Introduction

Motivation

Optimization has been the one of the main goals of computational since the beginning of computing. Designing something to perform optimally in all conditions is the end goal, but for the most part we simply focus on local optimization for a single application at a time. Optimization takes time, however. The optimization process can be a very long and arduous task of changing the value of a single variable each time and running a new simulation to determine what should be done with the new variable value. This leads to many simulations for a small benefit. This type of optimization tends to be so costly that it prohibits very large scale optimizations simply because of run time.

With this problem in mind, adjoint based optimization was implemented. Adjoint based optimization allows for optimization of a multitude of control variables in a single simulation. The process simply runs the simulation forward to the end of the interaction and then maps back the simulation values for each step with a reverse solver to determine the direction and magnitude that the control variables should be moved. Adjoint based optimization is not without flaws itself. To run an iteration of adjoint based optimization, the code must run the entire forward simulation, while storing every step for use in the reverse solver. This proves to be very costly for simulations of large size in either the time or space components. For real world problems, we often need at least one of those components to be very large, usually both. To get valid results for use with actual problems, we need high fidelity in the results, which means we need at least fairly fine meshes in the simulation region leading to even larger amounts of data to be stored for the reverse solver.

The adjoint problem becomes intractible for the naive approach of just simply writing out every step and then reading it back in when necessary. For this reason, the concept of checkpointing was originally applied. Checkpointing allows for the execution of adjoint-based optimization with a greatly reduced need for storage space. With checkpoionting, one runs a forward simulation and instead of storing every step, only stores a few checkpoints from which to restart the forward solver when the reverse solver gets to the point where it needs the steps between them. This greatly reduces the need for storage space to run a simulation. With improvements in how the checkpointing is done, the possible simulations are becoming larger and longer. Thanks to these advances, adjont based optimization has now become applicable to very large problems. We introduce a new implementation of checkpointing that allows for adjoint-based optiomization of fully 3D jet flows and noise caused by them.

Jet engine noise is a problem as old as jet fighters themselves. As the engines got and continue to get more powerful, the problem of engine noise gets continually worse. When looking at aircraft carriers, the problem is even larger. The solution to jet engine noise in commercial airliners has been universally to increase the bypass ratio. Increasing the bypass ratio allows for more cold air to be blown around the jet engine in the cowlings around the outside. This makes for a much larger mixing layer along with making for a much slower transition from the ambient air to the hot and fast jet engine exhaust. This is not feasible for military jets as the high demands on military jets mean that the loss of possible thrust to pushing the bypass air around the engine is completely unacceptable. Not only are there many jets on the carrier and taking off nearly constantly, but the workers on the deck of the carrier are exposed to the jet noise from a distance much less than anywhere else. This creates a situation that can be extremely dangerous for the hearing of the aircraft carrier deck workers. Under OSHA regulations, a person working on the deck of a modern aircraft carrier could work less than eight minutes before needing to take a full day off. The noise is even more pervasive, given that when the workers retire from the deck for the day, they sleep only a few floors below the source of the noise, thus getting exposed to a portion of the noise even in down time.

For the simulation of the real noise generated by the jets, one also has to take into account the material properties of the jet engine as well as the cowlings and exhaust materials. If the materials are too flexible, the jet exhaust tip may deform and lose thrust or possibly cause the flow to be even more over or underexpanded, leading to even more noise. This need leads to the need of a Fluid-Structure interaction feature in the code.

Fluid Structure interaction is also important for the other main motivation for this research. That motivation is protection from blast waves created by explosions, namely improvised explosive devices. Improvised explosive devices are a plague on the military in the current conflicts in the middle east. With the improvements in body armor and military intelligence along with drone warfare and bombing capabilities, more of the members of the military are surviving the actual conflicts. This has pushed the insurgent members of the enemy to resort to methods of sneak attack. The sneak attacks are accomplished by either proximity or remotely activated explosives. The blast waves from these devices are proving to move directly through the current body armor, thus penetrating the soldiers' bodies and creating internal hemoraging in any cavities with air such as the lungs. The blasts are also causing brain injuries at a rate previously unseen. This is probably due again to the advances in military technology and the advances in medical technology that is keeping these miliotary members who would have died in previous conflicts. Since these soldiers are now going to be surviving the blasts, it is the duty of science to undertake the job of trying to prevent the blast induced brain injuries that are now becoming a major problem for the returning soldiers.

