48th AIAA/ASME/SAE/ASEE Joint Propulsion Conference and Exhibit
July 30-Aug 1, 2012 / Atlanta, GA

Development of a Robust and Efficient Parallel Solver for Unsteady Turbomachinery Flows

Jeffrey Wright and Siddharth Thakur
Streamline Numerics, Inc., Gainesville, FL, 32609, U.S.A.

Ed Luke
Mississippi State University

Nathan Grinstead
GaN Corporation

Jeff West
NASA Marshall Space Flight Center

I. Introduction

THE traditional design and analysis practice for advanced propulsion systems relies heavily on expensive full-scale prototype development and testing. Over the past decade, use of high-fidelity analysis and design tools such as CFD early in the product development cycle has been identified as one way to alleviate testing costs and to develop these devices better, faster and cheaper. In the design of advanced propulsion systems, CFD plays a major role in defining the required performance over the entire flight regime, as well as in testing the sensitivity of the design to the different modes of operation. Increased emphasis is being placed on developing and applying CFD models to simulate the flow field environments and performance of advanced propulsion systems. This necessitates the development of next generation computational tools which can be used effectively and reliably in a design environment.

The turbomachinery simulation capability presented here is being developed in a computational tool called Loci-STREAM [1]. It integrates proven numerical methods for generalized grids and state-of-the-art physical models in a novel rule-based programming framework called *Loci* [2] 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 and complex multidisciplinary physics. An immediate application of interest is simulation of unsteady flows in rocket turbopumps, particularly in cryogenic liquid rocket engines. The key components of the overall methodology presented in this paper are the following:

(a) high fidelity unsteady simulation capability based on Detached Eddy Simulation (DES) in conjunction with second-order temporal discretization,

(b) compliance with Geometric Conservation Law (GCL) in order to maintain conservative property on moving meshes for second-order time-stepping scheme,

(c) a novel cloud-of-points interpolation method (based on a fast parallel kd-tree search algorithm) for interfaces between turbomachinery components in relative motion which is demonstrated to be highly scalable, and

(d) demonstrated accuracy and parallel scalability on large grids (~250 million cells) in full turbomachinery geometries.

II. Rule-Based Framework: Loci

The framework for application development called *Loci* [2] 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. 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. Applications in Loci are written 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 interesting 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.

III. Computational Algorithm and Governing Equations

The computational tool used in the present work is called Loci-STREAM. It is a second order accurate, pressure-based, Reynolds-averaged Navier-Stokes (RANS) code for unstructured grids, and is designed to handle all-speed flows (incompressible to supersonic). The flow solver is based on the SIMPLE (Semi-Implicit Method for Pressure-Linked Equations) algorithm [3]. It uses a control volume approach with a collocated arrangement for the velocity components and the scalar variables like pressure. Pressure-velocity decoupling is prevented by employing the momentum interpolation approach; this involves adding a fourth-order pressure dissipation term while estimating the mass flux at the control volume interfaces [4]. The velocity components are computed from the respective momentum equations. The velocity and the pressure fields are corrected using a pressure correction (p') equation. The correction procedure leads to a continuity-satisfying velocity field. The whole process is repeated until the desired convergence is reached [5]. The additional equations such as the enthalpy equation, species transport equation, and turbulence model equations are solved subsequently.

A. Favre-Averaged Governing Equations

The Favre-averaged Navier-Stokes equations of mass continuity, momentum, and energy are:

$$\frac{\partial \bar{\rho}}{\partial t} + \frac{\partial \bar{\rho} u_j}{\partial x_j} = 0 \tag{1}$$

$$\begin{aligned} &\frac{\partial \bar{\rho} u_i}{\partial t} + \frac{\partial \bar{\rho} u_j u_i}{\partial x_j} = -\frac{\partial \bar{p}}{\partial x_i} + \frac{\partial}{\partial x_j}\left(\tilde{\tau}_{ij} - \overline{\rho u_i'' u_j''}\right) \\ &\tilde{\tau}_{ij} = \mu\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right) - \frac{2}{3}\mu\frac{\partial u_k}{\partial x_k}\delta_{ij} \\ &-\overline{\rho u_i'' u_j''} = \mu_t\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right) - \frac{2}{3}\mu_t\frac{\partial u_k}{\partial x_k}\delta_{ij} - \frac{2}{3}\bar{\rho}k\delta_{ij} \end{aligned} \tag{2}$$

$$\begin{aligned} &\frac{\partial}{\partial t}\left(\bar{\rho}H - \bar{p}\right) + \frac{\partial}{\partial x_j}\left(\bar{\rho} u_j H\right) = \frac{\partial}{\partial x_j}\left[\left(\frac{\mu}{Pr} + \frac{\mu_t}{Pr_t}\right)\frac{\partial h}{\partial x_j}\right] + \frac{\partial}{\partial x_j}\left(\rho\sum_{k=1}^{NS} h_k D_k \frac{\partial Y_k}{\partial x_j}\right) \\ &+ \frac{\partial}{\partial x_j}\left[u_j\left(\mu + \mu_t\right)\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} - \frac{2}{3}\frac{\partial u_k}{\partial x_k}\delta_{ij}\right)\right] + \frac{\partial}{\partial x_j}\left[\left(\mu + \frac{\mu_t}{\sigma_k}\right)\frac{\partial k}{\partial x_j}\right] \end{aligned} \tag{3}$$

B. Turbulence Model

Menter's Shear Stress Transport (SST) model [6] is the primary turbulence model in Loci-STREAM. The SST model essentially reduces to the $k-\varepsilon$ model near solid walls and transitions to $k-\omega$ model away from the walls with the help of a blending function. Details of the model are given below. The averaging symbols will be dropped for simplicity.

Kinematic Eddy Viscosity:

$$\nu_t = \frac{a_1 k}{\max(a_1\omega, \Omega F_{SST})} \tag{4}$$

where Ω is the absolute value of the vorticity, $a_1 = 0.31$, and the blending function F_{SST} is given by:

$$\mathrm{F_{SST}} = \tanh(\arg_{\mathrm{SST}}^2) \tag{5}$$

where

$$\arg_{SST}^2 = \max\left(2\frac{\sqrt{k}}{0.09\,\omega y}, \frac{500\nu}{y^2\omega}\right) \tag{6}$$

Turbulent Stress Tensor:

$$\tau_{ij}^{'} = \mu_t(\partial_j u_i + \partial_i u_j) - \frac{2}{3}(\mu_t \partial_l u_l + \rho k)\delta_{ij} \tag{7}$$

Turbulent Kinetic Energy Equation:

$$\partial_t(\rho k) + u_l \partial_l(\rho k) = \tau_{ij}^{'} \partial_j u_i - \beta^* \rho \omega k + \partial_j\left[(\mu + \mu_t \sigma_k)\partial_j k\right] \tag{8}$$

Turbulent Dissipation Equation:

$$\partial_t(\rho\omega) + u_l \partial_l(\rho\omega) = \frac{\gamma}{\nu_t}\tau_{ij}^{'}\partial_j u_i - \beta\rho\omega^2 + \partial_j\left[(\mu + \mu_t\sigma_\omega)\partial_j\omega\right] + 2(1 - F_{SST})\rho\sigma_{\omega 2}\frac{1}{\omega}\partial_j k \partial_j \omega \tag{9}$$

Coefficients:

$k-\omega$ and $k-\varepsilon$ model coefficients are blended as:

$$\phi = F_{SST}\phi_{k\omega} + (1 - F_{SST})\phi_{k\varepsilon} \tag{10}$$

where $\phi_{k\omega}$ are:

$$\begin{aligned} &\sigma_{k1} = 0.85,\ \sigma_{\omega 1} = 0.5,\ \beta_1 = 0.075,\ \beta^* = 0.09,\ \kappa = 0.41, \\ &\gamma_1 = \beta_1/\beta^* - \sigma_{\omega 1}\kappa^2/\sqrt{\beta^*} \end{aligned} \tag{11}$$

and $\phi_{k\varepsilon}$ are:

$$\begin{aligned} &\sigma_{k2} = 1.0,\ \sigma_{\omega 2} = 0.856,\ \beta_2 = 0.0828,\ \beta^* = 0.09,\ \kappa = 0.41, \\ &\gamma_2 = \beta_2/\beta^* - \sigma_{\omega 2}\kappa^2/\sqrt{\beta^*} \end{aligned} \tag{12}$$

C. Detached Eddy Simulation (DES)

Detached-Eddy Simulation (DES) was first proposed in 1997 (Spalart et al. [7]). Since then been it has used for many turbulent flow applications and undergone several modifications (Strelets [8], Menter and Kuntz [9], Spalart et al. [10], Shur et al. [11], Spalart [12]). DES is defined by it originators as "a three-dimensional unsteady solution using a single turbulence model, which functions as a sub-grid-scale model in regions where the grid density is fine enough for a large-eddy simulation, and as a Reynolds-averaged model in regions where it is not" (Travin et al. [13]). So, the "LES mode" is activated where grid spacing in all directions is much smaller than the turbulent boundary layer thickness. DES senses the grid density and adjusts to a lower level of eddy viscosity (relative to RANS), thus allowing large-scale instabilities in the flow to occur. In other regions (essentially boundary layers) DES is in the RANS mode though the solution is unsteady in these regions also. Thus, there is a smooth transition between the different regions being modeled. DES was originally developed using the Spalart-Allmaras model as the underlying turbulence model (Spalart et al. [7]) but has since then been extended to Menter's SST model (Strelets [8], Menter et al. [14]).

