Unsteady vs. steady turbomachinery flow analysis: exploiting large-scale computations to deepen our understanding of turbomachinery flows

D. Graham Holmes
Branden J. Moore
Stuart D. Connell
General Electric Global Research
Niskayuna, NY 12309

ABSTRACT

Turbomachines, from automotive turbochargers to jet engines to large steam turbines, use alternating rows of stationary and moving blades to compress (and do work on) or expand (and extract work from) a working fluid. After more than a century of development, turbomachines are highly efficient, and further improvements in efficiency are hard to achieve.

The key simplifying assumption in the analysis of the flow in turbomachines has always been that the inherently *unsteady* flow in the machine can be treated as *steady* when viewed in the rotating reference frame of each blade row. The designs of all of the most efficient machines now in service have been made using this steady flow assumption. Further increases in efficiency may now come only if we are willing to move on to unsteady flow analyses—which can be orders of magnitude more expensive than unsteady analysis—and require massive computer resources.

This paper illustrates the challenge of such calculations, via the computational analysis of the flow in the front stages of an aircraft low pressure turbine. We discuss three challenges: (1) scaling down the calculation—without losing fidelity—to fit the available computer resources; (2) scaling up a CFD flow solver to perform well on thousands of processors on a machine like Jaguar at Oak Ridge National Laboratory; and (3) mining the terabytes of solution snapshots to obtain not only some overall performance numbers, but—more important—an understanding of how the unsteady interactions between blade rows can help us design more efficient and more durable machines.

INTRODUCTION

Turbomachines are an essential part of modern life. Almost all the world's electricity production is via turbomachinery: steam turbines, gas turbines, hydraulic turbines, and, more recently, wind turbines. Air transport became what it is today because of the shift from reciprocating I.C. engines to gas turbines in the nineteen fifties.

The classic turbomachine has one or more *stages*, with each stage comprising a rotating row of blades (rotors), preceded or followed by a stationary row of blades (stators). Here the wind turbine is unusual—the stator is missing. Each stage either does work on the fluid passing through it, raising the pressure and temperature of the fluid (compressor), or extracting work (turbine). Multiple stages are frequently stacked on the same shaft. In its simplest incarnation, for example, a jet engine compresses air via a multistage compressor, burns fuel in the compressed airstream, and expands the air through a multistage turbine, with the shaft power produced by the turbine driving the compressor. Leftover energy in the air stream is used to propel the airplane.

Turbomachines in a recognizably modern form have a long history. The axial flow steam turbine, for example, first appears in the late 19th century (Parsons), and a practical axial flow compressor around 1940. The General Electric Company has been building turbomachines since the early 20th century and today is a major producer of turbomachines for electric power generation and air transport. Given their long history, it is to be expected that turbomachines have evolved to high levels of efficiency, power density and durability. This is the case, despite the fact that digital computers only appeared relatively late in the development of these devices. Improvements were—as for any engineering discipline—partly experimental and partly analytical. The analytical approach relies on a series of simplifications, such that design can proceed in phases: (1) a simple block diagram (e.g., compressor → combustor → turbine); (2) one-dimensional models of each component, in which each blade row becomes an "actuator" that changes the flow in some defined, but highly simplified, way and passes the flow to the next row; (3) simplified two-dimensional models of the flow field in a blade row (with the third dimension that is dropped being either the spanwise or circumferential direction); (4) fully three-dimensional analysis of the flow in a blade row; and (5), more recently, three-dimensional analysis of the flow in multiple blade rows. The more complete the analytical model, the greater the necessity of computation. All turbomachinery design today is based on passing through these phases

The flow in a turbomachine is inherently unsteady; it is the unsteady nature of the flow that enables a turbomachine to compress or expand a fluid without a complex system of opening and closing valves of the sort seen in an internal combustion engine. Yet the design of turbomachinery has proceeded from the beginning up to the present without considering the unsteadiness of the flow. The reason is that the analyst, in each blade row, sits in the (half the time rotating) reference frame of each blade row, and assumes the flow to be steady in that reference frame. A rotating reference frame is, of course, non-Newtonian, but that is readily taken care of via some additional acceleration terms (centripetal, Coriolis). I do not know when, or by whom, this simplification was first *explicitly* invoked, but it is ubiquitous and has proven to be very powerful. Denton [1] has recently stated that "the assumption of steady flow in multistage calculations is a source of error whose magnitude is not fully understood." Similarly, He [2] said: "It is this basic single-passage steady flow model that has dominated turbomachinery flow analysis in a design environment for the past 40 years."

The assumption of steady flow in each blade row ignores some important phenomena. Each blade in a blade row produces wakes that pass downstream into the next row. Since the downstream row is in motion relative to the upstream row, the flow in the downstream row *must* be unsteady. Similarly, a blade row in which the flow is partly supersonic may produce shock waves that propagate upstream into the preceding row, again producing an unsteady flow. In order to maintain the useful fiction of steady flow, the flow at the interfaces between blade rows can be averaged in the circumferential direction (or, equivalently, averaged in time) as data is exchanged between blade rows. The assumption of steady flow enables one further simplification, beyond removing time from the analysis. Since the flow in a blade row is steady, and since all blades in a row are identical, it is possible to confine the analysis in a blade row to a *single passage* between a pair of blades. The multiplicity of identical passages is accounted for by invoking a form of symmetry boundary condition at the interfaces between adjacent blade passages. The resulting formulation gives rise to what is termed "steady multistage" analysis. This is the standard approach to turbomachinery design today and is the paradigm referred to by Denton and He above. A great deal of research has been devoted to devising averaging schemes at the interfaces between blade rows that enable steady multistage calculations to match experimental data.

