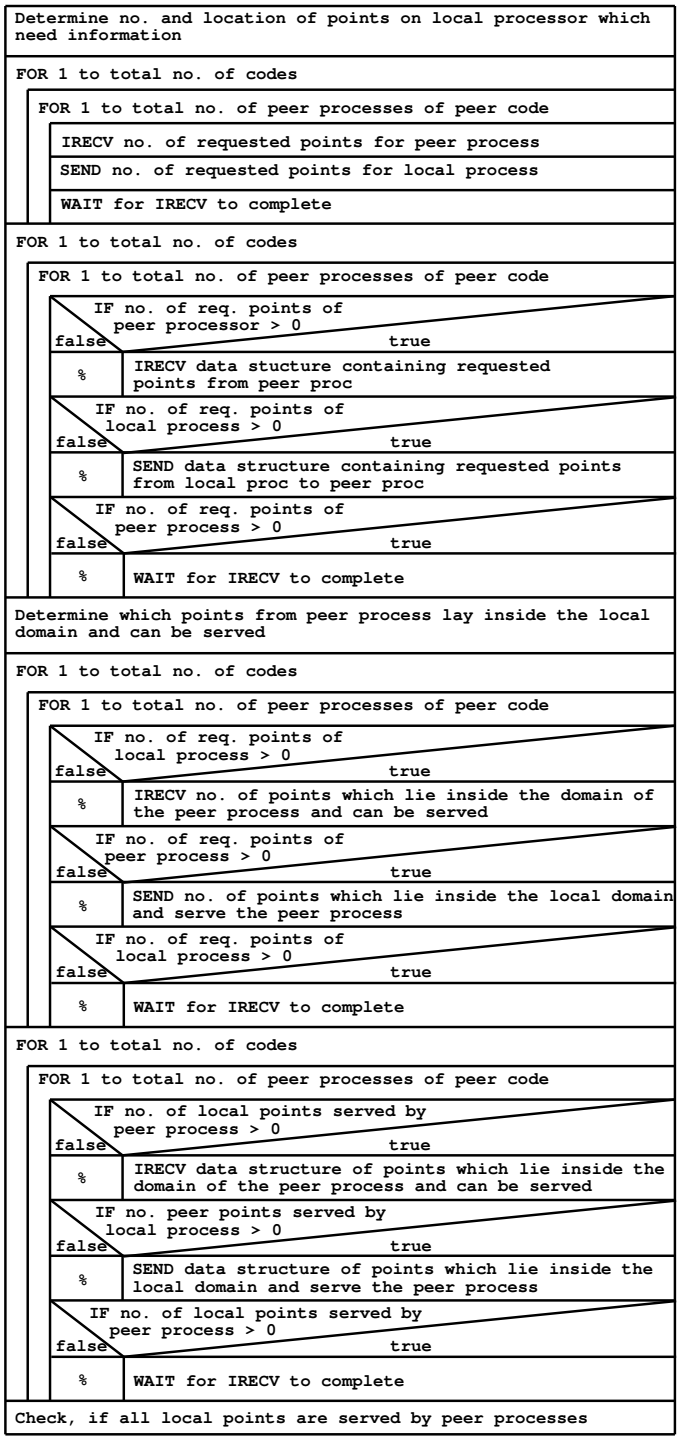


Figure 3. STRUCTURE CHART: INITIAL HANDSHAKE TO ESTABLISH DIRECT COMMUNICATION BETWEEN INTERFACE PROCESSORS

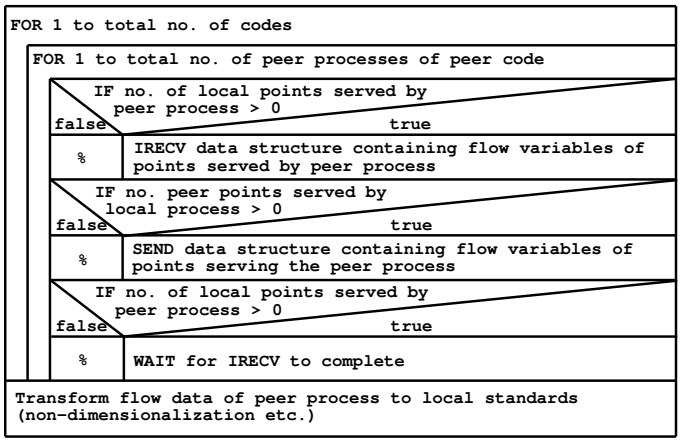


Figure 4. STRUCTURE CHART: COMMUNICATION OF FLOW DATA DURING FLOW COMPUTATIONS

flow data is expected, the data packages can be sent directly without going through the root process.