The idea of DES is applied to the SST model by switching from the SST-RANS model to an LES model based on the turbulent length scale predicted by the SST model which is given by $l_{k-\omega} = k^{1/2}/\beta^*\omega$. This length scale is replaced by the DES length scale based on the grid spacing Δ. Thus, the destruction term in the turbulent kinetic energy equation can be written as:

$$\beta^* k\omega = \frac{k^{3/2}}{l_{k-\omega}} \xrightarrow{\text{replaced by}} \frac{k^{3/2}}{\delta} \tag{13}$$

where

$$\delta = \min\left(l_{k-\omega}, C_{DES}\Delta\right) \tag{14}$$

$$\Delta = \max\left(\Delta_i\right) \tag{15}$$

with C_{DES} equal to 0.78 in the wall region and 0.61 in the outer part of the flow. The implementation of the kinetic energy equation used in the SST model can be easily modified for the DES formulation as follows:

$$\frac{\partial}{\partial t}(\rho k)+\frac{\partial}{\partial x_i}(\rho u_i k)=\frac{\partial}{\partial x_i}\left[(\mu+\sigma_k\mu_t)\frac{\partial k}{\partial x_i}\right]+\tilde{P}_k-\rho\beta^* k\omega.F_{DES} \tag{16}$$

$$F_{DES}=\max\left(\frac{l_{k-\omega}}{C_{DES}\Delta},1\right);\quad l_{k-\omega}=\frac{k^{1/2}}{\beta^*\omega} \tag{17}$$

A recognized problem with the DES is that there is no mechanism to prevent the DES limiter to become activated in an attached boundary layer. Thus, for fine grids, the transition from RANS mode to DES mode can happen inside the boundary layer. This can happen when the local grid spacing near a boundary is smaller than the boundary layer thickness. This can commonly occur with the use of unstructured grids and can lead to what is known as grid-induced separation (Menter et al. [14], Menter and Kuntz [9]). A solution to this has been provided by Menter et al. [14] which results in the so-called Delayed Detached-Eddy Simulation (DDES). It involves shielding the boundary layer from the DES limiter, thus avoiding grid-induced separation. The blending functions of the SST model are used to formulate a zonal DES limiter as follows:

$$F_{DES}\xrightarrow{replaced\ by}F_{DDES}=\max\left(\frac{l_{k-\omega}}{C_{DES}\Delta}(1-F_{SST}),1\right) \tag{18}$$

with $F_{SST}=0,\ F_1\ or\ F_2$. Using a value of 0 yields the original SST-DES model of Strelets [8]. Menter et al. [14] recommend using F_2 which shields most of the boundary layer.

D. Geometric Conservation Law (GCL)

In order to guarantee discrete conservation, the so-called geometric conservation law (GCL) must be satisfied [15-21]. It can be demonstrated that the satisfaction of the geometric conservation law is important for computations involving moving meshes by considering a uniform steady-state flow. Conservative finite-volume schemes with a fixed mesh have the property that a uniform flow is an exact solution of the numerical algorithm. For a moving mesh, the uniform flow is preserved only if the geometric conservation law is discretely satisfied. This is true even if the mesh moves rigidly and the cell areas are not changing geometrically. In the special case of uniform flow, the governing equation reduces to a purely geometric statement relating the time-dependent control volume, the control surface, the control surface velocity and the unit normal. The consistent treatment of these time-dependent geometric quantities is referred to as the geometric conservation law which may be regarded as an identity that must be satisfied, either explicitly or implicitly, if the conservative property is to be maintained. In essence, the GCL corresponds to the statement that no disturbances should be introduced by any arbitrary mesh motion for a uniform flow.

In the present work, we have implemented the GCL-compliant formulation of the second-order backward differencing (BDF2) scheme, given only the knowledge of the mesh point positions at discrete time levels. When the GCL-compliant control surface velocity and normal vectors are employed in the assembly of the convective flux, the method is geometrically conservative and is second-order-accurate in time on an arbitrarily moving mesh.

To illustrate the formulation of a GCL-compliant scheme in the present work, let us begin with the Navier-Stokes equations in conservative form:

$$\frac{\partial \boldsymbol{Q}}{\partial t}+\nabla\cdot\left(\boldsymbol{F}(\boldsymbol{Q})+\boldsymbol{G}(\boldsymbol{Q})\right)=0 \tag{19}$$

where $\boldsymbol{Q}$ is the vector of conserved variables (mass, momentum, energy), $\boldsymbol{F}(\boldsymbol{Q})$ represents convective fluxes and $\boldsymbol{G}(\boldsymbol{Q})$ represents viscous fluxes. Integration over a moving control volume Ω with boundary (surface) $\partial\Omega$ yields

$$\int_{\Omega(t)}\frac{\partial \boldsymbol{Q}}{\partial t}d\Omega+\int_{\partial\Omega(t)}\boldsymbol{F}(\boldsymbol{Q})\cdot\hat{n}\,dA+\int_{\partial\Omega(t)}\boldsymbol{G}(\boldsymbol{Q})\cdot\hat{n}\,dA=0 \tag{20}$$

Using the Leibnitz identity

$$\frac{d}{dt}\int_{\Omega(t)}\boldsymbol{Q}\,d\Omega=\int_{\Omega(t)}\frac{\partial \boldsymbol{Q}}{\partial t}d\Omega+\int_{\partial\Omega(t)}\boldsymbol{Q}(\dot{\boldsymbol{x}}\cdot\hat{n})\,dA \tag{21}$$

where $\dot{\boldsymbol{x}}$ and $\hat{n}$ are the velocity and the unit normal of the moving interface $\partial\Omega(t)$, we can write Eq. (20) as

$$\frac{d}{dt}\int_{\Omega(t)} \boldsymbol{Q} d\Omega + \int_{\partial\Omega(t)} \left(\boldsymbol{F}(\boldsymbol{Q}) - \boldsymbol{Q}\dot{\boldsymbol{x}}\right)\cdot\hat{n}\, dA + \int_{\partial\Omega(t)} \boldsymbol{G}(\boldsymbol{Q})\cdot\hat{n}\, dA = 0 \tag{22}$$

Considering $\boldsymbol{Q}$ as cell-averaged quantities, and neglecting the viscous fluxes, the semi-discrete version of Eq. (22) can be written as follows:

$$\frac{d}{dt}(\boldsymbol{Q}\Omega) + \sum_{f=1}^{nfaces} \int_f \left[\boldsymbol{F}(\boldsymbol{Q}) - \boldsymbol{Q}\dot{\boldsymbol{x}}\right]\cdot\hat{n}\, dA = 0 \tag{23}$$

Now considering $\boldsymbol{F}$ and $\boldsymbol{Q}$ as mean quantities over each face, we can further write

$$\frac{d}{dt}(\boldsymbol{Q}\Omega) + \sum_{f=1}^{nfaces} \boldsymbol{F}_f \cdot \hat{n}_f A_f - \sum_{f=1}^{nfaces} \boldsymbol{Q}_f\, \dot{\boldsymbol{x}}_f \cdot \hat{n}_f A_f = 0 \tag{24}$$

We now introduce two geometrical quantities, namely, $\boldsymbol{\eta}$ and λ. The quantity $\boldsymbol{\eta}$ can be interpreted as the mean value of the integral of the face normal area over the time interval Δt:

$$\boldsymbol{\eta} = \int_{\partial\Omega(t)} \hat{n}\, dA \tag{25}$$

The quantity λ can be interpreted as the integrated normal grid velocity on a face over the time interval Δt:

$$\lambda = \frac{1}{\|\boldsymbol{\eta}\|}\int_{\partial\Omega(t)} \dot{\boldsymbol{x}}\cdot\hat{n}\, dA \tag{26}$$

We now introduce a new parameter σ:

$$\sigma = \lambda.\|\boldsymbol{\eta}\| = \int_{\partial\Omega(t)} \dot{\boldsymbol{x}}\cdot\hat{n}\, dA \tag{27}$$

Using the two parameters $\boldsymbol{\eta}$ and σ, Eq. (24) can be written as

$$\frac{d}{dt}(\boldsymbol{Q}\Omega) + \sum_{f=1}^{nfaces} \left(\boldsymbol{F}_f \cdot \boldsymbol{\eta}_f - \boldsymbol{Q}_f\, \sigma_f\right) = 0 \tag{28}$$

For a stationary mesh, it is straightforward to compute these geometrical quantities. For the numerical formulation of the fluxes for a moving mesh, the crux of a GCL-compliant scheme lies in the appropriate estimation of the two geometrical quantities $\boldsymbol{\eta}$ and σ. To understand the reasoning behind this, let us examine the significance and the implications of the GCL next.