There are situations in which unsteady flow effects are known to be important and in which unsteady flow effects *must* be taken into account. One is blade flutter—the spontaneous (and possibly catastrophic) *self-excited* blade vibration induced by the flow. Another is "forced response," in which the frequency with which the upstream wakes pass through a blade row coincides with one of the fundamental vibration frequency of the blades, resulting in a resonant response, and possibly vibratory stress levels that can lead to fatigue failure. Specialized techniques exist for analyzing these cases—a frequency domain linearization of the governing equations for flutter (valid since the self-excited vibrations grow exponentially from infinitesimal amplitudes), a so-called gust response approach for forced vibration, by means of which one is still able to restrict analysis to a single blade passage in one row, although time has to be introduced. A third area in which unsteady calculations are coming into use is in the prediction of blade surface temperatures in the rotating turbine stages immediately behind the combustor of a gas turbine or jet engine. Accurate predictions of these temperatures is critical to designing blades that can simultaneously withstand the extreme stresses and temperatures. Here it has been demonstrated [3,4] that moving from a steady to an unsteady analysis can change temperature predictions by hundreds of degrees. Analyses of this type are now seen as essential, albeit under some restrictive assumptions that make the analysis affordable. Some other efforts to explore the effects of unsteady flows are listed in references [5,6,7,8,9].

Large, multistage unsteady calculations can be justified in an industrial environment only if they return information valuable to the designers at an affordable cost. The cost is twofold: the direct cost of the computation and, more important, the cost of the lost time in the design cycle. It is a fact that turbomachinery designers have to live with that companies want to improve already good designs (thus requiring ever more sophisticated analysis) using ever fewer engineers, and without lengthening the time period between launching the design project and delivering the first engine. This paper reports an attempt to determine whether a large unsteady multistage calculation can *today* meet two criteria: (1) that the differences between the steady and unsteady calculations tell designers something new and useful and (2) that the calculations can be completed on currently available machines in a short enough time that an industrial organization would contemplate doing them on a regular basis as part of the design process.

The component selected for this investigation is the low-pressure turbine of a large, high-bypass ratio jet engine.

LOW-PRESSURE (LP) TURBINE

An aircraft engine low-pressure turbine is a critical component, as it drives the large diameter fan that propels the airplane, and the LP turbine is our last chance to extract power from the gas stream (Figure 1). We wish to explore the impact of unsteady flow analysis on our understanding of this engine component—both in terms of overall quantitative assessments (e.g., efficiency) and in terms of furthering our understanding of the fundamental flow physics.

A typical LP turbine may have up to seven stages (14 blade rows). Each blade row will have around 100 blade passages. If each blade passage is meshed using 3 million grid cells, we obtain an overall mesh count of around 5 billion grid cells. If we were to require 100,000 time steps to reach a periodic state at which we can begin to query the solution, we would require 5×10^{14} unit operations—where a unit operation advances one grid cell one time step. The equations are nonlinear, and each unit operation involves many inner iterations. If we are to use unsteady simulations as a practical design and analysis tool, we need to simplify, even on the largest of current machines like Jaguar.

Our first simplification involves analyzing only four stages. The four stages are from a turbine test rig, with blade designs representative of the front stages of a modern aviation LP turbine. The second simplification is to adjust the blade counts. If the number of blades in a row were the same for all rows, we would only need to analyze the flow in one passage for each row, even when the flow is unsteady. Unfortunately blade designers are indifferent to the needs of computational analysts: there are good reasons for having different numbers of blades in each row, driven by the need to avoid forced vibration problems. If, however, the blade counts in each row were different but shared a common factor S, it would be possible to analyze only a fraction $1/S$ of the full annulus. What we do is to take the existing blade counts and ask what is the minimal shift in the blade counts needed to produce a significant common factor. Then we go back into the design system and produce a closely related design with this common factor. In this case we were able to come up with a common factor of 14 and thus analyze only 1/14 of the annulus. This allowed analysis of only 67 blade passages in the eight blade rows, with a total grid count of around 200 million cells (Figure 2). At this scale we felt that we had a good compromise between degrading the fidelity of the simulation (primarily because of the shifts in blade counts to make a limited sector analysis possible) and making the best use of the available CPU-hours.

Details of the solution setup for this case will be deferred until after a brief description of the flow solver.

FLOW SOLVER

The flow solver TACOMA has been under development at General Electric Global Research for sixteen years. It comprises almost 500,000 lines of code, from twenty or more authors. The equations solved are the steady, Reynolds-averaged Navier-Stokes equations (RANS), or the unsteady, Reynolds-averaged Navier-Stokes equations (URANS). TACOMA is a density-based, multiblock time-marching code using the well-established Jameson, Schmidt, Turkel (JST) algorithm [10]. Available turbulence models include several flavors of Wilcox's (k,ω) model [11] and Menter's SST model [12]. A two-equation transition model due to Menter is available [13]. Gas models include a fully perfect gas, a perfect gas in which the specific heat ratio is a linear function of temperature, and any gas model for which the user is able to supply *C0* or *C1* continuous tables of selected gas properties in the form $c = f(a,b)$—for example, pressure as a function of density and internal energy. A nonequilibrium wet steam model due to Gerber [14] is also available. Unsteady solutions are obtained using Jameson's dual time-stepping algorithm [15]. The code can provide steady solutions for a single blade passage (the workhorse application); quasi-steady multistage solutions using a variant of "mixing planes" to deal with the task of averaging out the unsteadiness at the interfaces between blade rows in the most physically useful way; single blade row unsteady solutions using a "frozen gust" input; multistage unsteady solutions using sliding mesh interfaces; linearized frequency domain solutions for blade flutter; and more. An analogous unstructured grid code is under development, to which TACOMA can be tightly coupled.