Each processor has to compile the data to be sent into a data structure. This data structure may vary between different flow solvers and has to be defined beforehand. However, as a standard data structure a set of 7 variables has been established. These variables contain $\rho, \rho u_1, \rho u_2, \rho u_3, \rho E, k$, and ν_t in this order, with ρ being the density, u_1, u_2, u_3 the velocity components in $x-, y-, z-$direction, respectively, E the total Energy, k the turbulent kinetic energy, and ν_t the eddy viscosity. This set of variables gives 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 $k-\varepsilon$ and $k-\omega$ turbulence models likewise. More sophisticated data sets are possible steered by the data structure sent in the handshake routine.

BOUNDARY CONDITIONS

LES Boundary Conditions

The formulation of unsteady LES boundary conditions from ensemble-averaged RANS data is one of the biggest challenges in the coupling of two flow solvers based on such different mathematical approaches like LES and RANS. Unsteady LES boundary conditions have to be generated which fulfill the statistical properties of the time-averaged solution delivered by the RANS flow solver. Even if an unsteady RANS computation is assumed, the time-step of the unsteady RANS computation is usually larger than the LES time-step by several orders of magnitude. The LES boundary conditions then have to correspond to the ensemble-averages delivered by the RANS computation.

LES Inflow Boundary Conditions Specifying inflow conditions for LES from upstream RANS data is a similar prob-

lem as specifying LES inflow conditions from experimental data, which is usually given in time-averaged form, and has therefore been investigated in some detail in the past. A method that has been successfully applied is to generate a time-dependent database for the inflow velocity fields by performing a separate LES simulation of a periodic pipe flow, in which virtual body forces are applied to achieve the required time-averaged solution [6]. However, unsteady RANS flow solvers may deliver *unsteady* ensemble-averaged velocity profiles. The generation of such a data-base is impossible, since the mean velocity field at the inlet is unknown prior to the LES computation.

The LES inlet conditions proposed here use a data-base created by a separate LES computation and then modifies its statistical properties in order to match the RANS solution:

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

with RANS denoting the solution delivered by the RANS computation and DB properties delivered by the database. Term *II* computes the velocity fluctuation of the database, while term *III* scales the fluctuation to the actual value needed. When added to term *I* a meaningful unsteady inlet condition is recovered. In order to keep corrections small, the generated inflow data-base should have statistical properties close to the actual prediction by the RANS flow solver, although it has been shown, that even very generic data-bases are able to recover meaningful LES inflow conditions [7, 8].

LES Outflow Boundary Conditions In order to take upstream effects of the downstream flow development into account, LES outflow conditions have to be defined that can impose mean flow properties on the unsteady LES solution matching the statistical properties delivered by a downstream RANS computation. A method, that has been used in the past, employs virtual body forces in the momentum equation to drive the mean velocity field of the LES solution to a RANS target velocity field [9, 10]. The body force F is given by:

$$F_i(x) = \frac{1}{\tau_{\mathrm{F}}} \left(\bar{u}_{i,\mathrm{RANS}}(x) - \bar{u}_{i,\mathrm{LES}}(x) \right), \qquad (2)$$

with $\bar{u}_{i,\mathrm{RANS}}$ being the solution of the RANS flow solver computed in an overlap region between LES and RANS domain, and $\bar{u}_{i,\mathrm{LES}}$ is a time-average of the LES solution over a trailing time-window. 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 an overlap region computed by a RANS simulation of the downstream region. This makes it possible to take upstream effects of the downstream flow into account.

RANS Boundary Conditions

The specification of RANS boundary conditions from LES data is essentially straight forward. The unsteady LES flow data is time-averaged over the time-step applied 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 quasi-incompressible low-Mach-number formulation of the LES code posed a challenge. While the RANS code allows for acoustic waves to propagate in the limits of the RANS formulation and its turbulence models, the density field of the LES solution is entirely defined by chemical reactions and not by acoustics. This leads to the necessity of the RANS inflow and outflow condition to be able to fluctuate the density field at the boundaries in order to let acoustic waves leave the domain.

