36th AIAA Fluid Dynamics Conference and Exhibit
5 - 8 June 2006, San Francisco, California

AIAA 2006-3063

# Validation of a New Parallel All-Speed CFD Code in a Rule-Based Framework for Multidisciplinary Applications

Ramji Kamakoti
Dep. of Mechanical and Aerospace Engineering
University of Florida
Gainesville, FL, U.S.A
ramji@ufl.edu

Siddharth Thakur
Streamline Numerics, Inc.
and
Dep. of Mechanical and Aerospace Engg., Univ. of Florida
Gainesville, FL, U.S.A , sst@ufl.edu

Jeffrey Wright
Streamline Numerics, Inc.
Gainesville, FL, U.S.A
jeffw@atlantic.net

Wei Shyy
Dep. of Aerospace Engineering
University of Michigan
Ann Arbor, MI, U.S.A
weishyy@umich.edu

## ABSTRACT

This paper focuses on the validation of a new all-speed Computational Fluid Dynamics (CFD) code called *Loci*-STREAM. This computational package is not just another CFD solver; rather, it integrates proven numerical methods and state-of-the-art physical models to compute all-speed flows using generalized grids in a novel rule-based programming framework called *Loci* which allows: (a) seamless integration of multidisciplinary physics in a unified manner, and (b) automatic handling of massively parallel computing. The objective is to be able to routinely simulate problems involving complex geometries requiring large unstructured grids with arbitrary polyhedral cells and complex multidisciplinary physics. As a first step towards achieving this objective, a wide range of model test cases are studied here, including incompressible laminar and turbulent flow cases, inviscid compressible flow cases, compressible turbulent flows with wall heat transfer as well as internal turbulent flows in 3D geometries and unsteady computations. Comparison of the code with experimental and prior benchmark numerical results is done to validate the robustness of the code for flows ranging from incompressible to supersonic regimes. A scalability analysis is performed as well to study the efficiency of parallelization.

## 1. INTRODUCTION

Over the last decade significant new technologies have been developed which allow the design of reliable computer codes for simulating multidisciplinary physics involved in a wide variety of engineering flow applications. In the authors' opinion, a state-of-the-art CFD solver must have: (a) a generalized unstructured grid (arbitrary polyhedral cells) capability to allow ultimate flexibility in grid generation for complex geometries and to allow simulation of various sub-components of modern engineering devices in a coupled manner, (b) good scalability on modern parallel computers, especially distributed memory clusters, (c) robust and efficient time-stepping schemes, (d) reliable engineering grade turbulence models for both steady and unsteady flows, and (e) a sound programming framework which reduces the complexity of assembling the various modules of the code while ensuring maximum parallel efficiency in an automatic manner. The *Loci*-STREAM code has been designed to incorporate all of the above elements.

In addition to numerical algorithm development, great strides have been made in the area of program development tools. Recently, a powerful framework for application development called *Loci* [1] has been developed with large-scale multidisciplinary simulations in mind. It is designed to reduce the complexity of assembling large-scale finite-volume applications as well as the integration of multiple applications in a multidisciplinary environment. *Loci* utilizes a rule-based framework for application design. Users of *Loci* write applications using a collection of “rules” and provide an implementation for each of the rules in the form of a C++ class. In addition, the user must create a database of “facts” which describe the particular knowns of the problem, such as boundary conditions. Once the rules and facts are provided, a query is made to have the system construct a solution. One of the main features of *Loci* is its ability to automatically determine the scheduling of events of the program to produce the answer to the desired query, as well as to test the consistency of the input to determine whether a solution is possible given the specified information. The other major advantage of *Loci* to the application developer is its automatic handling of domain decomposition and distribution of the problem to multiple processors.

The *Loci*-STREAM code is being developed in the *Loci* framework and is intended to be a general purpose finite volume multidisciplinary code for simulating fluid flow physics using the Reynolds-averaged Navier-Stokes (RANS) equations in conjunction with conjugate heat transfer and structural stress analysis in the surrounding solid boundaries. The applications include combusting flows in rocket engines, hydraulic turbomachinery, fluid-structure interactions, etc. The test cases presented in this paper are selected to validate individual elements of the code such as (a) accuracy of discretization schemes for incompressible and compressible flows, turbulence models, and time-stepping scheme, (b) efficiency in terms of parallel scalability, and (c) robustness for a wide range of flow regimes. The present work is a first step towards achieving multidisciplinary capability in *Loci*-STREAM involving incorporation of finite rate chemistry for combusting flows, conjugate heat transfer, coupled solid stress analysis, etc.

## 2. OVERVIEW OF LOCI-STREAM

### 2.1 Rule-Based Framework: Loci

Unlike traditional procedural programming systems (C, FORTRAN) in which one writes code with subroutines, or object-oriented systems (C++, Java) in which objects are the major program components, *Loci* uses a rule-based framework for application design. Users of *Loci* write applications using a collection of “rules” and provide an implementation (the code that performs the rule's function) in the form of a C++ class. In simple terms, a rule is nothing more than a small segment of the overall code which performs a specific function (such as assigning a boundary condition). The complete set of rules along with their connections forms the complete program.

A useful feature of *Loci* which makes it particularly suited to multidisciplinary programming lies in the way rules are connected to form the program. When a programmer writes a rule, the rule is documented via a few simple calls to *Loci* functions, which serve to define the input and output variables for the rule. For example, the input for a rule to compute the centroid of each of the triangles forming a 2-D unstructured grid would be the grid nodal coordinates and the node numbers for each triangle. The output for this rule would be an array containing the centroid of each triangle. Given the input and output information for each rule, *Loci* uses a graph-based logic algorithm to determine the connection of the rules to form the resulting computer program. This approach is markedly different from the traditional programming approach where the programmer directly writes not only the functions composing the program, but also explicitly codes the flow of function execution.