The original statement of the GCL was introduced by Thomas and Lombard [15]. The discrete geometric conservation law requires that the state $\boldsymbol{Q} = constant$ be an exact solution of equation (22). In this case, we have $\boldsymbol{G}(\boldsymbol{Q}) = 0$, since the viscous fluxes are based on gradients of $\boldsymbol{Q}$. Since the integral on a closed boundary of the flux of a constant function is identically zero, i.e.,

$$\int_{\partial\Omega(t)} \boldsymbol{F}(\boldsymbol{Q})\cdot\hat{n}\, dA = 0, \quad for\ \boldsymbol{F}(\boldsymbol{Q}) = const \tag{29}$$

it follows from Eq. (22), for this special case, that

$$\frac{d\Omega}{dt} - \int_{\partial\Omega(t)} \dot{\boldsymbol{x}}\cdot\hat{n}\, dA = 0 \tag{30}$$

which is the statement of the GCL. This basically states that the change in volume of each control volume between t^n and t^{n+1} must be equal to the volume swept by the cell boundary during $\Delta t = t^{n+1} - t^n$. The semi-discrete version of the above equation for a control volume containing *nfaces* number of faces can be written as

$$\frac{d\Omega}{dt} = \sum_{f=1}^{nfaces} \sigma_f = \sum_{f=1}^{nfaces} \|\boldsymbol{\eta}\|_f \cdot \left(\frac{\sigma}{\|\boldsymbol{\eta}\|}\right)_f \tag{31}$$

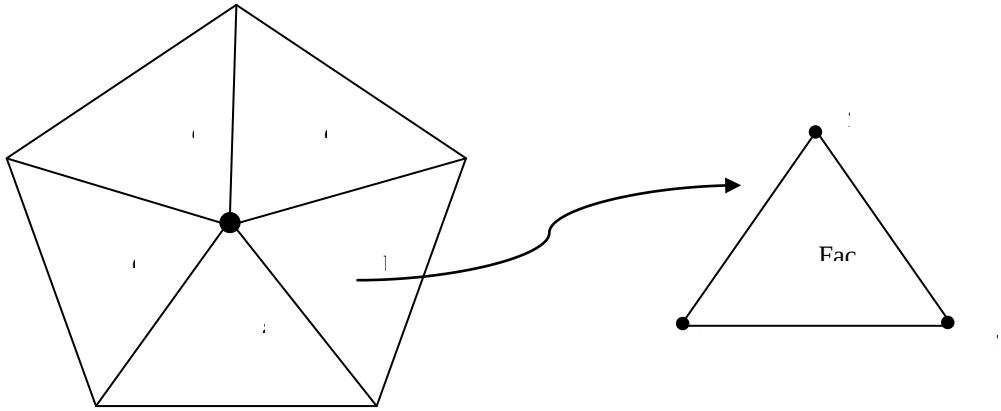


Figure 1. Splitting of a polygonal face into triangular facets: here a pentagon-shaped face is split into five triangular facets labeled a-e. A representative facet (labeled k) is shown on the right with its vertices numbered 1-3. Vertex numbered 3 is the centroid of the polygonal face.

Eq. (31) is a necessary condition for a numerical scheme to maintain a uniform initial field and hence the evaluation of the two geometrical quantities $\boldsymbol{\eta}$ and $\boldsymbol{\sigma}$ must be carefully conducted. In the following, we will describe a direct geometrical method to evaluate these two parameters for the case of a triangular facet moving in three-dimensional space such that Eq. (31) is satisfied. Since any face of a polygon can be split up into multiple non-overlapping triangular facets, as shown in Figure 1, this method can be used for a generalized unstructured grid.

An important assumption is that each node in the mesh moves in a linear manner between time levels n and n+1 ([16], [20]). Consider a triangular facet *k* shown in Figure 2. The volume swept over by this facet between time levels n and n+1 is given by the volume shown in Figure 2, which is given by

$$\delta\Omega_k^{n+1} \equiv \Omega_k^{n+1} - \Omega_k^n = \Delta t \int_k \left(\dot{\boldsymbol{x}}_k \cdot \hat{n}_k \; dA \right)^{n+1} = \Delta t . \sigma_k^{n+1} \tag{32}$$

The velocities at the vertices of the triangular facet (labeled 1, 2 and 3) are given by

$$\dot{\boldsymbol{x}}_1^{n+1} = \frac{\Delta \boldsymbol{x}_1^{n+1}}{\Delta t}, \;\; \dot{\boldsymbol{x}}_2^{n+1} = \frac{\Delta \boldsymbol{x}_2^{n+1}}{\Delta t}, \;\; \dot{\boldsymbol{x}}_3^{n+1} = \frac{\Delta \boldsymbol{x}_3^{n+1}}{\Delta t} \tag{33}$$

where

$$\Delta \boldsymbol{x}_1^{n+1} = \boldsymbol{x}_1^{n+1} - \boldsymbol{x}_1^n, \;\; \Delta \boldsymbol{x}_2^{n+1} = \boldsymbol{x}_2^{n+1} - \boldsymbol{x}_2^n, \;\; \Delta \boldsymbol{x}_3^{n+1} = \boldsymbol{x}_3^{n+1} - \boldsymbol{x}_3^n \tag{34}$$

Now the volume in Eq. (32) can be evaluated using the averaged-normal method as

$$\boldsymbol{\sigma}_k^{n+1} = \frac{1}{3} \left(\dot{\boldsymbol{x}}_1^{n+1} + \dot{\boldsymbol{x}}_2^{n+1} + \dot{\boldsymbol{x}}_3^{n+1} \right) \cdot \boldsymbol{\eta}_k^{n+1} \tag{35}$$

where

$$\boldsymbol{\eta}_k^{n+1} = \frac{1}{3} \left[\left(\hat{n}_k A_k \right)^{n+1} + \left(\hat{n}_k A_k \right)^n + \left(\hat{n}_k A_k \right)^* \right] \tag{36}$$

with

$$\left(\hat{n}_k A_k \right)^{n+1} = \tfrac{1}{2} \left(\boldsymbol{x}_2^{n+1} - \boldsymbol{x}_1^{n+1} \right) \times \left(\boldsymbol{x}_3^{n+1} - \boldsymbol{x}_1^{n+1} \right) \tag{37}$$

$$\left(\hat{n}_k A_k \right)^{n} = \tfrac{1}{2} \left(\boldsymbol{x}_2^{n} - \boldsymbol{x}_1^{n} \right) \times \left(\boldsymbol{x}_3^{n} - \boldsymbol{x}_1^{n} \right) \tag{38}$$

$$\left(\hat{n}_k A_k \right)^{*} = \tfrac{1}{2} \left[\tfrac{1}{2} \left(\boldsymbol{x}_2^{n+1} - \boldsymbol{x}_1^{n+1} \right) \times \left(\boldsymbol{x}_3^{n} - \boldsymbol{x}_1^{n} \right) + \tfrac{1}{2} \left(\boldsymbol{x}_2^{n} - \boldsymbol{x}_1^{n} \right) \times \left(\boldsymbol{x}_3^{n+1} - \boldsymbol{x}_1^{n+1} \right) \right] \tag{39}$$

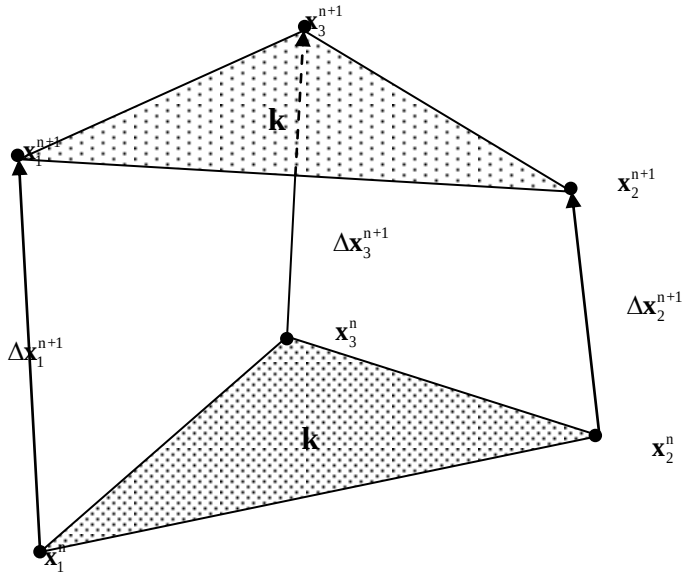


Figure 2. Nomenclature for the movement of a triangular facet from time level n to n+1.

