Daniel Dotson's Project Portfolio

Over the years, I have completed many projects with the intent of gaining a deeper understanding of the concepts introduced to me in my education. I have picked out a few to discuss in this document, which is divided into two halves. The first half presents the notable results I have achieved with the various codes I have written. I have a lot to say about these results, both good and bad, but here I have decided to just let them speak for themselves. The second half provides an overview of the methods I have used and where they have come from. While most of my projects focus on fluid mechanics, the lessons I have learned in completing them are relevant across every engineering discipline.

3D Unsteady Panel Code

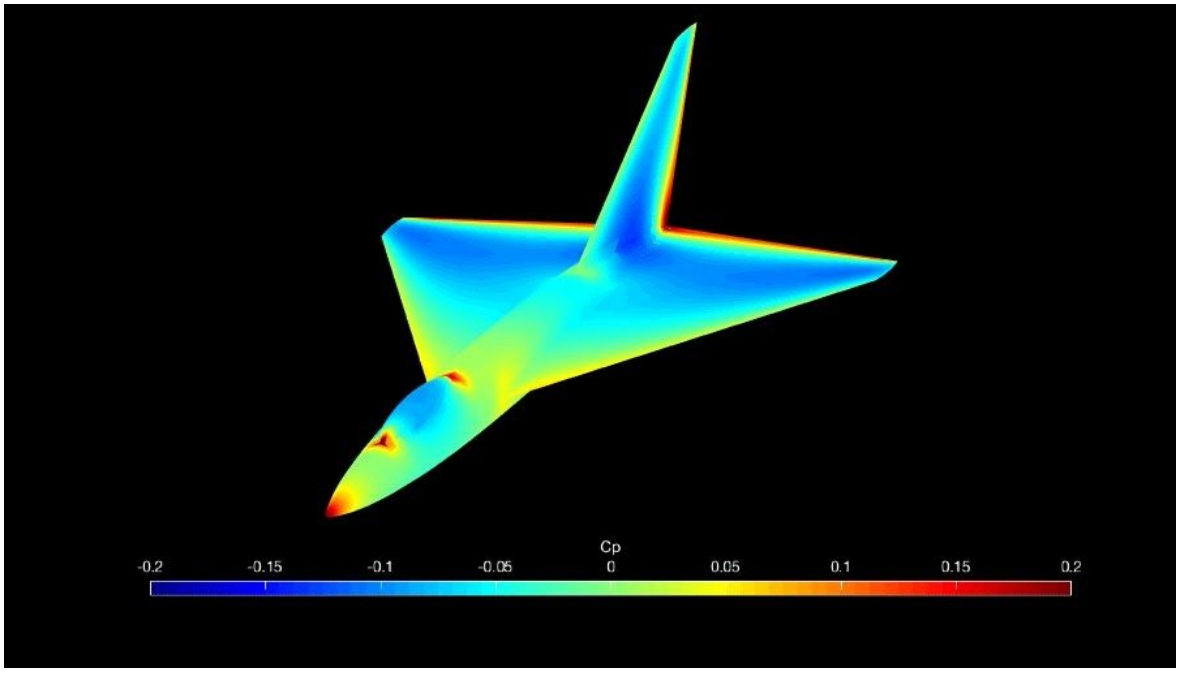

Figure 1. Pressure coefficient on the surface of a generic jet fighter at 0° angle of attack.