The goal of a *Loci* programmer is to define rules with their respective inputs and outputs such that when *Loci* combines the rules to form the program, the proper functional flow is achieved. Should a set of rules be specified which cannot be connected into a meaningful program to arrive at the desired solution (which is the output of the terminating rule), *Loci* will inform the user. This feature can be very useful in the multi-disciplinary coding environment, where often independent teams of developers write different portions of an application (CFD team and FEM team, for example). In such a case, if one is attempting to write a CFD application (defined by a series of rules) which will link with an existing FEM code (also defined by a series of rules), and the rules do not connect to form a meaningful code, *Loci* will fail to produce the code and inform the programmer of the failure. Contrasted with the traditional coding approach, in which the combined application would compile and run, producing meaningless results, *Loci* helps to reduce the burden of inter-application communication in the multidisciplinary coding environment.

### 2.2 Numerical Algorithm

The flow solver is based on the SIMPLE algorithm. It uses a control volume approach with a collocated arrangement for the velocity components and scalar variables like pressure. Pressure-velocity decoupling is prevented by employing the Rhie-Chow momentum interpolation approach [2]; this involves adding a fourth-order pressure dissipation term while estimating the mass flux at the control volume interfaces. The velocity components are computed from the respective momentum equations. The velocity and the pressure fields are corrected using a pressure correction equation. The correction procedure leads to a continuity-satisfying velocity field. The whole process is repeated until the desired convergence is reached. Details of the basic algorithm can be found in [3]. The inviscid fluxes are discretized using either (a) the second-order upwind scheme for incompressible flows or (b) Roe scheme in conjunction with Venkatakrishnan limiters [4] for compressible flows. Turbulence closure is achieved via Menter’s $k$-$\omega$ based BSL and SST models [5], with the option of invoking wall functions near solid boundaries [6].

### 2.3 Generalized Grid Capapbility

The code is designed to handle generalized grids, i..e, unstructured grids with arbitrary polyhedral cells. An example of this capability is shown in Figure 1 for a grid used to model a rocket engine injector. The grid shown consists of hexahederal and tetrahedral cells along with “soccer ball-shaped” cell clusters, which result from grid adaptation employed to track the flame front.

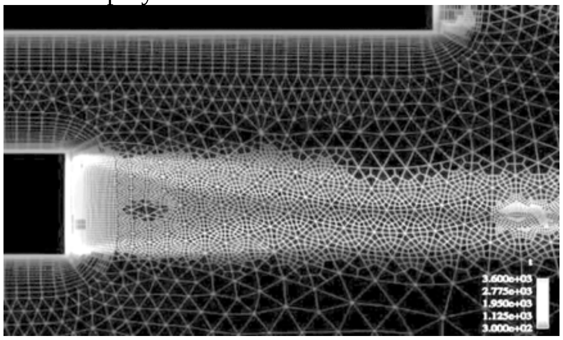


***Figure 1**. Generalized grid used for a rocket engine injector.*

### 2.4 Parallel Computing Capability

The scalability of *Loci*-STREAM for parallel architectures is demonstrated by running the code on two different parallel architectures:

(a) An SGI Altix computer at the University of Florida which has 16 Itanium 1.3 GHz processors with 16 GB shared memory. A laminar flow past a backward-facing step using a grid consisting of approximately 75,000 nodes is used. A perfectly linear scalability is achieved as shown in Figure 2.

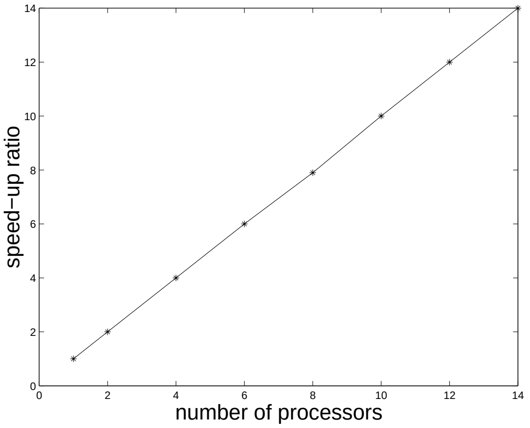


***Figure 2.*** *Scalability of Loci-STREAM on a shared memory parallel architecture.*

(b) The second parallel architecture used is a computer cluster at NASA Marshall Space Flight center. This cluster consist of Athlon 1800 Hz CPUs with 512 MB RAM per CPU, along with a network switch. The test case used here is a hydraulic turbine draft tube geometry with roughly one million nodes. It is observed that parallel benefit ceases when the number of cells per CPU drops below 15,000 cells for the 100 Mbps network and 2,500 cells for the 1,000 Mbps network as shown in Figure 3.

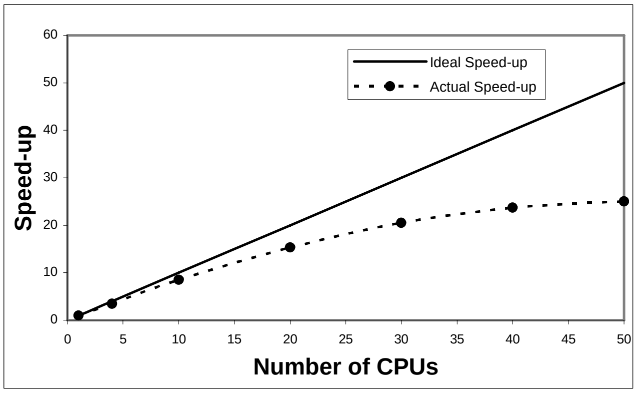


***Figure 3*** *Scalability of Loci-STREAM on a distributed memory networked cluster.*

## 3. RESULTS AND DISCUSSION

In this section we present the main results of our CFD simulations using different test cases and grids.