For an arbitrary polygonal face of a control volume (see Figure 1), we simply need to sum over all the triangular facets comprising that face:

$$\boldsymbol{\eta}_f = \sum_{k=1}^{nfacets} \boldsymbol{\eta}_k , \quad \sigma_f = \sum_{k=1}^{nfacets} \sigma_k \tag{40}$$

Now when BDF2 is used, the semi-discrete GCL, Eq. (30), is written as

$$\frac{1}{\Delta t}\left(\tfrac{3}{2}\Omega^{n+1} - 2\Omega^{n} + \tfrac{1}{2}\Omega^{n-1}\right) = \int \dot{\boldsymbol{x}} \cdot \hat{n}\, dA \tag{41}$$

The left hand side of Eq. (41) can be written as $\frac{3}{2}\left(\Omega^{n+1} - \Omega^{n}\right) - \frac{1}{2}\left(\Omega^{n} - \Omega^{n-1}\right)$. Thus, using Eq. (32), we can write

$$\boldsymbol{\eta}^{BDF2} = \tfrac{3}{2}\boldsymbol{\eta}^{n+1} - \tfrac{1}{2}\boldsymbol{\eta}^{n} \tag{42}$$

$$\sigma^{BDF2} = \tfrac{3}{2}\sigma^{n+1} - \tfrac{1}{2}\sigma^{n} \tag{43}$$

Using the BDF2 scheme, the discretized equation, Eq. (28), finally becomes

$$\frac{1}{\Delta t}\left[\tfrac{3}{2}\left(\boldsymbol{Q}\Omega\right)^{n+1} - 2\left(\boldsymbol{Q}\Omega\right)^{n} + \tfrac{1}{2}\left(\boldsymbol{Q}\Omega\right)^{n-1}\right] + \sum_{f=1}^{nfaces}\left(\boldsymbol{F}_f \cdot \boldsymbol{\eta}_f^{BDF2} - \boldsymbol{Q}_f\, \sigma_f^{BDF2}\right) = 0 \tag{44}$$

IV. Overset Grid Methodology for Multi-Stage Turbomachinery Applications: Interface Interpolation Algorithm for Relative Motion

A general overset grid methodology has been developed to interface different grid components in relative motion. The overset grid methodology is very important for turbomachinery applications which involve coupled simulations of stationary and moving domains. The present unstructured overset technology is based on a novel cloud-of-points interpolation technique (to be described below) which is able to generate compact interpolation stencils from an arbitrary distribution of point data. The interpolator is used to reconstruct fluid fields described using the primitive variables of pressure, temperature, mixture mass fractions, and velocity. In Loci-STREAM, which is a pressure-based code, the interface boundary of each subgrid is treated as a fixed pressure boundary where

pressure is interpolated at the interface from the neighboring subgrids. The velocity at the interface is also interpolated from the neighboring subgrids. During the course of the iterative algorithm, the mass flux is not corrected at the interface.

The basis of our interface reconstruction algorithm is a cloud-of-points interpolation method. The cloud-of-points algorithm is based on a fast parallel kd-tree search algorithm. Using this algorithm we can build a stencil to interpolate for any point in space utilizing the nearest point from the 8 octants around the point. From these 8 nearest points we identify the best tetrahedra that contains the interpolation point and utilize a linear barycentric interpolation to create the interpolation stencil.

To interpolate at interfaces, we do not use the interpolation technique described above to directly compute the value at the face, as there will not be sufficient upwinding required for solution stability. Instead we construct a structured style second order one-dimensional stencil about each face with two upwind points located at $\boldsymbol{x}+\hat{n}\Delta h/2$ and $\boldsymbol{x}+\hat{n}3\Delta h/2$ and one downwind point at $\boldsymbol{x}-\hat{n}\Delta h/2$. Here $\vec{n}$ $\hat{n}$ is the face normal, $\boldsymbol{x}$ is the face centroid, and Δh is estimated by using the size of the stencil generated by the cloud-of-points interpolator at the interface. The resulting face value is computed by using a second-order extrapolation utilizing the Van Albada limiter.

To enable interpolated interfaces for use on parallel processing setting, the described interpolation procedure is implemented using a scalable parallel technique. In this technique, the input interpolants are distributed to processors using an Orthogonal Recursive Bisection (ORB) partitioning technique. This ensures that the data is distributed with high locality. Then, when values need to be interpolated, a bounding box is computed from the interpolant points and the maximum stencil size. Bounding boxes are shared with all processors and all points within the requested bounding boxes are sent to the appropriate processor. Once the points are communicated, then each processor computes the interpolating stencil, and then a second communication step sends only the data that is accessed by the interpolation stencils of each processor.

V. Results

A. Interpolated Interface Test Cases

To test the interpolated interface we computed an inviscid low speed flow over a ramp. For this simulation we included a non-matching interpolated interface where on one side a uniform mesh spacing was prescribed, while the other side of the interface utilized a spacing that varied from twice on the top of the domain to half at the bottom. The resulting test should give a good estimate of how well the interface coupling works with non-matching mesh resolution. Figure 3 shows the resulting subsonic flow pressure contours. As is evident from this figure, the non-matching interface is robustly and accurately resolved by the interpolation scheme.

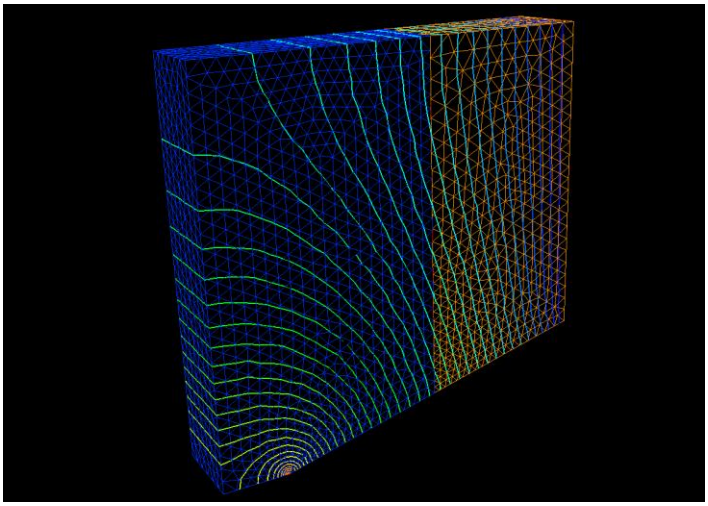


Figure 3: Subsonic flow over ramp with non-matching interface.

In theory the interpolated interface should also perform well when meshes are overlapping. To confirm this we ran a series flows over cylinders using overlapping meshes. Here we show an inviscid flow at low Mach number. The pressure contours from these simulations, shown in figure 4, illustrate that the contour lines in the overlap region nearly match perfectly for this case. In general, these tests show that the interpolated interface methodology is robust and accurate.

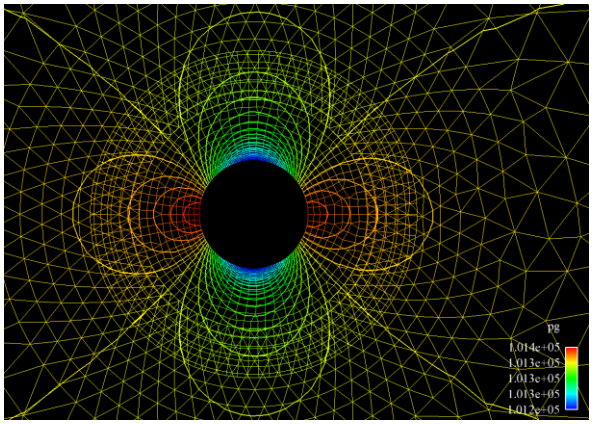


Figure 4. Pressure contours for overlapping subsonic mesh interpolation case.

As another test case for the overset capability in Loci-STREAM, a square driven cavity of size 1 by 1 unit at Reynolds number of 1000 (based on the lid velocity) was used. First, a single structured grid consisting of 81x81 grid points was used to compute the flow field inside the cavity and used as the benchmark solution. Next, an overset grid was created consisting of two unstructured subgrids. The lower subgrid goes from $0.0<y<0.55$, whereas the upper subgrid goes from $0.45<y<1.0$. Thus, the overlap region between the two subgrids is 0.1 unit wide. The second-order upwind (SOU) scheme was used for the computation. Figure 5 shows the mesh (with the two unstructured subgrids) and pressure contours. It can be seen that pressure contours are smooth and continuous across the interface. Figure 6 shows the comparison between the velocity profile along the centerline of the cavity obtained on the single gird and the overset grid. The two solutions match exactly.

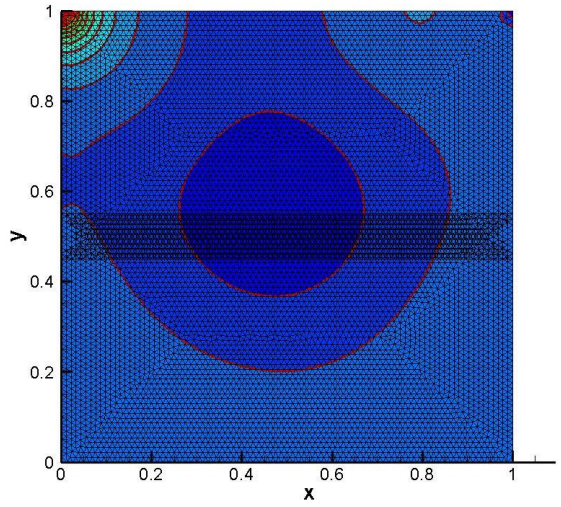


Figure 5. Overset grid with pressure contours.