Parallel solutions are obtained by domain decomposition, and communication between processors is handled using MPI. The code contains over one thousand calls to MPI routines. Load balance is achieved by assuming that load is simply a function of the number of grid cells on a processor. Grid blocks are assigned to processors via a simple greedy algorithm: assign the largest unassigned grid block to the least loaded processor; repeat.

One aspect of the load balance/domain decomposition issue may be of interest. In a single blade passage, containing of order 3 million grid cells, the grid cells may be grouped in of order 10 to 20 grid blocks, each a structured three-dimensional array of cells. If we want to run on one hundred or more processors per blade passage, the ten or twenty grid blocks must be split. The number of possible combinations of block splits is large—easily more than 10^{100}. Thus, though we have finessed the NP complete problem of block-to-processor assignment via our crude but effective greedy algorithm, we have created another probably NP complete problem of deciding the optimum block splitting. If we think of grid blocks as rocks and pulverize them to gravel, our greedy algorithm gives excellent load balance—*at least as we have defined it*. Gravel, however, entails a massive communication load between the small pieces. What we need is a block split that gives the largest, most compact pieces (and thus the lowest additional communication load) with adequately uniform processor loads. We know of no algorithm analogous to our simple assignment algorithm that will give an acceptable solution. What we do is to enlist every available processor at the start of the run to do some searching. If each blade passage in a row owns 100 processors, and there are ten passages in the sector of the annulus we are analyzing, we have 1,000 processors available to us. If each processor searches 1 million block splits (taking only a few seconds to do this), we can search a billion candidate block splits. Obviously, 1 billion is obviously not 10^{100}. But we let the processors search systematically through half a billion of the lowest-order split combinations, with each block split into one, two, three, etc., pieces (remember that the multiplicity of solutions lies not in the individual block splits but in the *combinations* of splits). The remaining half-billion split combinations to be tested are selected by a Monte Carlo method, with a simple function biasing the random selection of splits toward the simplest splits. There is no way of knowing, of course, how closely this simple algorithm approaches the optimal split, but it observed that if we list the 100 best splits, they are all very, very similar in terms of load balance and communication cost. Also, increasing the number of searches each processor does by a factor of 10 usually does not provide more than a marginal improvement in the quality of the result.

Further improvements are possible. Clearly, a more effective load balance metric is desirable, one that includes the cost of the noncommunication boundary conditions at the grid block surfaces: flow inlet, flow outlet, walls, and so forth. Another attractive option is that if we have *N* processors available, we could task each processor with performing a serial METIS [16] assignment on the *N* most promising split combinations that are derived on the Monte Carlo/greedy approach and pick the best of those.

As currently configured, the code provides acceptable scaling down to the level of approximately 30,000 grid cells per processor. We are sure we can do better.

SOLUTIONS ON JAGUAR

Solutions were obtained on Jaguar at Oak Ridge National Laboratory, under a 2-million CPU-hour Director's Discretionary Grant. Minimal code changes were required to compile and run on Jaguar.

Each of the 63 blade passages was meshed with approximately 3 million cells, using our standard meshing techniques. The grids had smooth endwalls: to minimize grid counts, we omitted the complex cavity geometries of the tip clearances over the shrouded rotors and the purge cavities at the blade roots. The runs were made with the SST turbulence model, plus a two-equation transition model. The inlet boundary conditions were taken from probe rakes at the inlet to the first stator, taken from the rig test. These included spanwise profiles of total temperature, total pressure, and swirl angle; all profiles were uniform in the circumferential direction. The exit boundary condition was a specified static pressure. One thousand time steps were used for a 1/14 wheel rotor sector to pass a 1/14 wheel stator sector, or 14,000 time steps per wheel revolution. Each time step was set to use thirty inner iterations.

Jaguar runs were made on approximately 8,000 processors, giving a per processor load of roughly 25,000 cells. The solution is started from the quasi-steady multistage solution. The unsteady solution was deemed to have converged when the unsteadiness settled down to a periodicity of 1,000 time steps—the time taken for a sector passing. This is easily observed by outputting a 1,000-time step moving average of key quantities, such as the exit mass flow, average exit total pressure, and average total pressure at the exit of each blade row. When the solution is periodic, these averages are flat, and vice versa. It was at first surprising that the solution stabilized in only 6,000 time steps—or less than one-half of a wheel revolution. A better time scale than the rotation period is the throughflow time—the approximate time a fluid particle takes to pass from the inlet to the first stator to the exit of the last rotor. On this scale 6,000 time steps represents between twenty and thirty throughflow times, and convergence on this time scale looks more plausible. Once a periodic state was reached, two further runs, each of 1,000 time steps, were undertaken. In one the code created a 1,000-time step average of the unsteady solution, for direct comparison with the steady multistage solution. In the second, solution snapshots were captured over 1,000 time steps, at an interval of five time steps, or 250 snapshots in all. This was a compromise between sufficient time resolution of the unsteady flow for visualization, and excessive data storage requirements. The total storage required for the snapshots was approximately 3.5 terabytes. All these runs consumed about half of our allocation. Convergence to a periodic state over 6,000 time steps required approximately 60 wall-clock hours. Each additional run for 1,000 time steps required approximately 10 wall-clock hours. , the total solution time was around 80 hours on a small fraction—around 3.5%—of the available processors on Jaguar.