### 3.1 Lid Driven Cavity Flow

This case was chosen to validate laminar, incompressible, steady state computation for a lid-driven cavity flow and compare the results with the benchmark solution of Ghia et al. [7]. Four different grids are compared here: 21x21, 41x41 and 81x81 structured grids as well as a 6500 node unstructured mesh with triangular elements. The Reynolds numbers considered for this case is 1000 based on cavity depth. The central line u- and v-velocity for the four different grid configurations are shown in Figure 4. The benchmark solution by Ghia is also shown in the figure for comparative purposes. It is evident from the plot that the 81x81 grid and the unstructured mesh produced the most accurate result when compared with the benchmark solutions. Results from the 81x81 structured grid and the 6500-node unstructured grid are indistinguishable, as can be seen from the plot.

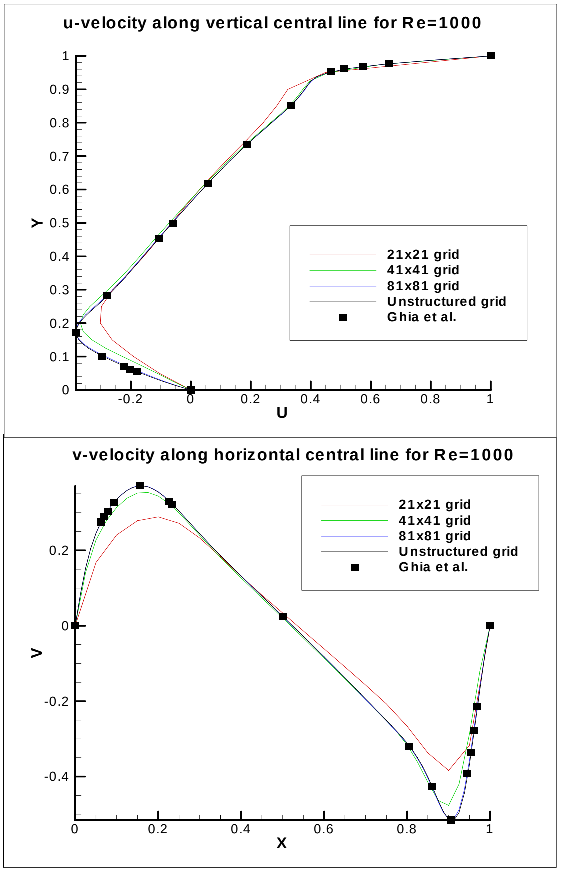


***Figure 4****. u- and v-velocity plot along cavity central line for Re=1000 using different grid configurations and comparison with benchmark solution by Ghia et al.*

### 3.2 Turbulent Flow over a Backward-Facing Step

This case is used to validate a turbulent, incompressible flow situation past a backward facing step. The height of the step is 0.0127 m. The grid used for this case is a hybrid grid with a structured region near the wall regions and an unstructured mesh in the interior domain. The overall mesh had a total of 101K points. The mesh, zoomed in near the step, is shown in Figure 5. The case studied here is identical to the benchmark model used by Menter [5]. The inlet velocity profile, $k$ and $\omega$ available from experimental data [8] are specified at the inlet. A fixed pressure of 0 is specified at the outlet. The corresponding Reynolds based on inlet height for this case is about 40,000.

Several turbulence models were tested for this case. Both Menter's [5] SST and BSL models with first-order and second-order upwinding schemes for turbulence inviscid fluxes with and without Venkatakrishnan limiters [4] were considered to study the impact of each of these models on solution accuracy. A summary of these models is presented via the plot of skin friction coefficient at the lower wall, as shown in Figure 6. It can be seen from the plot that Menter's SST model with second-order upwind for turbulent inviscid flux, with or without limiters, produced the closest agreement to experimental findings. A small offset from the SST results are observed while using Menter's BSL model and second-order upwind scheme for turbulence inviscid fluxes. The difference between the BSL and SST models lies in the way the turbulence viscosity is computed. However, when first-order upwind scheme is used for turbulence inviscid fluxes, we see a significant deviation from experiment. All models were consistent in predicting the skin-friction coefficient near the inlet region leading to the step. The reattachment length obtained for the SST and BSL models were 6.4 and 5.74 respectively, compared to the experimental value of 6.4. Menter, in his paper [5], obtained a value of 6.5 and 5.9 for the SST and BSL models, respectively for a 240x240 structured grid. Our close agreement to the experimental value could be attributed to the choice of grid used, however, Menter did not mention about the value of y+ near the wall in his paper. For our grid, the y+ along the entire bottom wall was less than 1. The plot of recirculation zone for the SST model is shown in Figure 7.

### 3.3 Compressible Flow Over a Bump

An inviscid flow in a channel with a bump is used to assess the code for compressible flows over a range of Mach numbers, including subsonic, transonic and supersonic flows. The chord length of the bump is the same as the channel height, and the channel length is three times its height. The thickness of the bump is 6% of its length. Two grids are used for this case: a single block structured mesh comprising of 121x81 nodes and an unstructured mesh with 10K nodes. The computational domain with the boundary condition planes is shown in Figure 8. The zoomed in regions of the structured and unstructured grids are shown in Figure 9.