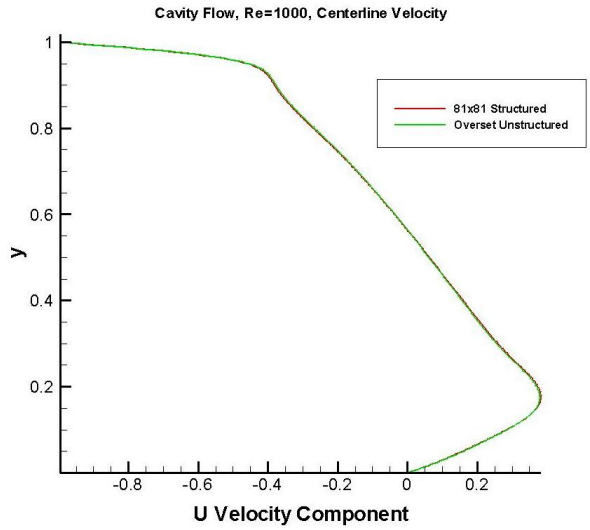
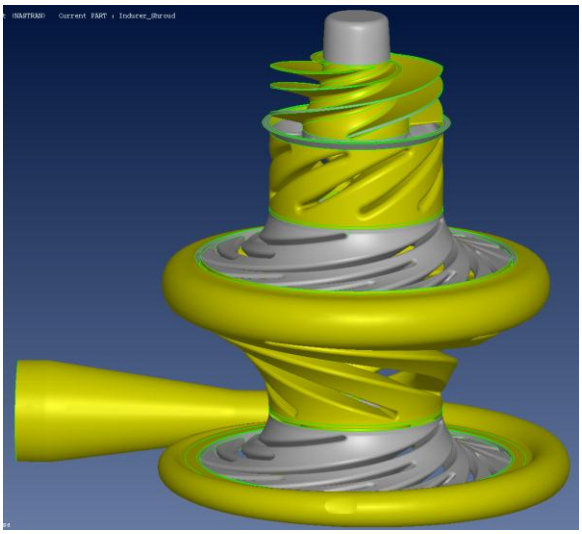


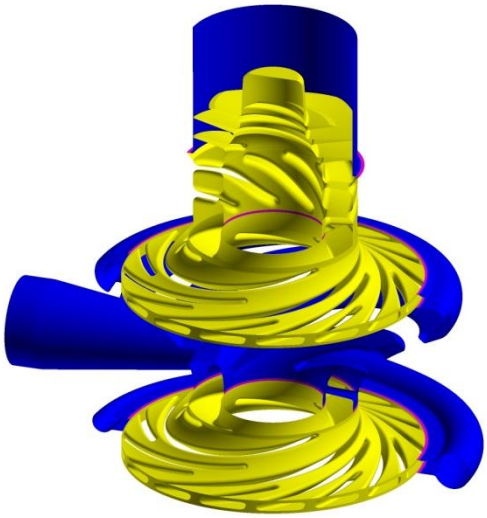
Figure 6. Comparison between single & overset grids.

B. Turbomachinery Flow Results

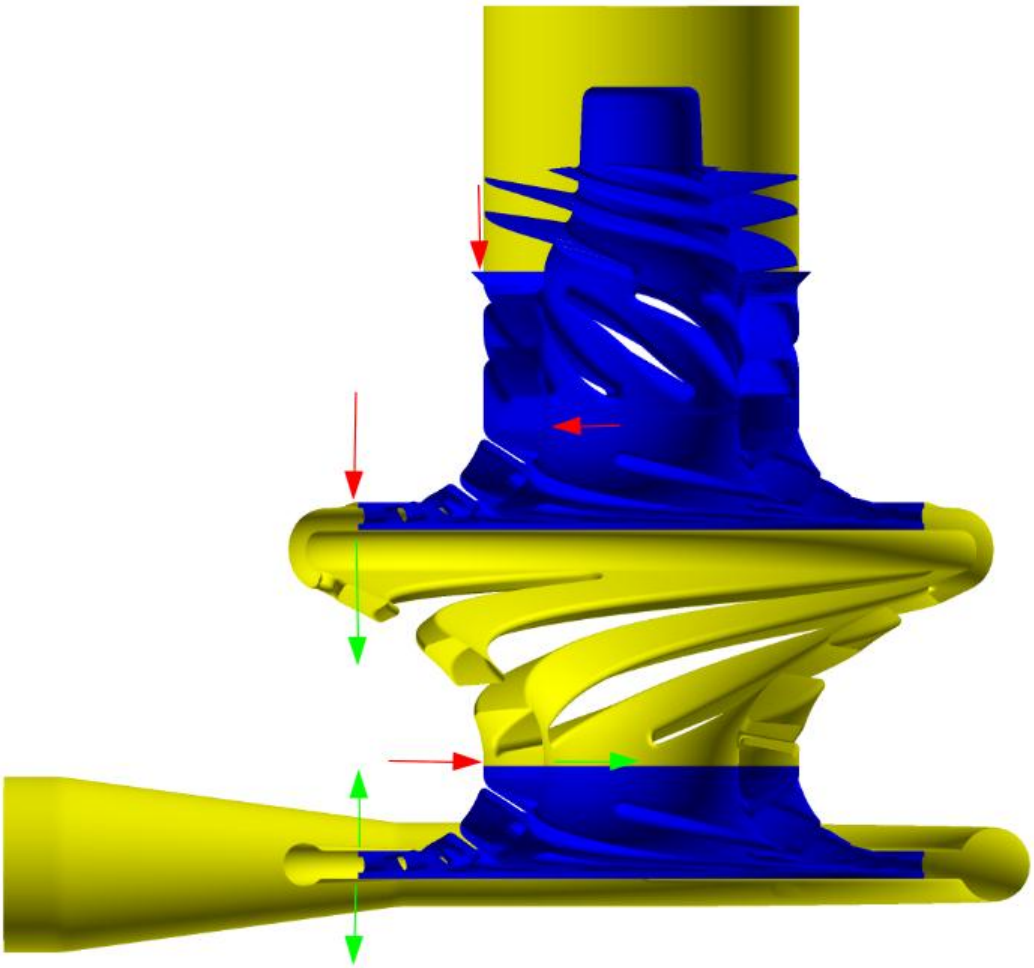
The Lunar Surface Access Module (LSAM) fuel pump is used as a testbed to demonstrate the turbomachinery simulation capability. This pump consists of a four bladed inducer, eight bladed kicker, 4+4+8 bladed impeller, five vaned crossover, 4+4+8 bladed impeller, and a single-tongue volute, all of which are included in the computational domain. There is no test data available at this time since the pump is in the assembly phase, but comparisons can be made to meanline predictions and other predictive tools to assess the accuracy of CFD simulations. The primary purpose of this case is to test the robustness of the turbomachinery simulation capability of Loci-STREAM in the presence of multiple inlets and outlets.



(a) Full geometry showing dimensioned edges.



(b) Overall leakage paths.

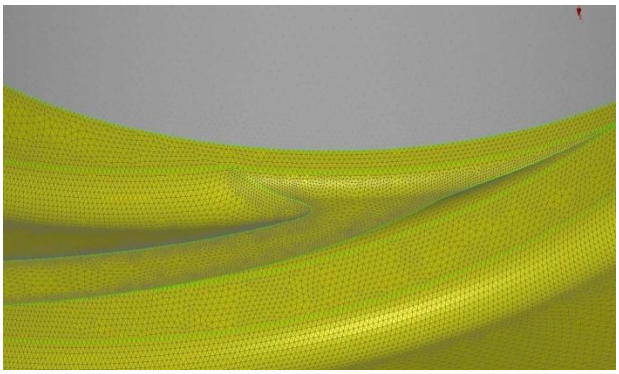


(c) Leakage locations: red arrows indicate inflow and green arrows indicate outflow.

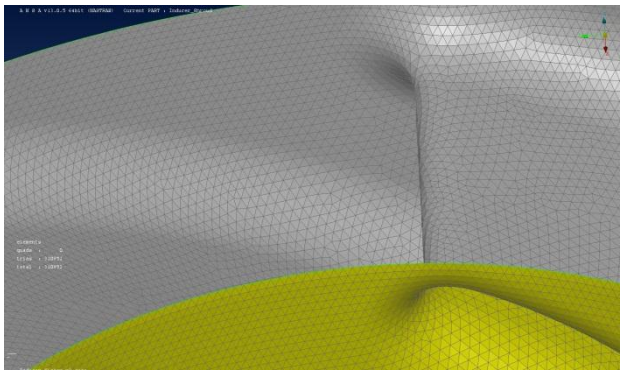
Figure 7. Pump geometry.

Figure 7 shows the geometric model of the LSAM with areas where leakage flows will enter or exit the model in a 360 degree boundary surface. Red indicates inflow and green indicates outflow. Yellow surfaces are non-rotating walls and blue surfaces are rotating walls. The inducer tunnel extends approximately half of one diameter upstream of the inducer nose to the inlet boundary of the mesh. The radial clearance between the inducer blades and the shroud is 0.007 inch and the inducer shroud was modeled as a smooth cylinder extending upstream from the front edge of the kicker shroud. The leakage paths were included in the surface mesh as flat smooth surfaces meshed just like the neighboring walls but when simulated, those leakages surfaces were set to behave first as walls, and then later as inflows and outflows without having to rebuild or alter the mesh. Each individual leakage path constitutes a significant percentage of the primary mass flow rate, ranging from as little as 3% for some and as high as 12% for another. With this in mind, the ability to include these flows in the main path is important to accurate prediction of the pump's fluid environment.