For the second half of our allocation we repeated the calculation, but this time allowing for the presence of twelve mid-frame struts ahead of the turbine (Figure 3). The struts interact with the unsteady flow from the upstream high-pressure turbine and produce strong wakes. Our assumption of circumferential nonuniformity of the inlet boundary conditions eliminates the strut wakes from the calculation. By a careful combination of experimental data obtained from probes between the strut and the first stator (the inlet surface for our first calculation above), and unsteady CFD involving the high-pressure turbine rotor upstream of the strut and the strut itself, we were able to reconstruct our best estimate of the circumferential variation in the upstream boundary conditions at the inlet to our domain—scaling, of course, to a 1/14 wheel sector. Variations of total temperature, total pressure, and flow angle are shown in Figure 4. This calculation also converged quickly, in 6,000 time steps.

SOLUTIONS

Turbomachinery designers use CFD for several purposes: (1) to provide *absolute* performance predictions (not usually seen as reliable); (2) to provide *relative* performance predictions when comparing design iterations (critical for the design process); and (3) to provide designers with a microscope to examine—and thus understand—the flow field with a resolution, and at a cost, not available in an experimental test.

In terms of absolute performance, the results obtained are, in a sense, remarkable. In going from a steady to the first unsteady analyses, the efficiencies of the four individual stages of the turbine do not shift by much more than one percentage point, and the overall efficiency of the four stage turbine does not shift at all (Table 1). The shifts in going from the unsteady calculation without strut wakes to the calculation that includes the strut wakes are of similar magnitude. This is a validation—*in this case, at least*—of the usefulness of the assumption of steady flow in the relative frame. This is a negative conclusion, perhaps, but nevertheless useful.

Figure 5 shows contours of static pressure at 20% of the span, for three cases: (1) the steady multistage analysis and two instantaneous snapshots from the unsteady analysis, (2) with the strut wakes, and (3) without the strut wakes. At the resolution of the figure it is impossible to discern any differences between the three images. Since it is primarily the pressure forces at the blade surfaces that drive the turbine and produce the exchange of power between the gas stream and the shaft, this is in accord with the minimal shifts in efficiency noted above.

Figure 6 shows contours of entropy at 20% span. In the steady multistage solution the blade wakes are mixed out at each interface between blade rows. This behavior is consistent with the mixing plane assumption of the steady multistage model. In the first unsteady solution, the wakes are seen to persist in, and to move unsteadily through, the downstream blade row. In the second unsteady solution, the wakes of the strut, marked by high entropy, persist throughout the turbine, moving diagonally from upper left to lower right of the picture. Much other detail can be seen as well, detail that images of this size cannot do justice to. In the leftmost row of stators, for example, immediately behind the strut, the stator blade fully bathed by the strut wake shows a clear pressure side separation. But, again, the effect of this detail on the overall performance is minimal

Figure 7 shows spanwise profiles of entropy at the inlet to the turbine and at the exit of each blade row. The stators are labeled LPN*; the rotors LPR*. The solid lines are from the steady multistage calculation. The dashed lines are from a time average of the first unsteady calculation (inlet boundary condition varying in span only; no strut wakes). The steady increase in entropy from inlet to exit is, of course, to be expected. What is notable is that, at each blade row exit, the unsteady solution shows more entropy at mid-span and less entropy near the endwall (0% and 100% span). This does not mean that in the unsteady solution more entropy is generated (i.e., there is more loss) at mid-span than in the steady multistage solution, and vice versa near the endwalls. What we are seeing are the effects of spanwise mixing that is included in the unsteady solution and missing in the steady multistage solution. In the unsteady solution higher entropy fluid is being *transported* from the endwalls toward mid-span.

A wealth of other information can be mined from the unsteady solution snapshots and from their time averages, and many questions remain to be answered:

- How much does the unsteady flow increase the mixing of the higher loss fluid near the endwalls into the relatively clean flow at mid-span?
- Does the transition model properly predict the influence of the upstream blade wakes on the transition from laminar to turbulent flow?
- How do the secondary flow vortices near the endwalls behave as they pass from one row to another?

Mining of the "data" contained in these solutions is ongoing and will surely lead to a better understanding of low-pressure turbine aerodynamics and to better designs. Many questions also could be answered by other similar or more extensive calculations:

- To what extent has the scaling of the blade counts changed the solution; how different would the true full annulus solution (fourteen times as expensive) be?
- These four stages are the front half of a turbine with seven stages. What are we missing by omitting the last three stages?
- This analysis neglects purge flows coming from the interior of the turbine, leaks across the roots of the stators, and leaks across the tips of the rotors. Inclusion of these flow features would be expensive—probably doubling the total grid size and slowing convergence of the solution (because of the low Mach numbers of the leakage flows). What might we learn from including purge flows and leakages?
- Are the conclusions we can draw from this case any different from the conclusions we might draw for a low-pressure turbine that occupied a different portion of the available design space?

ACKNOWLEDGEMENTS

The authors thank the Oak Ridge National Laboratory for the Director's Discretionary Grant of the 2 million Jaguar CPU-hours used in performing these calculations, and GE Aviation and GE Global Research for their support. This research used resources of the Oak Ridge Leadership Computing Facility at the Oak Ridge National Laboratory, which is supported by the Office of Science of the U.S. Department of Energy under Contract No. DE-AC05-00OR22725.