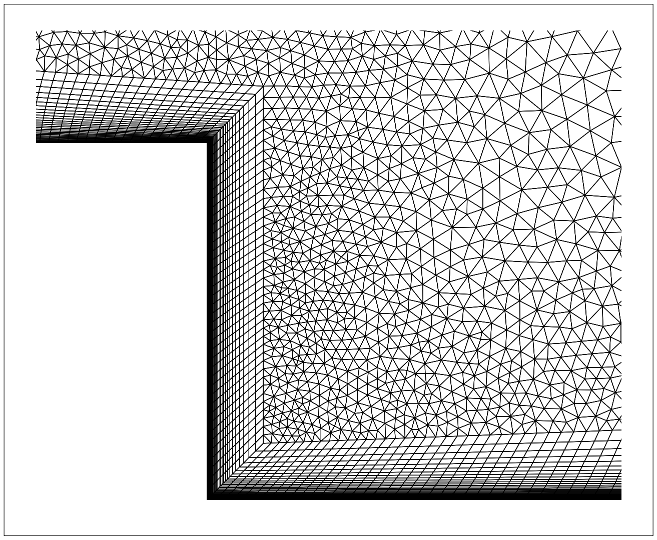


***Figure 5***. *Computational domain for backward facing step along with boundary condition planes*

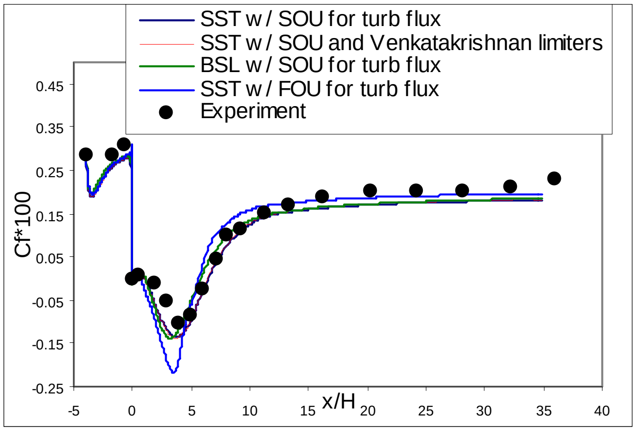


***Figure 6***. *Plot of skin-friction coefficent versus x on the lower wall using various turbulence models*

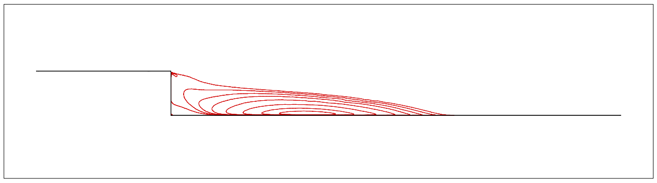


***Figure 7***. *Recirculation zone0 past the backward-facing step using Menter's SST model with no wall functions*

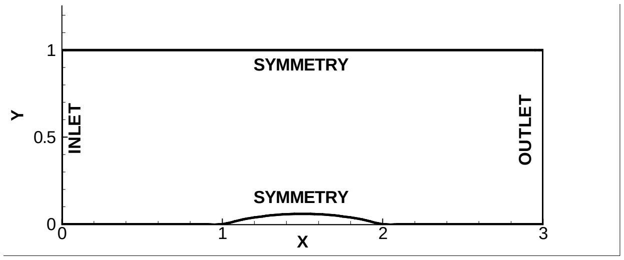


***Figure 8**. Computational domain along with boundary condition planes for flow over a bump*

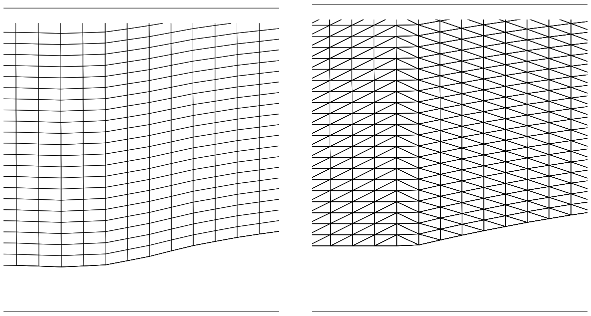


***Figure 9**. Structured and unstructured meshes zoomed in near the bump.*

For the subsonic and transonic case, the boundary conditions involve specifying (a) the mass flux, total temperature and the flow angle at the inlet, or (b) total pressure, temperature and flow angle at inlet, and static pressure at the outlet. For the supersonic case, all the flow variables are specified at the inlet. At the outlet, all the variables are extrapolated from the interior of the domain. The assigned boundary conditions for the different boundary conditions are tabulated in Table 1.