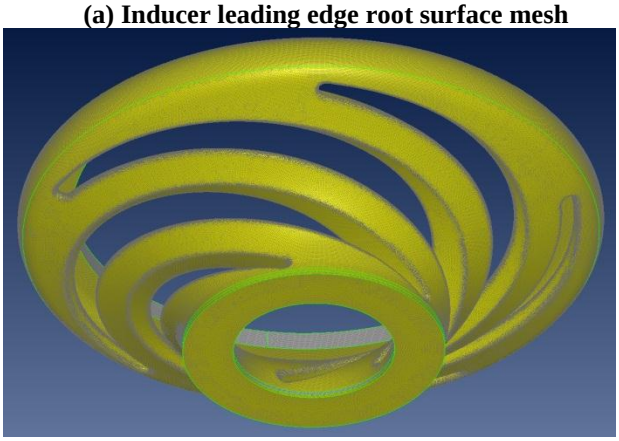
For accurate CFD simulation of the pump, areas with large surface deformations were well refined both on the surface and in the volume. A surface element generation technique was used that reduces element size to meet a maximum amount of deformation from one element to another. Figure 8 shows zoomed images of several areas of the pump where this can be seen. The volume mesh was created using AFLR3. Boundary layer creation was enabled, with tetrahedral merging to form prisms and pyramids, and a boundary layer cell growth rate of 1.2. Initially a mesh was built with wall spacing of 2e-5 inches, but in the interest of reducing the number of cells, a new mesh was built for use with wall functions with a wall spacing of 4e-4 inches for the first cell. This reduced the number of cells in the entire pump from 246 million to 142 million and made post processing activities much easier, as well reducing the computational cost of the simulations. To visualize the cell size and density, a cross-section of the Inducer/Kicker mesh and the Impeller-1 mesh is shown in Figure 9 as well as a sample overlap showing the local cell size and the size of the overlap. The simulation was run on several thousand processors on the Pleiades cluster at the Supercomputing Facility at NASA Ames.



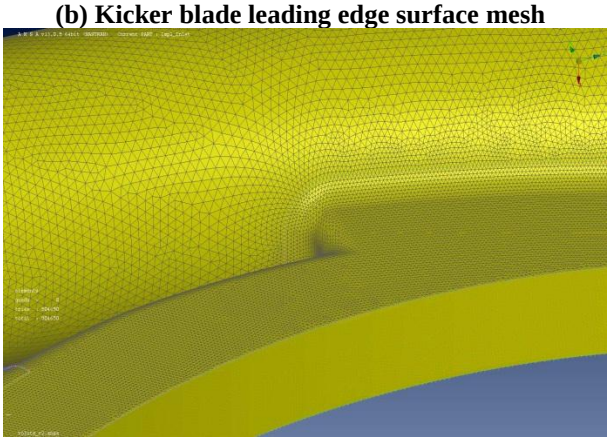
(a) Inducer leading edge root surface mesh



(b) Kicker blade leading edge surface mesh



(c) Crossover surface mesh showing resolution and dimensioned edges



(d) Volute tongue showing local surface resolution

Figure 8. Details of the surface mesh.

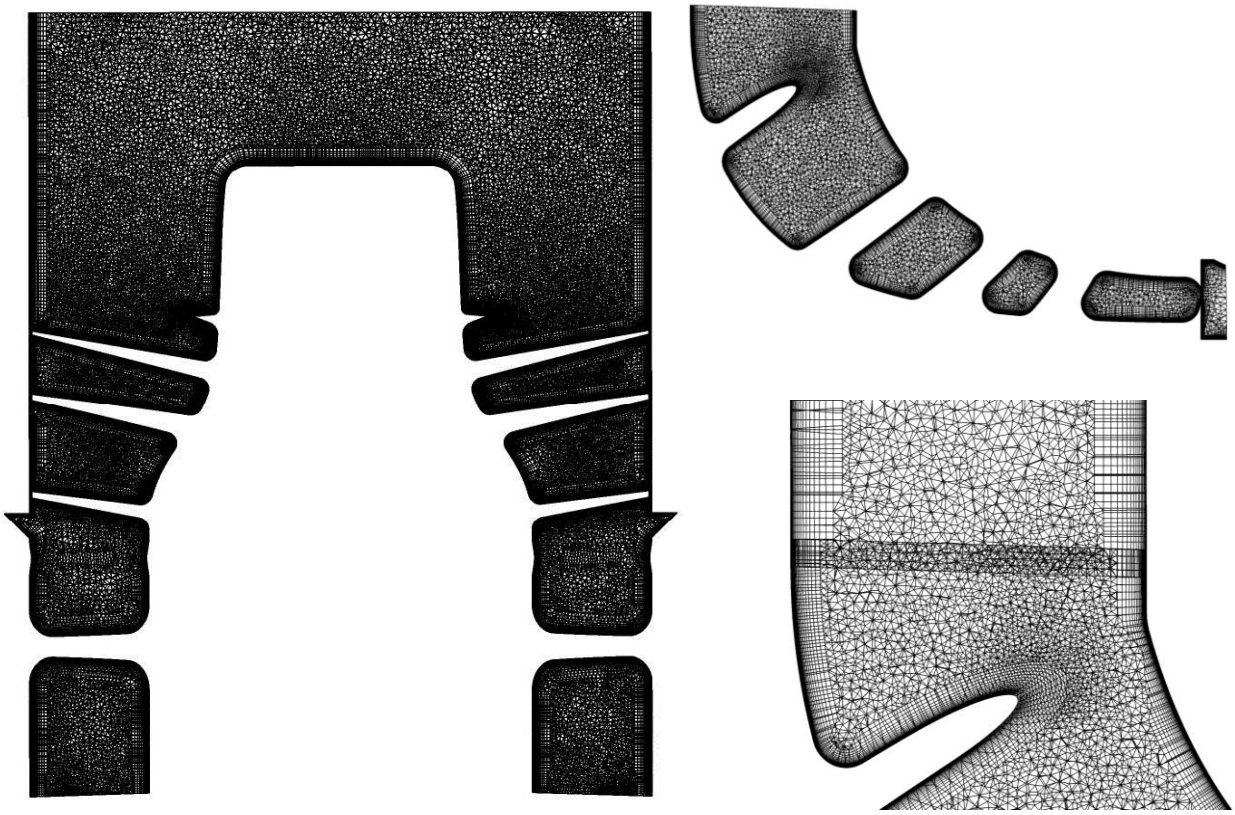


Figure 9. (a) Cross section of the inducer/kicker volume mesh), (b) Cross section of the impeller-1 volume mesh, and (c) Closeup of the kicker/impeller-1 mesh overlap

The initial conditions used were the outlet static pressure applied everywhere in the domain, and an axial velocity matching the axial velocity at the inlet needed to provide the proper inlet mass flow. The temperature is initialized to 41.6 Rankine, which corresponds roughly to the temperature in the middle of the pump. The turbulence quantities are extracted from a specified turbulence intensity and inlet velocity. The eddy-to-laminar viscosity ratio is then interpolated from a table using the logarithm of the Reynolds number which relates nearly linearly to the viscosity ratio.

For the initial case without leakage paths, the inlet was set to a subsonic boundary condition specifying a mass flow rate of 2.8654 lbm/s, and a temperature of 39.3 R. The outlet boundary was set to use a mean static pressure of 2380.7 psi(which includes 1000 psi pressure offset to compensate for low inlet pressures that may become negative). All other surfaces in the simulation used a noslip adiabatic boundary condition, with the rotating surfaces assigned the specified angular velocity. For the case where the leakage paths were enabled, it was restarted from a case using different boundary specifications. It was instead set to a constant total pressure constraint at the inlet boundary and a fixed mass flow rate at the outlet boundary. All leakages were set to maintain a fixed inlet mass flow rate or outlet mass flow rate, as appropriate for the particular leakage path.

The calculation uses a 'stiffened' equation of state (EOS), which is formulated much like an ideal gas EOS with pressure-invariant sound speed but with parameters m, P_0, and n picked to reflect the nearly incompressible nature of the fluid in the pressure and temperature range of interest and comparing the model to real properties taken from NIST tables. The basic equation of state is the following:

$$P = \rho \frac{\hat{R}}{m} T - P_0 \tag{45}$$

with dP/dρ derived as

$$\frac{dP}{d\rho} = \frac{\hat{R}}{m}T \tag{46}$$

The first step is optimizing the molecular weight, m, at the reference temperature T to match dP/dρ as well as possible in the problem domain according to Equation (46). Second, the value of P_0 is optimized to match the density as well as possible in the problem domain using Equation (45). Lastly, the number of degrees of freedom, n, is optimized to match the sound speed as well as possible in the problem domain according to Equation (47), below:

$$a = \sqrt{\frac{\gamma\left(P + P_0\right)}{\rho}} \tag{47}$$

For this calculation, a reference temperature of 28.33 K and range of pressures centered around ~430 psi were used to generate the EOS. The resulting parameters were P_0 = 24.0584 MPa, a gas index n = 0.7641, and an effective molecular weight m = 0.4453 kg/kmol. The corresponding reference sound speed is 1105.17 m/s with density ranging from about 60-75 kg/m^3 at the reference temperature over the span of expected pressures. Figure 10 shows the sound speed and density as a function of pressure used by the Loci/STREAM calculation compared to NIST data. Over the range of pressures of the problem, a constant sound speed is shown to be in error by as much as 30% at low pressures, although this plot is misleading since the temperature for which it is calibrated is far from the real temperature in those regions; the sound speed is more accurate than stated, due to variations in temperature from inlet to outlet. The density is better matched with error below 4% everywhere, and reaching its maximum near the outlet of the problem.