REFERENCES

1. Denton, J. D., 2010, "Some limitations of turbomachinery CFD," ASME Paper GT2010-22540
2. He, L., 2010, "Fourier methods for turbomachinery applications," *Progress in Aerospace Sciences* 46 (1010) 329-341.
3. Felten, F. N., Kapetanovic, S., Holmes, D. G., Ostrowski, M., 2008, "Gas turbine temperature prediction using unsteady CFD and realistic non-uniform 2D combustor exit properties,"ASME Paper GT2008-50275.
4. Felten, F. N., Kapetanovic, S., Holmes, D. G., Ostrowski, M., 2007, "Impact of combustor exit circumferential flow gradients for gas turbine temperature prediction," in Proceedings of the 5th International conference on Computational Heat and Mass Transfer, Canmore, Canada, June 18-22.
5. Yao, J., Davis, R. L., Alonso, J. J., Jameson, A., 2002,"Massively parallel simulation of the unsteady flow in an axial turbine stage," *AIAA Journal of Propulsion and Power* 18, no. 2, pp. 465-471 (presented as AIAA Paper 01-0529, 2001, "Unsteady flow investigations in an axial flow turbine using the massively parallel flow solver TFLO").
6. Shankaran, S., Alonso, J. J., Liou, M., Liu, N., and Davis, R. L., 2001, "A multi-code-coupling interface for combustor/turbomachinery simulations," AIAA Paper 01-0974.
7. Davis, R. L., Yao, J., Clark, J. P., Stetson, G., Alonso, J. J., Jameson, A., Haldeman, C. W., and Dunn, M. G., 2004, "Unsteady interaction between a transonic turbine stage and downstream components," *International Journal of Rotating Machinery* 10, no. 6 (presented as ASME GT-2002-30364, International Gas-Turbine Conference, Amsterdam, The Netherlands, June 2002) .
8. Davis, R. L., Yao, J., Alonso, J. J., Paolillo, R. and Sharma, O. P., 2004, "Prediction of main/secondary-air system flow interaction in a high-pressure turbine," *AIAA Journal of Propulsion and Power* 20, no. 6 (presented as AIAA paper AIAA-2003-4833 at the AIAA/ASME Joint Propulsion Conference, July 2003).
9. Yao, J., Cargill, P. L., Holmes, D. G., Gorrell, S., 2010, "Aspects of numerical analysis for unsteady flows in aircraft engines," AIAA paper AIAA-2010-1603.
10. Jameson, A., Schmidt, W., Turkel, E., 1981, "Numerical solution of the Euler equations using finite volume time-stepping schemes," AIAA paper AIAA 1981-1259 (presented at the 14th Fluid and Plasma Dynamics Conference, Palo Alto, June 1981).
11. Wilcox, D. C., 1988, "Re-assessment of the scale-determining equation for advanced turbulence models," *AIAA Journal* 26, no. 11, pp. 1299-1310.
12. Menter, F., 1993, "Zonal two equation k-ω turbulence models for aerodynamic flows," AIAA paper 1993-2906.
13. Langtry, R., Menter, F. , 2006, "Overview of industrial transition modelling in CFX," ANSYS technical report TPL 8126.
14. Gerber, A.G., 2008, "Inhomogeneous multifluid model for prediction of non-equilibrium phase transition and droplet dynamics," *J. Fluids Eng.* 130, no. 3.
15. Jameson, A., 1991, "Time dependent calculations using multigrid with applications to unsteady flows past airfoils and wings," AIAA paper AIAA 1991-1596.
16. Karypis G., Kumar, V., 1999, "A fast and high quality multilevel scheme for partitioning irregular graphs," *SIAM Journal on Scientific Computing* 20, no. 1, pp. 359-392.

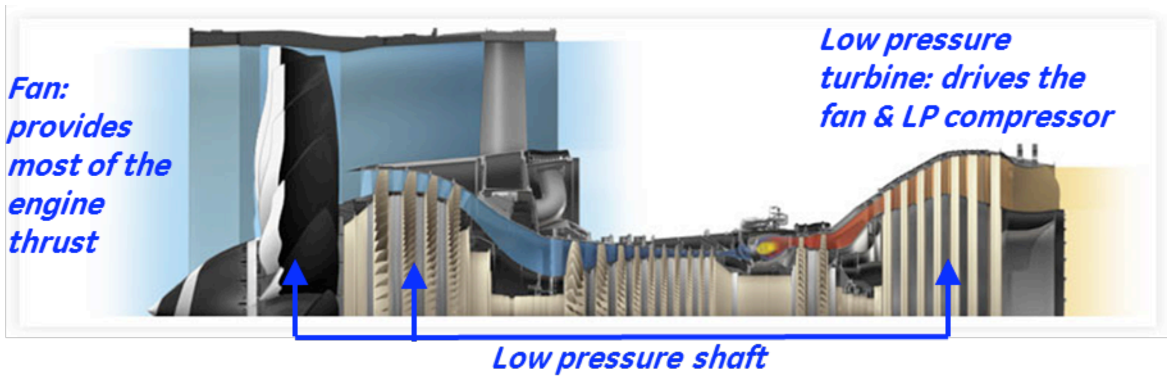

Fan: provides most of the engine thrust
Low pressure turbine: drives the fan & LP compressor
Low pressure shaft

Figure 1. Low-pressure turbine of a modern, high-bypass jet engine.