Currently, the mass-flux vector at every point of the inlet is being specified corresponding to the value delivered by the LES computation. This means ρu_i (and T) are imposed at the boundaries. This allows the density ρ to fluctuate to account for passing acoustic waves. The velocity components u_i are adjusted accordingly in order to conserve the mass-flux. Variations of ρ are in the order of $< 2\%$.

Other boundary conditions are possible, especially Navier-Stokes characteristic boundary conditions [11], which provide a more accurate treatment of acoustic waves.

VALIDATION OF THE INTERFACE

In order to validate the interface and the boundary conditions, an LES flow solver and a RANS flow solver were equipped with the interface and the newly developed boundary conditions. Integrated flow computations were performed in an LES-LES and an LES-RANS environment.

The LES Flow Solver

The LES flow solver chosen for this work, is an code developed at the Center for Turbulence Research (CTR) at Stanford by Pierce and Moin [12]. The flow solver solves the filtered momentum equations with a low-Mach number assumption on an axi-symmetric structured single-block mesh. A second-order finite-volume scheme on a staggered grid is used [13]. The subgrid stresses are approximated with an eddy-viscosity approach, where the eddy viscosity is determined by a dynamic procedure [14, 15].

The RANS Flow Solver

The RANS flow solver used for this investigation is the TFLO code developed at 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 multi-block meshes [16].

The solution procedure is based on efficient explicit modified Runge-Kutta methods with several convergence acceleration techniques such as multi-grid, residual averaging, and local time-stepping. These techniques, multi-grid in particular, provide excellent numerical convergence and fast solution turnaround. Turbulent viscosity is computed from a $k-\omega$ two-equation turbulence model. The dual-time stepping technique [17–19] is used for time-accurate simulations that account for the relative motion of moving parts as well as other sources of flow unsteadiness.

Numerical Experiment: Swirl Flow

The computation of a swirl flow presents a challenging testcase in order to validate the interface and the boundary conditions, due to the complexity of the flow and its sensitivity to inflow and outflow parameters. Yet, this test case is simple enough to perform an LES computation of the entire domain in order to obtain an 'exact' solution, which serves as a reference solution to assess the accuracy of integrated computations.

A swirl flow at an expansion with a subsequent contraction three diameters D downstream of the expansion is considered (Fig. 5a). Inlet velocity profiles are taken from an actual experiment in a similar geometry [20]. 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 extension and strength of this recirculation zone is strongly influenced by the presence of the downstream contraction.

In a first computation the entire domain is computed by LES. All subsequent computations assume, that this domain is to be computed by two or more separate flow solvers. The geometry is split in two computational domains with a short overlap region. The expansion is to be computed with the LES code, while the contraction is computed either by a second instance of the LES code or by the RANS code (Fig. 5c). 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.

Integrated LES/LES Computations

The first test for the interface and the LES boundary conditions is to use an LES flow solver for the second part of the domain. Hence, in these integrated LES/LES computations the same LES flow solver is used twice. The time-interval can be chosen arbitrarily when communication between the two instances of the flow solver takes place. Each LES computation can chose a time-step to advance the solution between two iterations of its own, only limited by the CFL condition in its