Figure 10 shows pressures that are 1000 psi higher than expected in the problem because the EOS used for this case used a technique known sometimes as a 'reference' pressure where the EOS is formulated using pressures that are offset by some constant value (1000 psi in this case) and then the problem is run at a pressure offset by the same amount. The results are then factored back out when post processing. This technique allows problems that would normally produce negative pressures (e.g. in cavitating regions) to remain numerically stable when solving the pressure equations. For simulations where cavitation and phase changes are not being simulated, this technique should be indistinguishable from the same case which does not use a reference pressure.

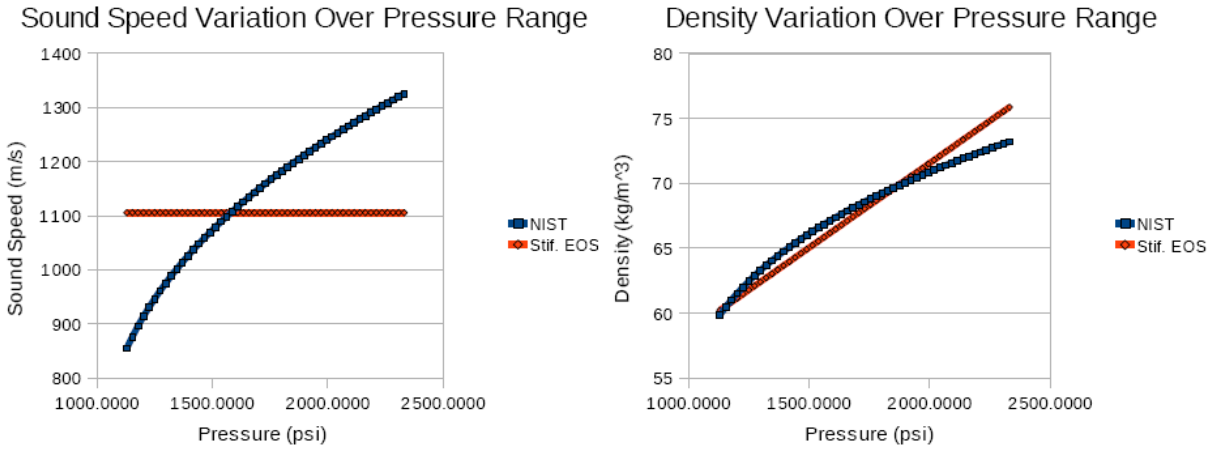
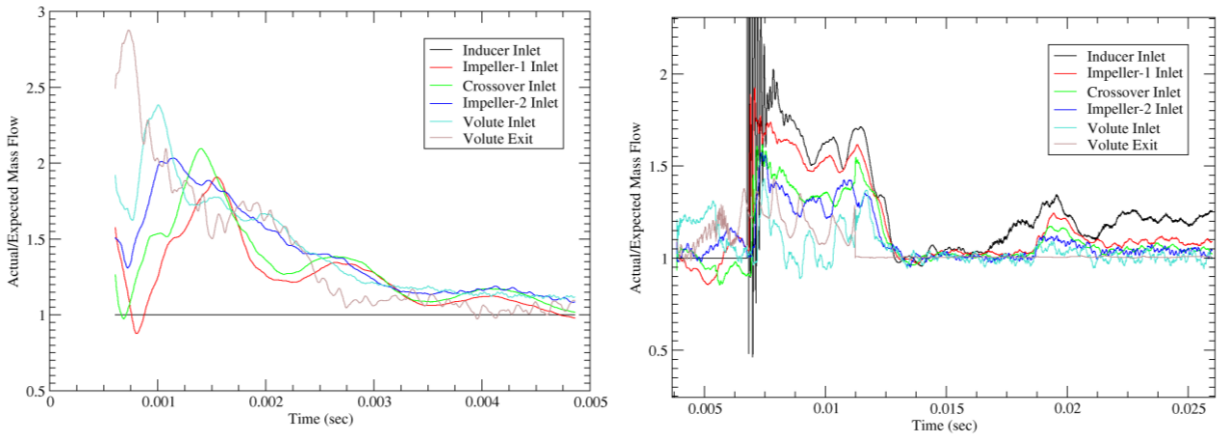


Figure 10: Comparison of stiffened EOS properties with NIST tables at reference temperature.

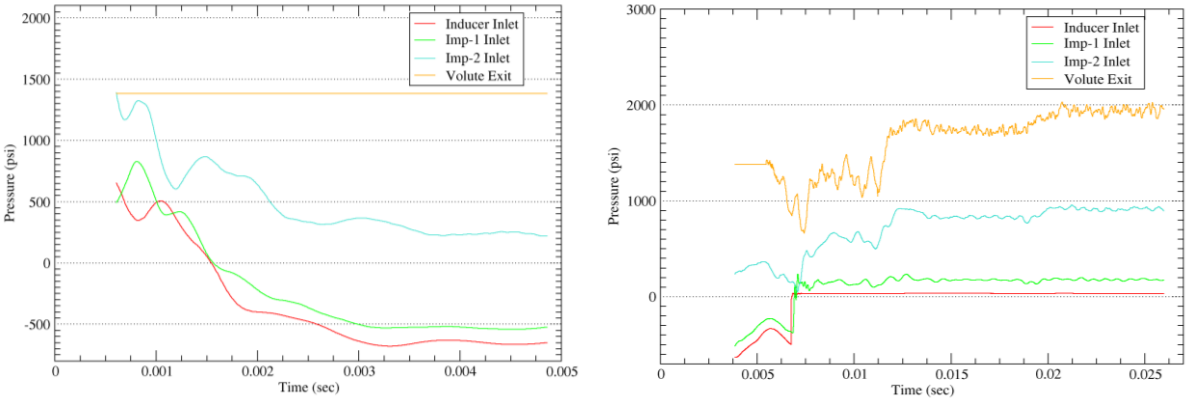
Shown in Figure 11 is a plot of the mass flow rates, integrated over the area of each component interface for each instant in time. It shows the initial transient of the calculation as well as the convergence time scale. At the end of the simulation the mass flow rate was converged to about 5% of the intended mass flow rate.



(a) No leaks **(b) With leaks**

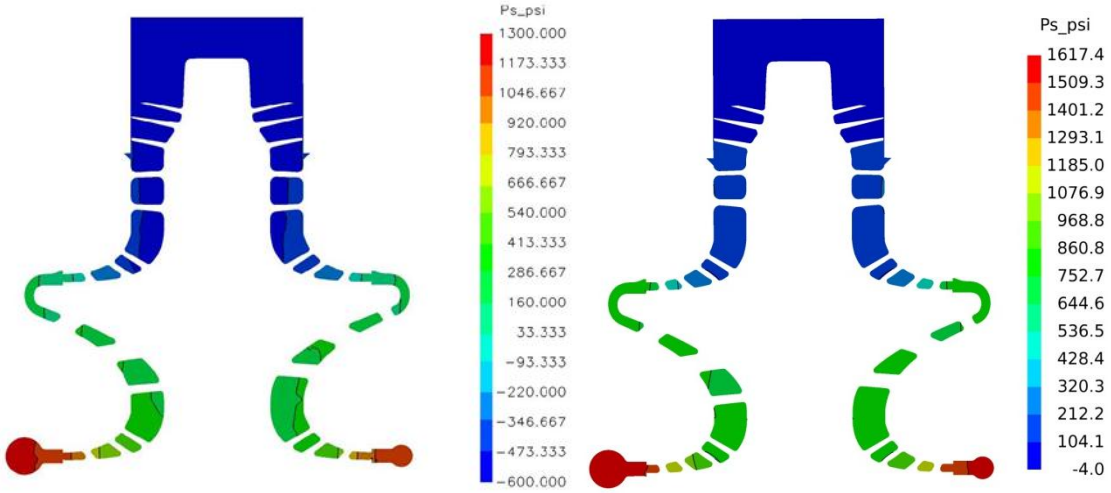
Figure 11: Instantaneous Mass Flow Rates at Component Interfaces

Figure 12 shows the pressures at the axial interfaces of in the pump. The values are obtained from the integrated axial force and dividing by the area. By the last revolution in this simulation, the pressures appear to have stabilized and have converged to about +/- 30 psi. As can be seen, the simulation stabilized at a pressure well below zero indicating an overprediction of pressure rise through the pump compared to the expected performance. Figure 13 shows the pressure field on a cross section through the midline, with large pressure rises in the impellers and a moderate pressure rise in the crossover.