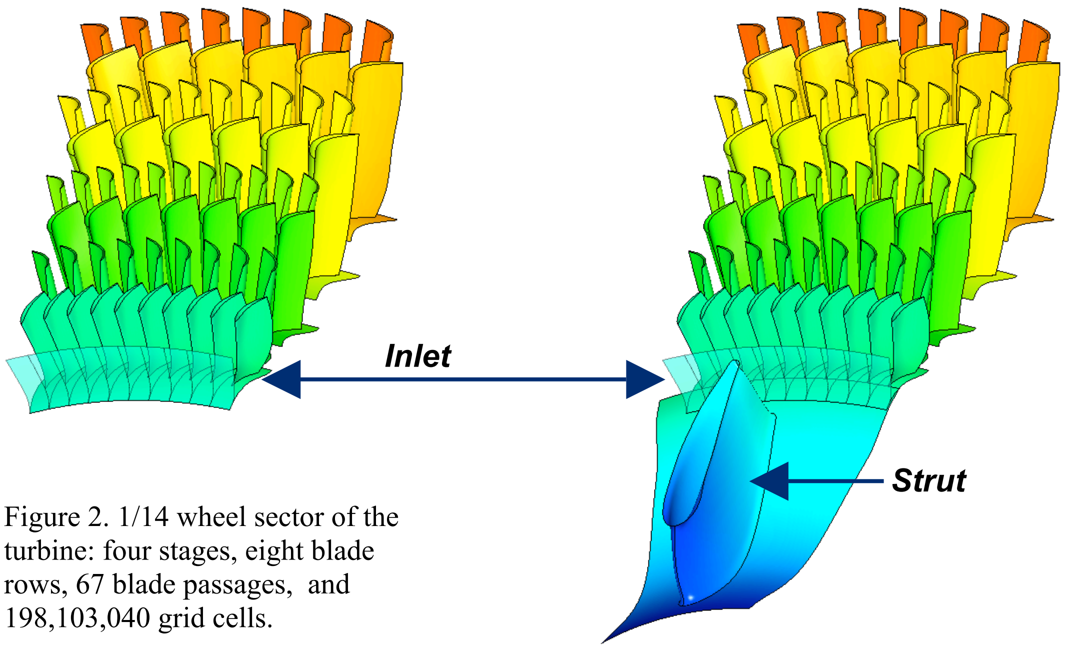
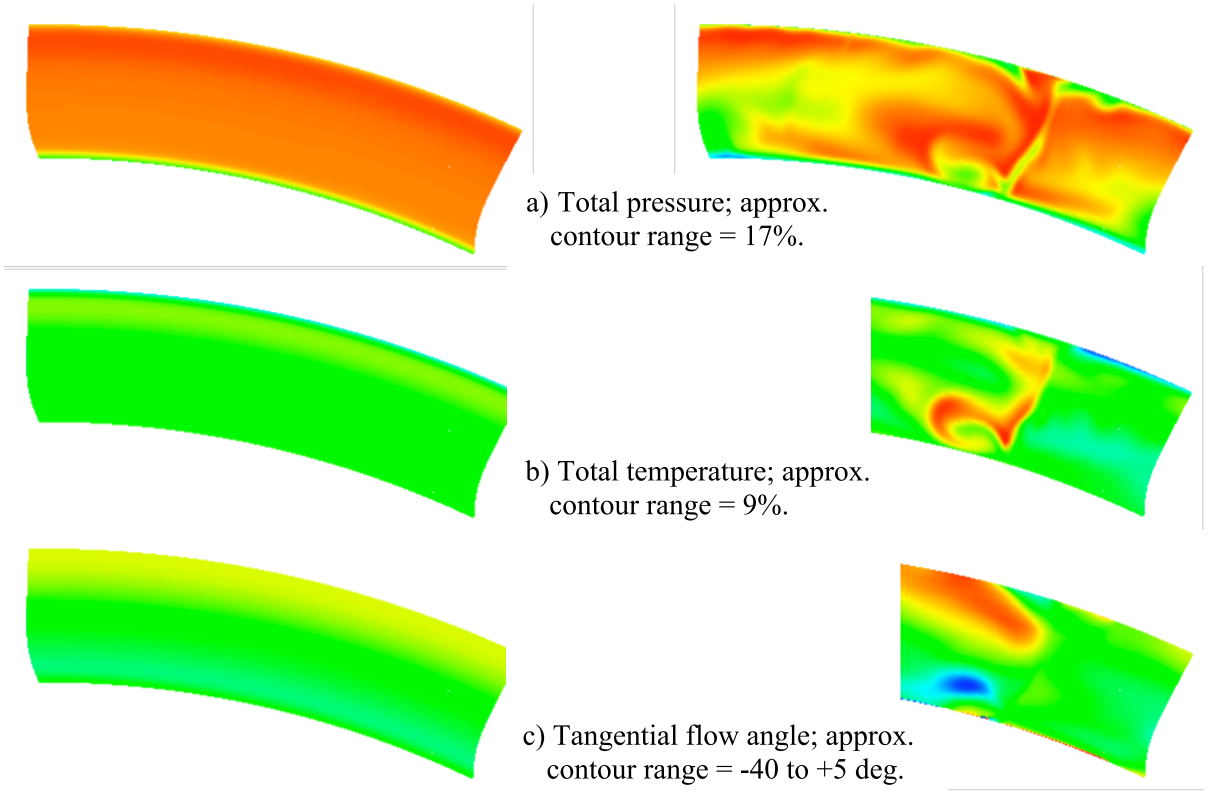


Figure 2. 1/14 wheel sector of the turbine: four stages, eight blade rows, 67 blade passages, and 198,103,040 grid cells.

Figure 3. Mid-frame strut. The mid-frame strut is not included in the calculation but provides pitchwise variation at the inlet boundary surface.



a) Total pressure; approx. contour range = 17%.

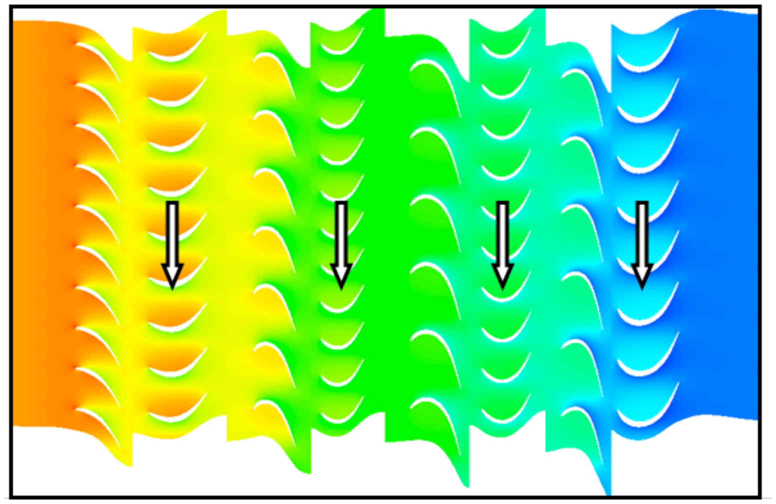
b) Total temperature; approx. contour range = 9%.

c) Tangential flow angle; approx. contour range = -40 to +5 deg.