***Table 1**. Boundary conditions for different Mach numbers for the bump case*

| | SUBSONIC (M=0.5) | | TRANSONIC (M=0.75) | SUPERSONIC (M=1.65) |
|---|---|---|---|---|
| **Inlet** | $\dot{m}$ =3.3636 slugs/s<br>T=558.77 R | Po=46.653 psi | $\dot{m}$ =5.05 slugs/s<br>T=558.77 R | v=1943.6 ft/sec<br>p=39.329 psi<br>T=558.77 R |
| **Outlet** | Ps=39.329 psi | Ps=39.329 psi | Ps=39.329 psi | Extrapolated |
| **Initial Condition** | v=588.98 ft/s<br>p=39.329 psi<br>T=558.77 R | v=588.98 ft/s<br>p=39.329 psi<br>T=558.77 R | v=883.5 ft/s<br>p=39.329 psi<br>T=558.77 R | v=1943.6 ft/sec<br>p=39.329 psi<br>T=558.77 R |

Computations are performed using two types of convection schemes, namely, second order upwind (SOU) and the Roe upwinding scheme, however only results pertaining to the Roe scheme are presented here. The Mach number contours for the three different Mach numbers using the Roe scheme for the different grids used are shown in Figure 10. The subsonic case result agrees with the analytical solution in that it is symmetrical with no shock waves observed in the domain. For the transonic case, a mild shock is seen at the trailing edge, as expected for theoretical and prior numerical results [9]. For the supersonic case, two oblique shocks are formed at either corners of the bump. The shock from the leading edge reflects off the top wall and intersects with the shock from the trailing edge. This is also in excellent agreement with literature [9].

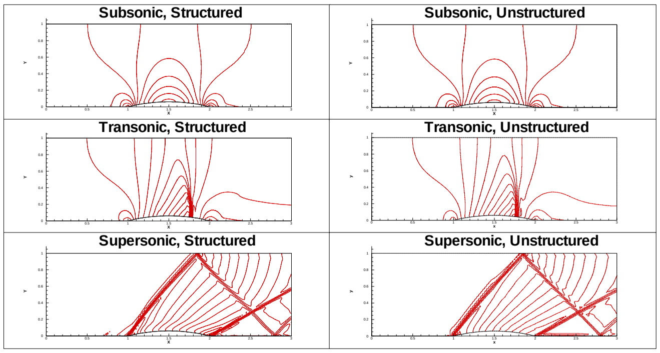


***Figure 10**. Mach number contours for subsonic, transonic and supersonic Mach numbers for flow over a bump for both structured and unstructured grids*

### 3.4 Unsteady Laminar Flow Over a Circular Cylinder

This particular case was studied to test the robustness of the unsteady feature of the code. The Reynolds number used here is 100 based on the diameter of the cylinder. A random perturbation is given downstream of the cylinder to quickly reach the periodic oscillation pattern. Two types of grids are investigated here. A single block O-grid with 201x121 points and a multi-block grid with 8 blocks, which includes an inner O-grid are used. The multi-block grid has almost the same number of points (25K) as the single block grid. The computational setups for the 2 grids are shown in Figure 11 and Figure 12.

A constant horizontal velocity is specified for inlet boundary condition along with extrapolation of velocity and specified pressure at the outlet boundary. The computed Strouhal number (fD/U) for both cases tabulated in Table 2 shows excellent agreement with the experimental value [10].

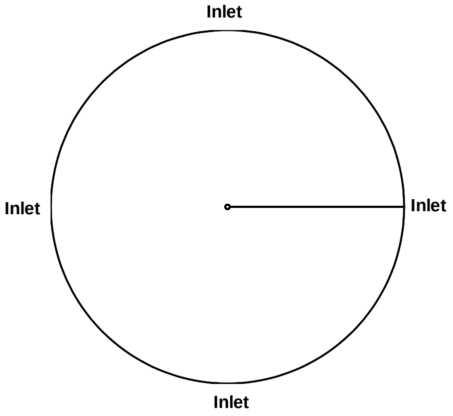


***Figure 11**. Single block O-grid for circular cylinder case*

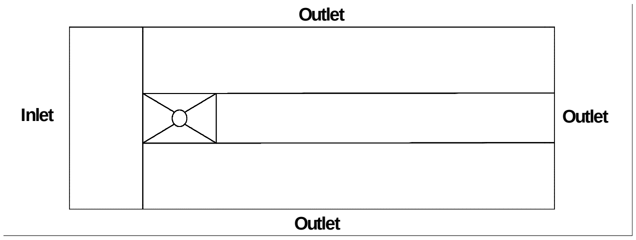


***Figure 12**. Multi-block grid for circular cylinder case*

***Table 2**. Impact of different type of grid on Strouhal number*

| Case | Strouhal Number |
|---|---|
| Singe-block grid | 0.163 |
| Multi-block grid | 0.164 |
| Experiment | 0.164 |

The periodic behavior of the cross-stream velocity component (v) obtained by probing at a single point downstream of the cylinder is shown in Figure 13. The v-component of the velocity alternates due to the presence of vortex shedding behind the cylinder, forming the well-known von Karman vortex street as shown in Figure 14.