(b) No leaks **(b) With leaks**

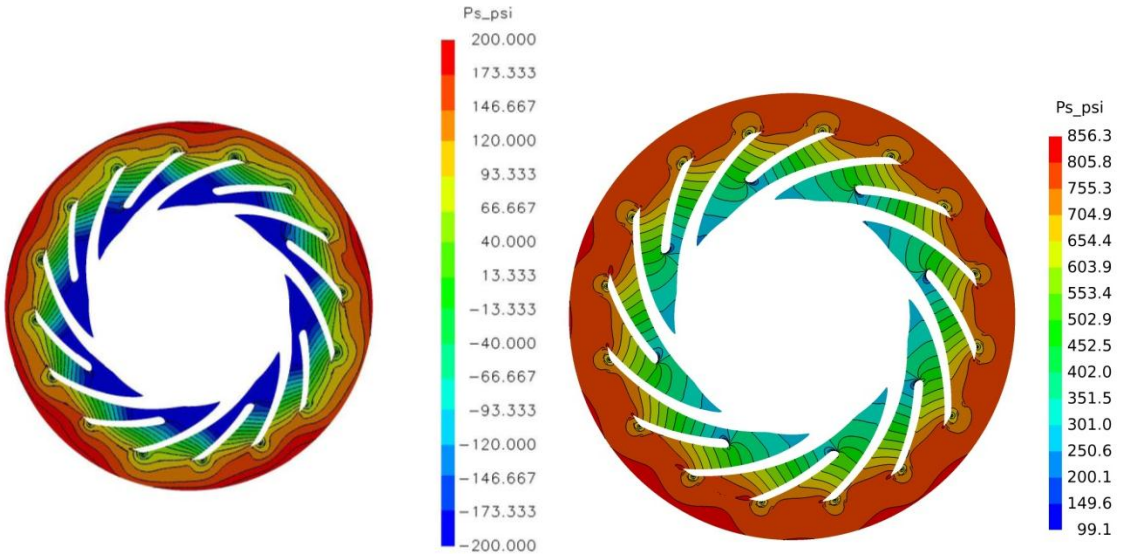
Figure 12: Instantaneous static pressures at component interfaces



(c) No leaks **(b) With leaks**

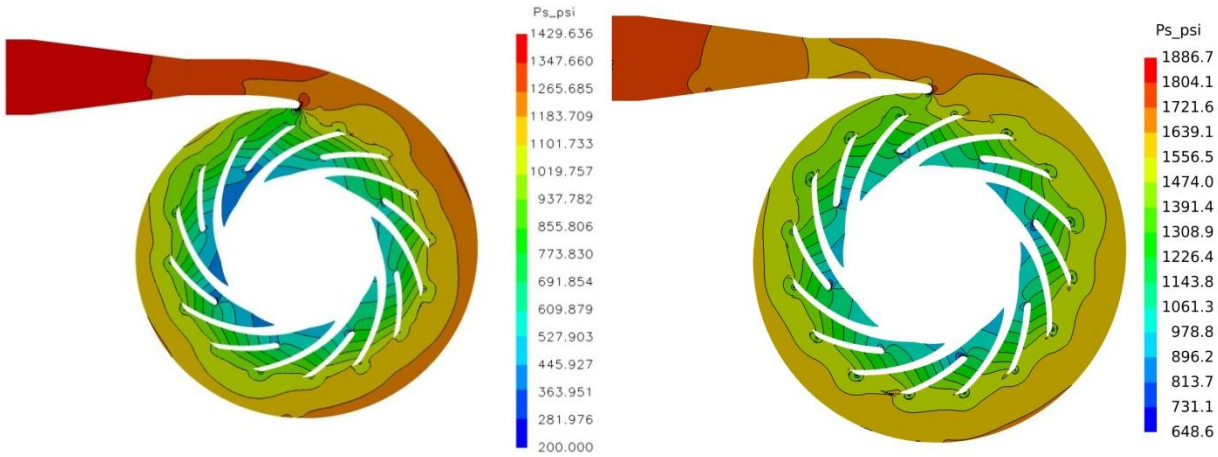
Figure 13: Cross section showing pressure field

Although the performance of the pump is being overpredicted when compared to the meanline estimate, the pressure field is well behaved without any problems between component boundaries, flowfield irregularities from the mesh, or any problems handling the rotation of some components in the domain. Two more views are shown in Figure 14 and Figure 15. The first view shows the outlet of the first impeller where the pressure exhibits a 5-lobed spatial distribution. The 5 lobes are the imprint of the 5 vane crossover which is immediately downstream of the impeller. The lower view shows the second impeller and volute. The volute tongue leaves a single lobed spatial imprint on the pressure distribution around the second impeller which creates an asymmetric load on the impeller.



(d) No leaks **(b) With leaks**

Figure 14: Pressure field around the Impeller-1 exit



(e) No leaks **(b) With leaks**

Figure 15: Pressure field around the Impeller-2 exit

VI. SUMMARY AND CONCLUSIONS

Overall, the turbomachinery test case presented above demonstrates that Loci-STREAM can handle the complexity of turbomachinery simulations quite well. Integrated results of interface pressures and mass convergence show that the computation of the test problem is stable and convergent. Predicted total head rise is ~20% high compared to the meanline calculation while static pressure rise is ~25% high. Further work is required to explore and investigate the effect of other factors on the outcome of the simulations including possible tunings to mass specified outflows, detached eddy simulation (DES) turbulence models, etc. The case presented above was intended to act as a pathfinder for further simulations of liquid turbopumps with Loci-STREAM which are under way.

References

[1] Wright, J. and S. Thakur, "A Rule-Based Algorithm for Liquid Rocket Combustion", *43rd AIAA/ASME/SAE/ASEE Joint Propulsion Conference*, Paper No. AIAA-2007-5528, Cincinnati, OH (July 2007).

[2] Luke, E.A., "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 (Dec. 1999).

[3] Patankar, S. V., Numerical Heat Transfer and Fluid Flow, Hemisphere, Washington, DC (1980).

[4] Rhie, C.L. and Chow, W.L., "A Numerical Study of the Turbulent Flow Past an Isolated Airfoil with Trailing Edge Separation," *AIAA Journal*, **21**: 1525-1532 (1983).

[5] Shyy, W., Thakur, S. S., Ouyang, H., Liu, J. and Blosch, E., Computational Techniques for Complex Transport Phenomena, Cambridge Univ. Press, NY (1997).

[6] Menter, F.R., "Two-Equation Eddy-Viscosity Turbulence Models for Engineering Applications," *AIAA J.*, Vol. 32, No. 8, pp 269-289 (1994).

[7] Spalart, P.R., Jou, W.H., Strelets, M. and Allmaras, S.R., "Comments on the Feasibility of LES for Wings and on a Hybrid RNAS/LES Approach," *1st AFOSR Conference on DNS/LES*, (1997).

[8] Strelets, M., "Detached Eddy Simulation of Massively Separated Flows," *AIAA Paper 2001-0879* (2009).

[9] Menter, F.R. and Kuntz, M, "Adaptation of Eddy-Viscosity Turbulence Models to Unsteady Separated Flows Behind Vehicles," The Aerodynamics of Heavy Vehicles: Trucks, Buses and Trains, McCallen, R., Browand, F. and Ross, J., eds., Springer, New York (2004).

[10] Spalart, P.R., Deck, S., Shur, M.L., Squires, K.D., Strelets, M.K. and Travin, A., "A New Version of Detached-Eddy Simulation, Resistant to Ambiguous Grid Densities," *Theor. Cimput. Fluid Dyn.*, Vol. 20, pp 181-195 (2006).

[11] Shur, M.L., Spalart, P.R., Strelets, M.K. and Travin, A.K., "A Hybrid RANS-LES Approach with Delayed-DES and Wall-Modelled LES Capabilities," *Int. J. Heat Fluid Flow*, Vol. 29, pp 1638-1649 (2008).

[12] Spalart, P.R., "Detached-Eddy Simulation," *Ann. Rev. Fluid Mech.*, Vol. 41, pp 181-202 (2009).

[13] Travin, A., Shur, M., Strelets, M. and Spalart, P.R., "Detached-Eddy Simulations Past a Circular Cylinder," *Flow Turb. And Comb*., Vol. 63, pp 293-313(1999).

[14] Menter, F. R., Kuntz, M., and Langtry, R., "Ten Years of Industrial Experience with the SST Turbulence Model," Turbulence, Heat and Mass Transfer 4, ed: K. Hanjalic, Y. Nagano, and M. Tummers, Begell House, Inc., pp. 625 – 632 (2003).

[15] Thomas P.D., Lombard C.K., "Geometric conservation law and its application to flow computations on moving grids," *AIAA Journal*; Vol. 17, pp 1030–1037 (1979).

[16] Nkonga, B. and Guillard, H., "Godunov type method on non-structured meshes for three-dimensional moving boundary problems," *Comput. Methods Appl. Mech. Engrg*. Vol. 113, pp.183-204 (1994).

[17] Koobus, B. and Farhat, C., "On the Implicit Time Integration of Semi-Discrete Viscous Fluxes on Unstructured Dynamic Meshes." *Int. J. Numer. Meth.* Fluids, Vol. 29, pp 975–996 (1999).

[18] Farhat, C., Geuzaine, P. and Grandmonty, C., "The Discrete Geometric Conservation Law and the Nonlinear Stability of ALE Schemes for the Solution of Flow Problems on Moving Grids," *J. Comp. Phy*., Vol. 174, pp 669–694 (2001)

[19] Smith, R.W. and Wright, J., "An implicit edge-based ALE method for the incompressible Navier–Stokes equations," *Int. J. Numer. Meth. Fluids*; 43:253–279 (2003).

[20] Hassan, O., Sorensen, K.A., Morgan, K. and Weatherill, N.P., "A method for time accurate turbulent compressible fluid flow simulation with moving boundary components employing local remeshing," *Int. J. Numer. Meth. Fluids*; **53**, pp 1243–1266 (2007).

[21] Mavriplis, D.J. and Nastase, C.R., "On the geometric conservation law for high-order discontinuous Galerkin discretizations on dynamically deforming meshes," *J. Comp. Phy*., Vol. 230, pp 4285–4300 (2011).