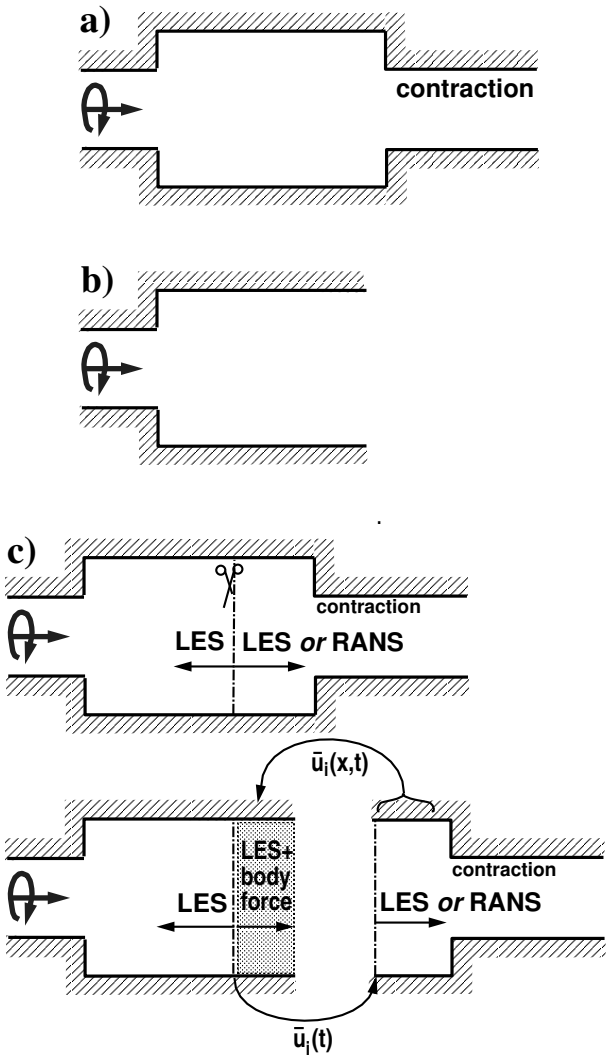


Figure 5. GEOMETRY FOR INTERFACE VALIDATION: a) FULL GEOMETRY, b) REDUCED LES DOMAIN, c) SCHEMATIC SPLITTING OF DOMAIN INTO TWO COMPUTATIONAL DOMAINS

own domain. After several iterations, after both LES computations have computed the same physical time-span, an exchange of time-averaged quantities, the mean velocities $\bar{u}_i$ and the turbulent kinetic energy k, takes place. While it would have been possible in the case of two LES computations to exchange more information, especially about turbulent quantities, it was aim to proof the validity of the LES boundary conditions defined earlier using RANS-type information from the respective other solver.

Figure 6 shows the velocity profiles for three different computations. The velocity profiles denoted by the circles represent the LES computation of the entire domain (Fig. 5a), and hence, the target for the integrated computations.

To show the importance of integrated computations for this case, the dashed lines show the velocity profiles of an LES com-

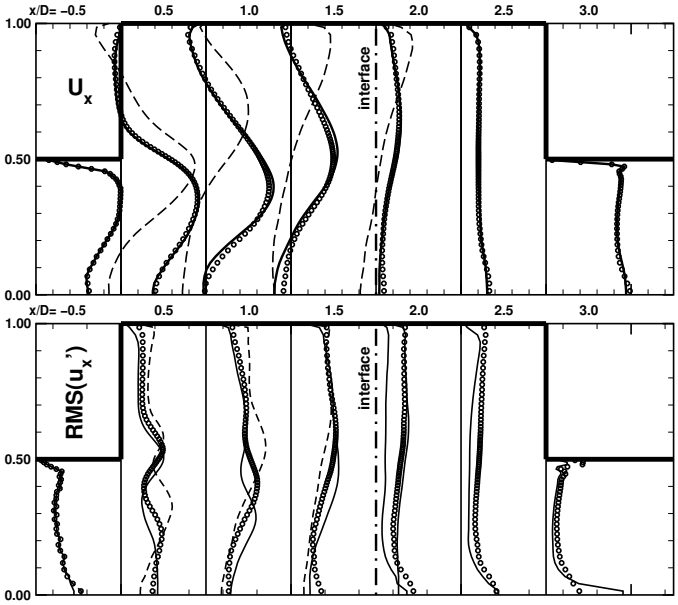


Figure 6. INTEGRATED LES/LES COMPUTATIONS: VELOCITY COMPONENTS FOR SEVERAL DOWNSTREAM LOCATIONS. CIRCLES: LES OF FULL GEOMETRY (Fig. 5a), DASHED LINE: LES OF THE EXPANSION W/O CONTRACTION (Fig. 5b), SOLID LINE: INTEGRATED LES/LES COMPUTATION (Fig. 5c).