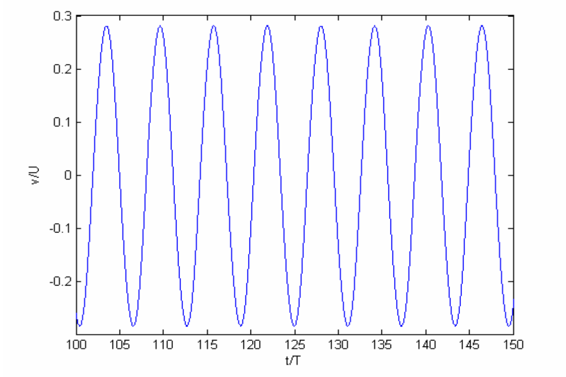


***Figure 13**. Periodic oscillation of the cross-stream component of velocity (v).*

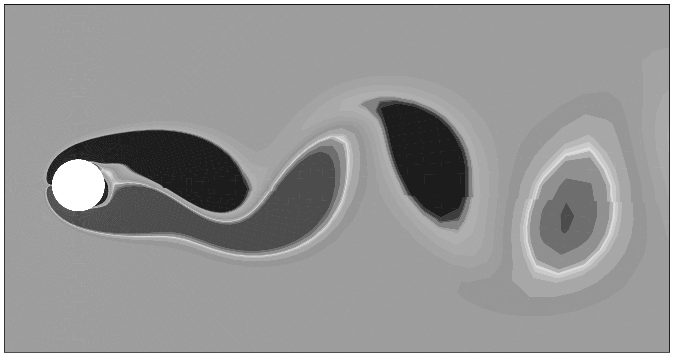


***Figure 14**. Snapshot of the vortex shedding downstream of the cylinder*

### 3.5 Flow Through a Nozzle with Heat Transfer

This case is chosen to assess *Loci*-STREAM for compressible turbulent flows with wall heat transfer using available experimental results [11]. The flow through a supersonic nozzle with heat transfer at the wall is studied here for different grid configurations. It results in a turbulent, compressible flow situation with heat transfer at the wall. The nozzle has a throat diameter of 0.0458 meters and an exit diameter of 0.0744 meters. The constant area duct before entering the converging-diverging nozzle has a length of 0.4572 meters and a diameter of 0.1275 meters. The total pressure and temperature are specified at the inlet along with extrapolated pressure outlet. The temperature is specified at the upper wall, corresponding to half the value at the inlet. The nozzle geometry and boundary conditions are shown in Figure 15. The exit Mach number for this case is 2.5.

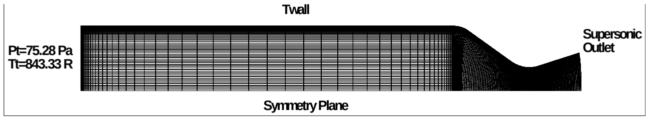


***Figure 15**. Nozzle geometry along with boundary conditions.*

Three different grids were constructed to vary the value of y+ near the heat transfer wall. The grids used had y+ values of 1, 20 and 100, corresponding to grid sizes of 161 x 69, 121 x 53 and 71 x 22, respectively. All cases were run using Menter's SST model with wall functions and second order upwind scheme for turbulence inviscid flux and venkatakrishnan limiters. The wall heat transfer is shown in Figure 16 for all the cases and compared to the experimental observation [11] as well. Some observations from the plot are as follows:

- The $y^+$=20 grid with wall functions case produced the most accurate results compared to experiment

- The $y^+=1$ grid with wall functions seems to under-predict the experimental wall heat transfer distribution
- The $y^+=100$ grid with wall functions seems to overshoot the experimental heat transfer distribution

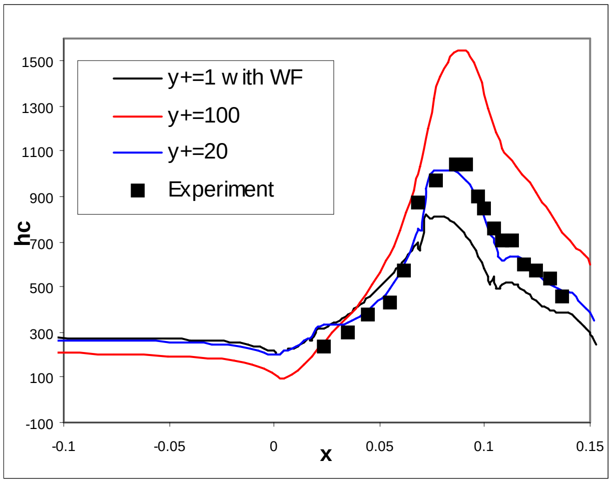


***Figure 16***. *Effect of wall spacing on heat transfer distribution using the SST turbulence model*

### 3.6 Turbulent Flow in a Hydraulic Turbine Draft Tube

The last case presented in this paper is an engineering application, namely, turbulent flow inside a hydraulic turbine draft tube. The efficiency of a hydraulic reaction turbine is significantly affected by the performance of the draft tube. Its main purpose is to utilize the kinetic energy leaving the turbine runner by converting it into pressure energy and thereby creating an additional head. The design of the draft tube is, however, not so straight forward since the flow is very complex and includes many flow features such as unsteadiness, turbulence, swirl, vortex rope, adverse pressure gradients, separation and secondary flows. All these phenomena interact and make the numerical flow prediction very difficult to achieve. This geometry has been investigated in two ERCOFTAC workshops (1999, 2001) [12] and experimental data is available [13]. The results were compared to existing experimental values. The geometry of the draft with its surface mesh along with the computational setup is shown in Figure 17. Steady state RANS computations were performed for this geometry. Computations were performed on 2 grids: a) 1 million points grid with y+ of 1 and b) 1 million points grid with a y+ of 50. Boundary conditions at the inlet include specifying axi-symmetric velocity components and initial values of $k$ and $\omega$ specified by the workshop. Also, the hub runner velocity, based on runner speed of 595 rpm, is specified as boundary conditions. A fixed outlet pressure of 0 is used at the outlet boundary.