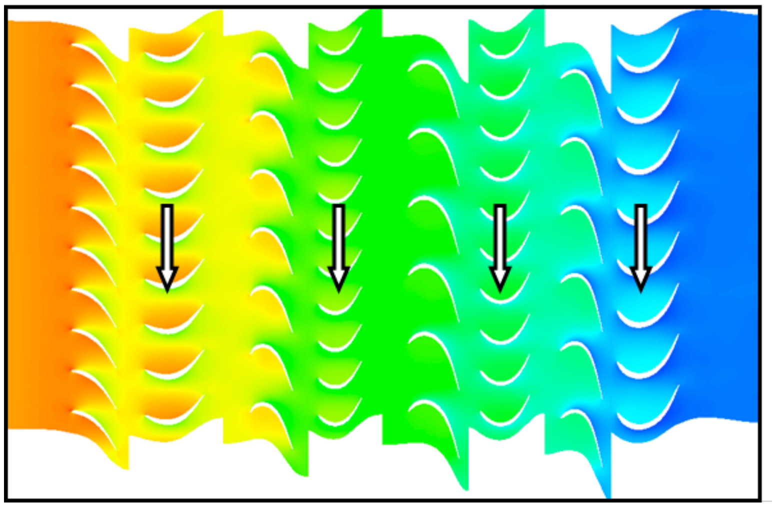
Figure 4. Inlet boundary conditions. Left column: varying spanwise, uniform pitchwise; right column: varying spanwise ***and*** pitchwise. (a) Total pressure, (b) total temperature, (c) tangential flow angle.

Table 1. Changes in stage and component efficiency (percentage points), unsteady solutions vs. steady multistage solution.

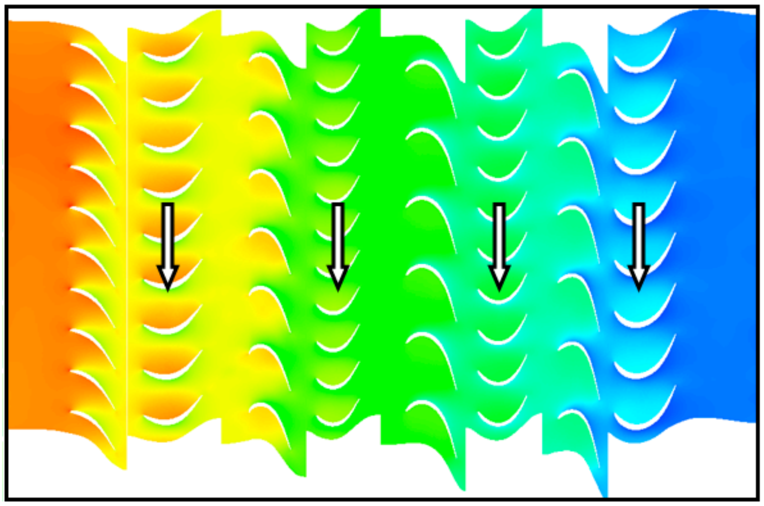
	Uniform Inlet	Nonuniform Inlet
Stage 1	+0.03	-0.49
Stage 2	-1.27	-0.07
Stage 3	+0.22	-0.02
Stage 4	-0.27	+0.00
Component	+0.00	-0.10



a) Steady multistage solution.

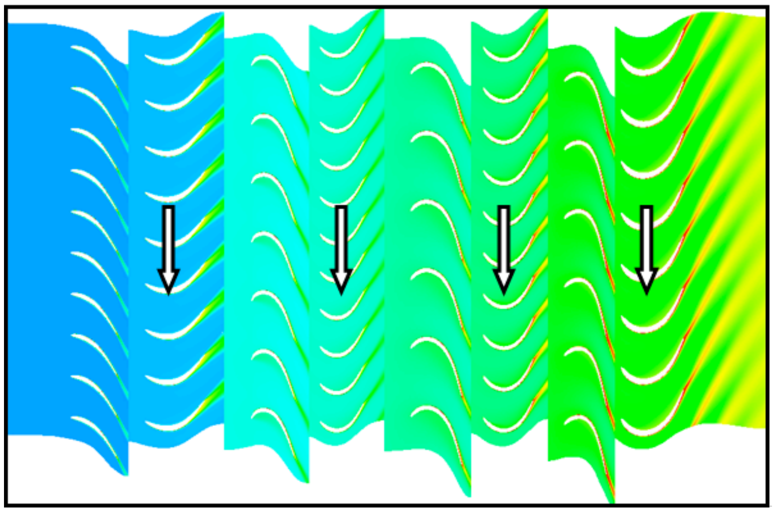


a) Instantaneous solution snapshot, unsteady solution #1.