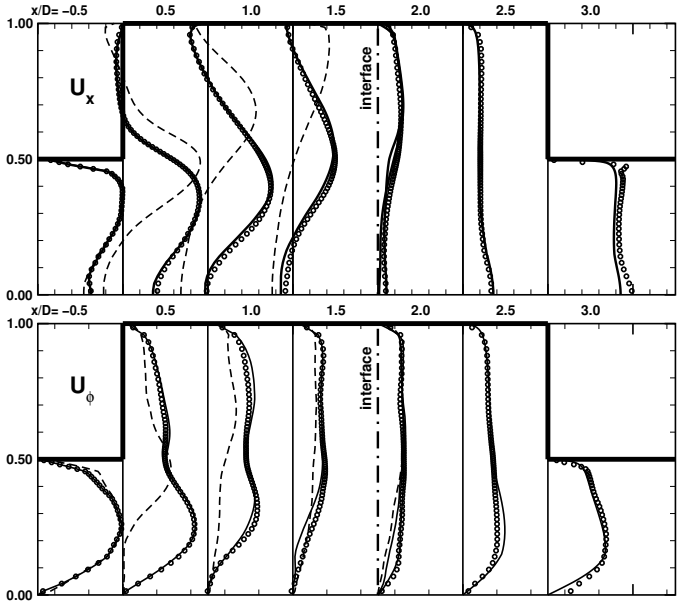


Figure 7. INTEGRATED LES/RANS COMPUTATIONS: VELOCITY COMPONENTS FOR SEVERAL DOWNSTREAM LOCATIONS. CIRCLES: LES OF FULL GEOMETRY (Fig. 5a), DASHED LINE: LES OF THE EXPANSION W/O CONTRACTION (Fig. 5b), SOLID LINE: INTEGRATED LES/LES COMPUTATION (Fig. 5c).

putation of the expansion without the computation of the contraction by a second flow solver (Fig. 5b). It can be seen that the obtained velocity field differs substantially from the first simulation, and hence, the influence of the downstream contraction can not be neglected.

The solid lines in Fig. 6 show the integrated LES/LES-computation using two LES solvers for the two domains (Fig. 5c). The location of the interface is denoted with a dot-dashed line. The velocity profiles on the left-hand side of the interface are computed with the first LES computation and the profiles on the right hand-side are from the second LES. The LES 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/LES-computation tend towards the velocity profiles of the LES of the entire domain. The inlet conditions of the second LES are defined from the mean velocity profiles obtained from the upstream LES.

In the integrated LES/LES computation, the velocity fluctuations $u_i'^2$ are handed over as the turbulent kinetic energy $k = 0.5 \cdot (u_1'^2 + u_2'^2 + u_3'^2)$, and reconstructed as $u_i'^2 = 2 \cdot k/3$. This explains a mismatch in the axial velocity fluctuations at the interface. Although it would have been possible in integrated LES/LES computations to hand over the entire Reynolds-stress tensor, the use of the RANS standard data set allows better comparison with the following computations.

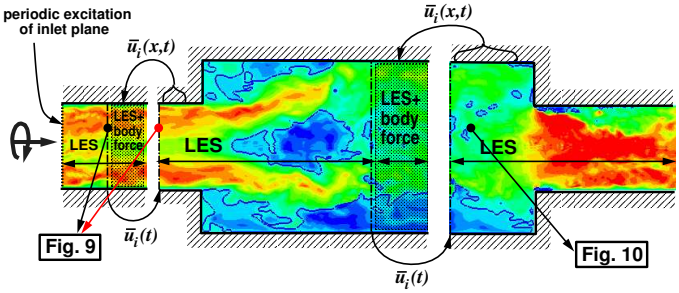
Integrated LES/RANS Computations

The final step in assessing integrated flow computations is to perform a simulation, where the second LES flow solver is replaced by a RANS flow solver. The swirl flow at the expansion is computed by the LES flow solver while the contraction is computed with the RANS flow solver TFLO.

Figure 7 shows the mean velocity profiles obtained by an integrated LES-RANS computation. The circles show the LES of the entire domain, and, for comparison, the dashed line represents the LES solution of the swirl flow without the computation of the contraction.

The integrated LES-RANS computation (solid lines) essentially matches the velocity profiles from the LES of the entire domain. This means that integrated LES-RANS computations are able to successfully predict complex flows such as the swirl flow considered here.

The time advantage of integrated LES-RANS computations is strongly dependent on the chosen RANS time-step. For the present case, the RANS time step was chosen approximately $2 \cdot 10^3$ times longer than the LES time-step limited by the CFL condition. This resulted in a decrease of computational costs by a factor of ≈ 2.