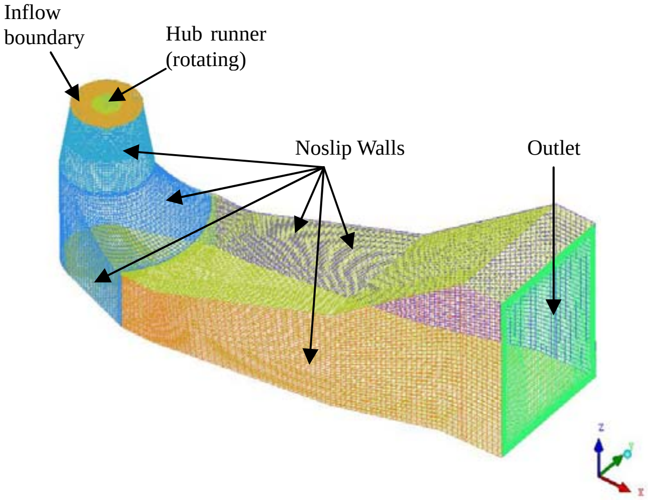


***Figure 17***. *Draft tube geometry along with computational setup*

Results from our code were compared with the commercially available CFX-5 solver [14]. The CFX-5 solver solves the governing equations with a finite element based finite volume method, applied on an unstructured grid. The discretization of the pressure gradient term and the diffusion term are obtained with shape functions, while the discretization of the advection term is usually specified with a blend factor, $\beta$, which determines the level of correction for the upwinding scheme.

Certain engineering quantities investigated in this work, as suggested by the organizers of the workshop, are as follows:

(a) The pressure recovery factor, $C_{pr}$:

$$C_{pr} = \frac{\overline{p_{Out:Wall}} - \overline{p_{In:Wall}}}{\frac{1}{2}\rho\left(\frac{Q_{in}}{A_{in}}\right)^2}, \tag{13}$$

(b) The mean pressure recovery factor, $C_{prm}$:

$$C_{prm} = \frac{\frac{1}{A_{out}}\iint_{A_{out}} p\,dA - \frac{1}{A_{in}}\iint_{A_{in}} p\,dA}{\frac{1}{2}\rho\left(\frac{Q_{in}}{A_{in}}\right)^2} \tag{14}$$

(c) The energy loss coefficient, $\zeta$:

$$\zeta = \frac{\iint_{A_{in}} p_{tot}\, \mathbf{u}\cdot\mathbf{n}\, dA + \iint_{A_{out}} p_{tot}\, \mathbf{u}\cdot\mathbf{n}\, dA}{\left| \iint_{A_{in}} p_{dyn}\, \mathbf{u}\cdot\mathbf{n}\, dA \right|}. \qquad (15)$$

(d) The pressure coefficient $c_p$, defined as:

$$c_p = \frac{p - \overline{p_{In:Wall}}}{\frac{1}{2}\rho\left(\frac{Q_{in}}{A_{in}}\right)^2}, \qquad (16)$$

where $p$ is the pressure, $\overline{p_{Out:Wall}}$ is the outlet averaged static wall pressure, $\overline{p_{In:Wall}}$ is the inlet averaged static wall pressure, $\rho$ is the density, $Q$ is the flow rate, $A_{in}$ is the area of inlet and $A_{out}$ refers to the outlet area.

The main flow structures obtained agreed well with experimental results for both grids (1 million with y+=1 and 1 million with y+=50) when second order accurate schemes were used for the turbulence equations. The two codes captured regions of separated flow near the runner hub and in the elbow corner. Also, the secondary flow with two main vortices and a vortex core located at the right side seen upstream (see Figure 18) were in good agreement with prior numerical results.

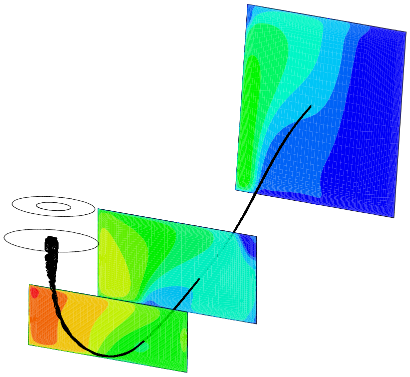
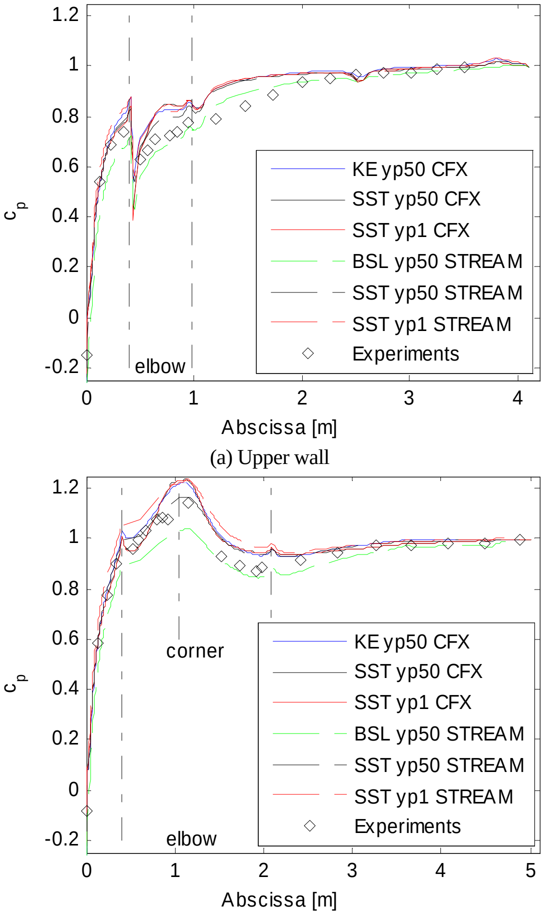


***Figure 18**. Calculated streamlines from the runner and velocity contours for the steady RANS calculations.*

When comparing the engineering quantities, it was observed that the *Loci*-STREAM code with the SST turbulence model was closest to the experimental value of the pressure recovery factor, $C_{pr}$, as shown in Table 3. Also, the code performed slightly better at the elbow compared to CFX when using the SST turbulence model and the y+=1 grid while comparing the pressure coefficient, $c_p$, at the central lines to experimental values (see Figure 19). The scatter in the engineering quantities for the SST turbulence model between y+=50 and y+=1 was however larger for the STREAM code and almost negligible for the CFX code.