Background

CFD

Computational Fluid Dynamics is a field of computational physics that models the flows of fluids. This uses boundary conditions and definitions of the gas or fluid constitution in order to simulate the interaction of the fluids in the flow and the surrounding fluid.

LES

Large Eddy Simulations are simulations of fluid dynamics that include near field and far field flows of trubulent flows. They allow for better resolution than RANS models and better computational efficiency than DNS models.

Unsteady Simulations

The simulations used in the studies are unsteady in nature, meaning that the flow does not ever reach a steady state and is in constant flux.

Unstructured Grids

The code is built on unstructured grids, allowing for much more intricate geometries than are possible for the same number of elements in a structured code. It also allows for easier refinement in some regions without need for global refinement.

CSD

Computational Structural Dynamics models the motion and reaction of solids during collisions and in response to stresses from inside and outside forces.

FSI

Fluid-Structure Interaction is a coupling of a fluid solver with a structural sover in order to model the reaction of a solid to changes in a fluid flow and in return the reactions of the fluid flow to the solid and the changes in the solid.

Adjoint-Based Optimization

Optimization of a flow with respect to a certain trait that one would like to minimize or maximize. Given a control surface with n points, adjoint-based optimization can give a large step toward the optimal solution for roughly the cost of 2-3 forward simulation compared to n forward simulations with standard optimization techniques. Storage and communication requirements tend to be a limiting factor for adjoint-based optimization making it prohibitively costly without additional estimation or other advances.

Finite Element Method

Finite element method description

Eulerian motion

Fluid is described in a global reference frame.

Lagrangian motion

Motion is described in the frame of reference of the particle and the motion of said particle

Previous Work

Jet Engine Noise Reduction

There have been a multitude of papers on jet engine noise reduction. The main work that will be focused on in this paper is based on the work by Kailasanath et al. This work is on the addition of flow disrupters on the end of the exhaust nozzle called chevrons. These chevrons allow for the sheer regions to be minimized by helping to add some vorticity in the streamwise direction of the flow and thus increasing the mixing layer. The major benefit of this is a small reduction in noise in the aft direction, but a large reduction in the forward direction. Namely, the chevron addition helps to reduce and eliminate the screech from supersonic jets. Screech is the forward propogation of the shock wave from the supersonic gasses escaping the nozzle in the subsonic ambient air. The research also implies that the main noise-generating region of the noise that propogates to the far field is the shear layer noise and in order to reduce overall far field noise the goal should be to increase the mixing layer and thus decrease the shear layer. The pressure waves from the mixing layer tend to disipate in the wake of the flow in much less time than that of the shear noise. Arguably, this would be because the shear noise is escaping without being subjected to the vorticity of the mixing layer.

There have also been a number of suggestions for flow interrupters inserted into the nozzle of the jet exhaust. These are shown to reduce the noise level of the jets, but the need of military style jets to utilize every bit of thrust available tends to make these solutions a bit too expensive in the thrust cost to be deployable. For that reason, the research will be looking into ways of decreasing jet engine noise with a minimal and ideally negligible cost to thrust.

One possible solution which could provide both a reduction in noise and minimal cost to possibly even marginal gain in thrust is fluidic injection at the nozzle mouth. Fluidic injection could introduce streamwise vorticity while adding a bit of material flowing in the direction of the jet exhaust.

Fluid Structure Interaction

Fluid Structure interaction has been used for generations in one form or another. It started out as one way coupling of fluid pressures to solids and then grew to a two-way coupling of the fluid and structure. Most of the modern fluid structure codes are using ALE solvers. ALE stands for arbitrary Lagrangian-Eulerian solutions. This means that the fluids are solved in eulerian form and the solids in lagrangian form, as both are naturally done in their respective individual codes. With ALE forms, a code can only map small deformations before it must be regridded in order to avoid the development of nonconvex elements which will lead to useless reults. This can introduce large time sinks which make the ALE forms extremely costly and can introduce some extra error which could be avoided if one could write a fluid solver and solid solver in the same coordinates.

Unified continuum methods attempts to do just that. Most unified continuum methods take a strictly eulerian approaches for the fluid and the solid. In most cases they treat the solid as eulerian for the interaction but continually map it back to original position in order to calculate the standard lagrangian for of the solids and then map back to eulerian for the interface. The interfact is typically tracked from the movement of the solid and is kept in a continuum method such that a small amount of area in the boundary region between the solid and fluid is treated as a mixture of the two. This leads to a semi-stagnation of the fluid around the solid and can lead to some non-physical results.