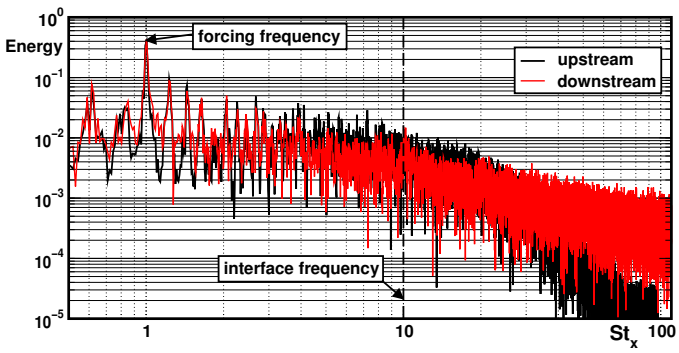


Figure 9. ENERGY SPECTRA AT THE INTERFACE, ENERGY NORMALIZED TO 0HZ, BOTH SPECTRA AT THE SAME PHYSICAL LOCATION, ONE COMPUTED BY UPSTREAM LES, ONE BY DOWNSTREAM

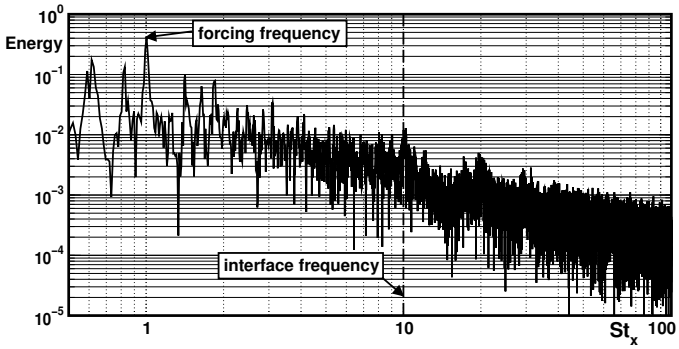


Figure 10. ENERGY SPECTRA BEHIND THE SECOND INTERFACE.

PROPAGATION OF DYNAMIC PROPERTIES

Finally, the capability of the interface to propagate dynamic perturbations, such as blade passing frequencies, to other components of the system has to be assessed.

The previous test-case was made more challenging by adding an additional LES domain upstream of the expansion (Fig. 8). The inlet plane of this LES domain was periodically excited in order to simulate a periodic perturbation, such as the passing of a compressor blade.

Figure 9 tracks the periodic perturbation over the first interface. It can be seen, that the energy spectra of the turbulence show the same long wave spectrum on both sides of the interface. Most notably, the forcing frequency at the Strouhal number $St_D = 1.0$ (with $St = f \cdot D_{ref}/U_{ref}$) is transmitted correctly in frequency and amplitude. Both spectra show great similarities in a frequency range 0Hz$< f < 0.5 * F_{\text{interface}}$ with $F_{\text{interface}}$ being the frequency of the information exchange between the flow solvers. The high frequency spectrum is not resolved by the information exchange and hence, differs on both sides of the interface. Fig. 10 shows that the forcing frequency is able to pass a second interface downstream.

Future work will investigate the proper transfer of the phase spectrum [21] over the interface. This property is important in order to determine accurately the phase lag of reflected waves and hence, instabilities due to interaction of different components.

CONCLUSIONS

The complexity of flow problems encountered in the flow path of a gas turbine calls for a split of the work load onto several flow solvers in order to decrease the computational costs while maintaining the accuracy. So far, most flow solvers are developed to work alone. In this study, an interface was developed and implemented that enables two or more flow solvers to run simultaneously and to exchange data at the overlapping boundaries.

In order to show the advantages and to prove the feasibility of the interface, it was tested on a simple test case. A swirl flow at an expansion with a subsequent contraction was computed, which has been split into two parts, the upstream expansion and the downstream contraction part. Each of these domains is computed by a separate flow solver in a fully coupled simulation. The integrated LES-LES and LES-RANS computations have demonstrated to yield the same flow prediction as an LES computation of the entire domain.

The LES boundary conditions developed in earlier work were put to a real-time test. RANS boundary conditions were adapted to accommodate the different approaches (compressible/low-Mach number) on both sides of the interface.