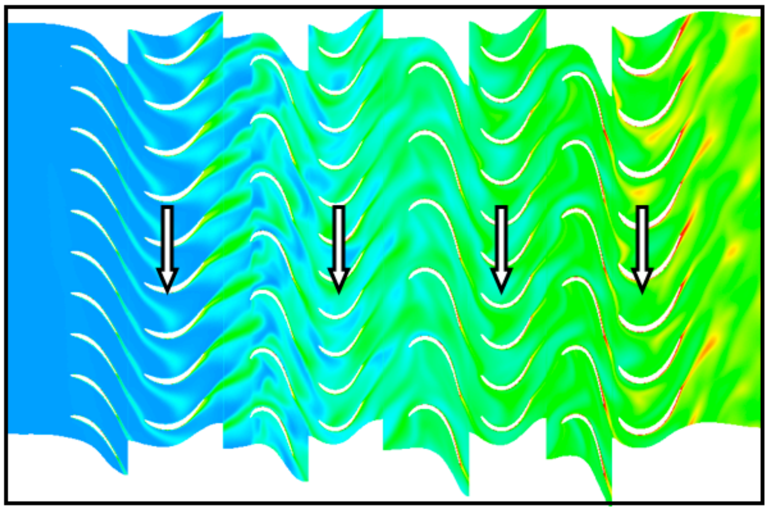


b) Instantaneous solution snapshot, unsteady solution #2.

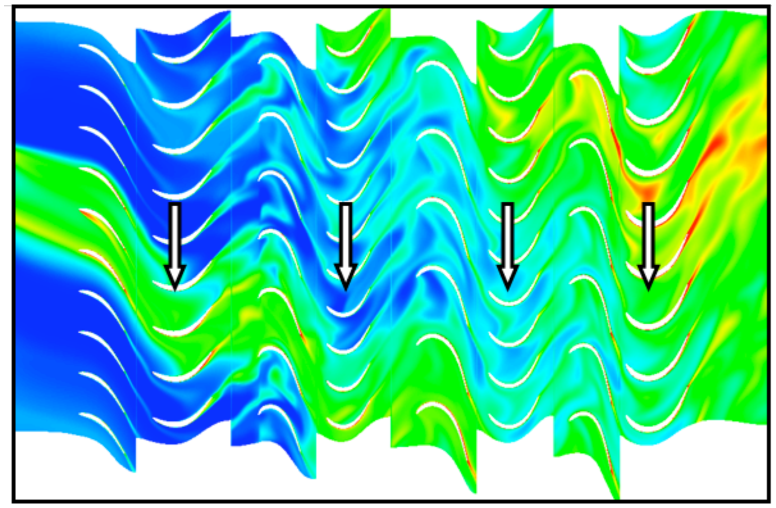
Figure 5. Static pressure at 20% span.



a) Steady multistage solution.



b) Instantaneous solution snapshot, unsteady solution #1.



c) Instantaneous solution snapshot, unsteady solution #2.

Figure 6. Entropy at 20% span.

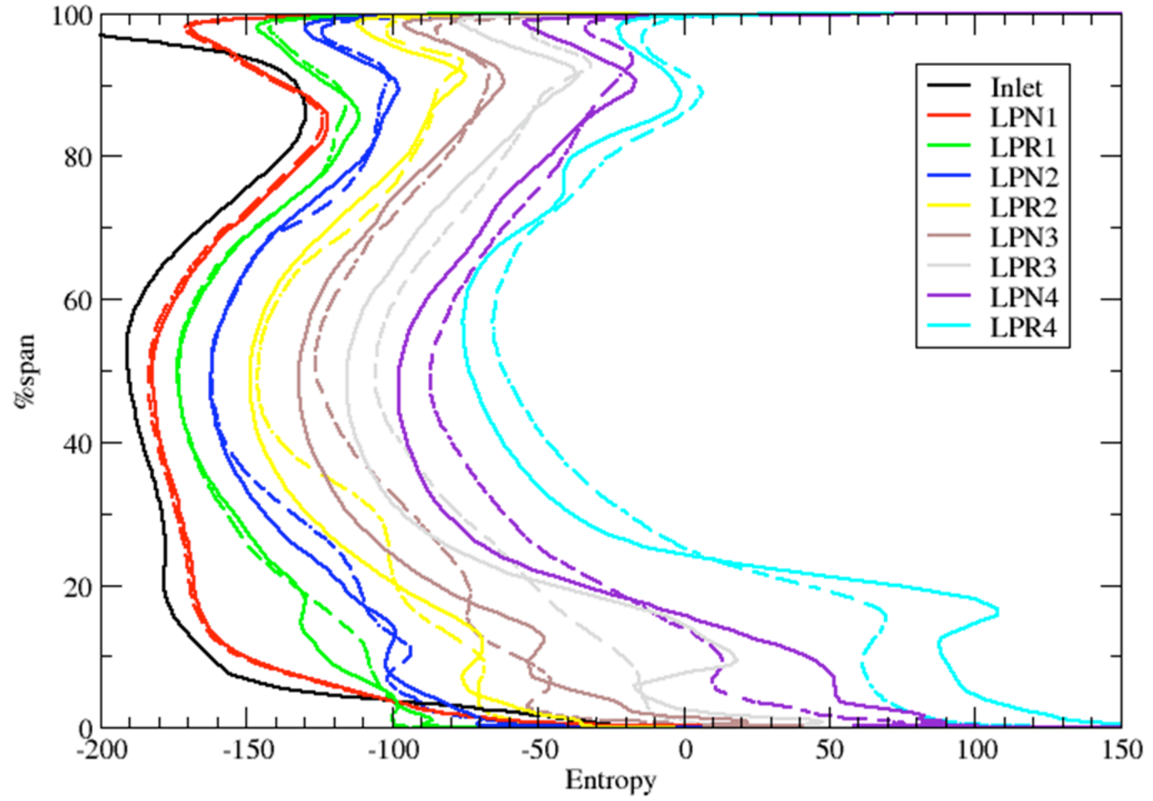


Figure 7. Spanwise profiles of entropy at the inlet (black) and at the exit of each blade row. Solid lines: steady multistage solution; dashed lines: time average of unsteady solution #1 (pitchwise uniform inlet conditions).