***Table 3**. Engineering quantities for the steady RANS calculations.*

| Case | Model | Code | $C_{pr}$ | $C_{prm}$ | $\zeta$ |
|---|---|---|---|---|---|
| 1 | SST<br>yp50 | CFX<br>STREAM | 1.219<br>1.174 | 0.884<br>0.934 | 0.176<br>0.237 |
| 1 | SST<br>yp1 | CFX<br>STREAM | 1.218<br>1.113 | 0.885<br>0.873 | 0.175<br>0.207 |
| *Experiments* | | | *1.120* | - | - |



(a) Upper wall

(b) Lower wall

***Figure 19**. Pressure coefficient $c_p$ along the centerline for the steady RANS calculations ($c_p$=1 at outlet).*

## 4. SUMMARY

A new CFD code called *Loci*-STREAM is assessed in this paper using a wide range of test cases, including incompressible laminar and turbulent flow cases, inviscid compressible flow cases, compressible turbulent flows with wall heat transfer as well as internal turbulent flows in 3D geometries and unsteady computations. Validation of the *Loci*-STREAM with experimental and prior benchmark numerical results is done to validate the robustness of the code for flows ranging from incompressible to supersonic regimes. The scalability of *Loci*-STREAM on shared and distributed memory computers is also assessed. The results presented in this paper demonstrate the accuracy and efficiency of *Loci*-STREAM for a broad range of flow regimes. This has laid the foundation for further development of *Loci*-STREAM which involves the incorporation of finite rate chemistry for combusting flows, conjugate heat transfer capability, efficient time-stepping schemes based on the PISO algorithm, etc.

## ACKNOWLEDGEMENTS

This work was conducted as a part of an SBIR Phase II project awarded to Streamline Numerics, Inc. by NASA under contract number NNM05AA87C with Dr. Jeff West as the project monitor. Partial funding was also provided by NASA's Constellation University Institute Program (CUIP) program, with Ms. Claudia Meyer as program monitor. The authors appreciate the technical discussions with Jeff West and Kevin Tucker of NASA Marshall Space Flight Center. Technical support from Ed Luke of the Mississippi State University on *Loci* issues is also gratefully acknowledged.

## REFERENCES

[1] Luke, E.A. (1999) LOCI: A Deductive Framework for Graph-Based Algorithms, *Third International Symposium on Computing in Object-Oriented Parallel Environments*, edited by S. Matsuoka, R. Oldehoeft and M. Tholburn, No. 1732 in Lecture Notes in Computer Science, Springer-Verlag, pp 142-153.

[2] Rhie, C. M. and Chow, W. L. (1983) Numerical Study of the Turbulent Flow Past an Airfoil with Trailing Edge Separation, *AIAA J.*, v. 21, n. 11, pp. 1525-1535

[3] Wright, J. and Thakur, S. (2005) *Loci*-STREAM: All-Speed CFD Solver for Arbitrary Polygonal Meshes in the *Loci* Framework, Streamline Numerics, Inc., Gainesville, Florida, U.S.A. (www.snumerics.com)

[4] Venkatakrishnan, V. (1995) Convergence to steady state solutions of the Euler equations on unstructured grids with limiters,. *J. Comput. Phys.*, v. 118, pp.120 –130.

[5] Menter F.R. (1994) Two-Equation Eddy-Viscosity Turbulence Models for Engineering Applications, *AIAA Journal*, Vol. 32, No. 8, pp. 1598-1605.

[6] Nichols, R. H. and Nelson, C. C. (2004), Wall Function Boundary Conditions Including Heat Transfer and Compressibility, *AIAA J.,* v. 42, n. 6, pp. 1107-1114.

[7] Ghia, U., Ghia, K. N., and Shin, C. T. (1982) High-Re solutions for incompressible flow using the Navier-Stokes equations and a multi-grid method, *J.Comp.Phys.*, v.48, pp.387-411

[8] Driver, D. M. and Seegmiller, H. L. (1985) Features of a Reattaching Turbulent Shear Layer in Divergent Channel Flow, *AIAA J.*, v. 23, n. 2, pp. 163-171.

[9] Ferziger, J. H. and Peric, M. (1998). *Computational methods for fluid dynamics*. Springer Verlag, Berlin Heidelberg.

[10] Williamson, C.H.K. (1989) Oblique and Parallel Mode of Vortex Shedding in the Wake of a Cylinder at Low Reynolds Number, *J. Fluid Mech.* v. 20, n**.** 6, pp. 579-627.

[11] Back, L. H., Massier, P. F., and Gier, J. L. (1964) Convective Heat Transfer in a Convergent-Divergent Nozzle, *Int. J. of Heat and Mass Transfer,* v. 7, pp. 549-568.

[12] *The IAHR Workshops on Draft Tube Flow* (1999, 2001, 2005) URL http://www.turbine-99.org

[13] Andersson U. (2005) Test Case T – Some New Results and Updates Since Workshop I, *Proceedings of Turbine 99 – Workshop 2: The second ERCOFTAC workshop on draft tube flow*, URL http://www.turbine-99.org

[14] CFX® (Copyright © 1996-2005), Version 10, ANSYS Europe Ltd.