The ability of the interface to transmit unsteady information, such as a passing a predominant frequency in an upstream domain towards a downstream domain has been demonstrated. This ability is of importance in integrated gas turbine computations, where blade passing frequencies may influence the flow field in other components of the gas turbine.

The computation reported in this work proof the feasibility, accuracy and efficiency of integrated LES/RANS computations. LES and RANS flow solvers were successfully combined in order to improve the efficiency of the flow prediction without compromising the accuracy. This is an important step towards the application of this concept to industrial applications.

Future work will concentrate on turbo-machinery applications in order to proof the feasibility of this concept in an industrial framework.

ACKNOWLEDGMENT

We gratefully acknowledge support by the US Department of Energy under the ASCI program.

REFERENCES

[1] 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 (GT-2002-30364).

[2] Mahesh, K., Constantinescu, G., Apte, S., Iaccarino, G., and Moin, P., 2001. "Large-eddy simulations of gas turbine combustors". *Annual Research Briefs 2001*, pp. 3–18. Center for Turbulence Research, NASA Ames/Stanford Univ.

[3] Schlüter, J., Schönfeld, T., Poinsot, T., Krebs, W., and Hoffmann, S., 2001. "Characterization of confined swirl flows using large eddy simulations". ASME Turbo Expo 2001 (2001-GT-0060).

[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*, pp. 1–8. Kingston, Canada 2001.

[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 (GT2003-38356).

[6] Pierce, C. D., and Moin, P., 1998. "Method for generating equilibrium swirling inflow conditions". AIAA Journal, **36** (7), pp. 1325–1327.

[7] Schlüter, J., 2002. "Consistent boundary conditions for integrated LES/RANS simulations: LES inflow conditions". *CTR Annual Research Briefs*. Center for Turbulence Research, Stanford.

[8] Schlüter, J. U., Pitsch, H., and Moin, P., 2003. "Boundary conditions for LES in coupled simulations". AIAA paper (AIAA-2003-0069).

[9] Schlüter, J., and Pitsch, H., 2001. "Consistent boundary conditions for integrated LES/RANS simulations: LES outflow conditions". *CTR Annual Research Briefs*, pp. 19–130. Center for Turbulence Research, Stanford.

[10] Schlüter, J. U., Pitsch, H., and Moin, P., 2002. "Consistent boundary conditions for integrated LES/RANS simulations: LES outflow conditions". AIAA paper (2002-3121). 32nd AIAA Fluid Dynamics Conference, June 24-27, St, Louis, MO.

[11] Poinsot, T. J., and Lele, S. K., 1992. "boundary conditions for direct simulations of compressible viscous reacting flows". Journal of Computational Physics (101), pp. 104–129.

[12] Pierce, C., and Moin, P., 1998. "Large eddy simulation of a confined coaxial jet with swirl and heat release". AIAA Paper (AIAA 98-2892).

[13] Akselvoll, K., and Moin, P., 1996. "Large-eddy simulation of turbulent confined coannular jets". *Journal of Fluid Mechanics,* **315**, pp. 387–411.

[14] 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.

[15] 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.

[16] 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 (AIAA-00-0882).

[17] Jameson, A., 1991. "Time dependent calculations using multigrid, with applications to unsteady flows past airfoils and wings". AIAA paper (AIAA Paper 91-1596). AIAA 10th Computational Fluid Dynamics Conference, Honolulu, HI, June 1991.

[18] Alonso, J. J., Martinelli, L., and Jameson, A., 1995. "Multigrid unsteady Navier-Stokes calculations with aeroelastic applications". AIAA Paper (AIAA 95-0048). AIAA 33rd Aerospace Sciences Meeting and Exhibit, Reno, NV, 1995.

[19] 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". In Proceedings of the 15th International Conference on Numerical Methods in Fluid Dynamics, Monterey, CA.

[20] Dellenback, P. A., Metzger, D. E., and Neitzel, G. P., 1988. "Measurements in turbulent swirling flow through an aprupt axisymmetric expansion". AIAA Journal, **26** (6), pp. 669–681.

[21] Schlüter, J. U., 2003. "Consistent boundary conditions for integrated LES/RANS computations: LES inflow conditions". AIAA paper (AIAA-2003-3971). accepted at 16th AIAA CFD conference 2003.