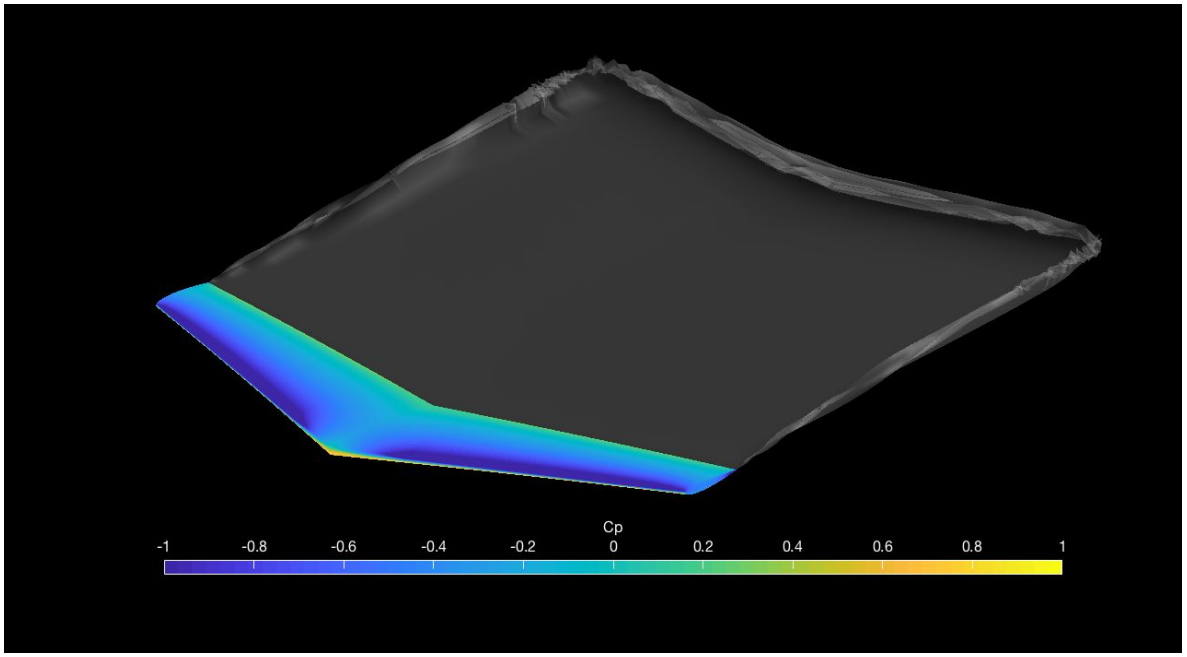

Figure 2. Unsteady pressure coefficient on the surface of a typical wing at 6° angle of attack, moments after a sudden start. Wake structure plotted as a translucent shaded surface.

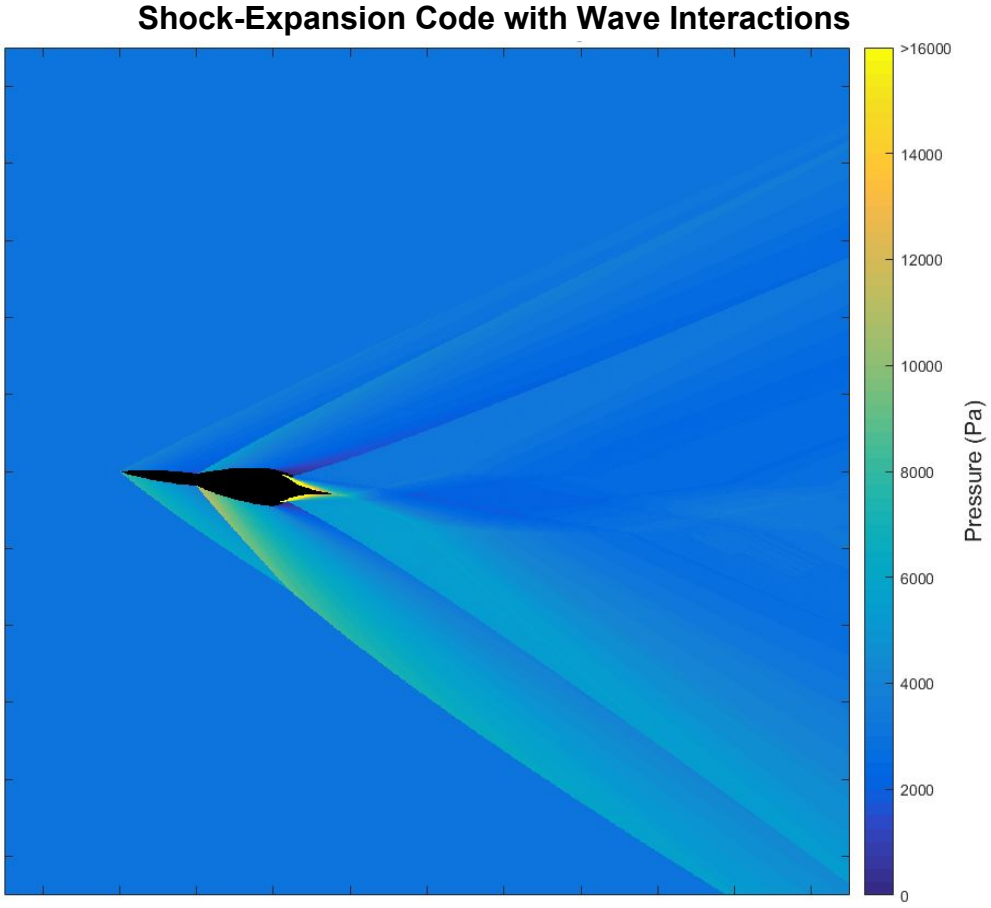

Figure 3. Pressure surrounding a vehicle at 6° angle of attack, powered by an underexpanded aerospike engine, traveling at Mach 2.5 and at an altitude of 24km.

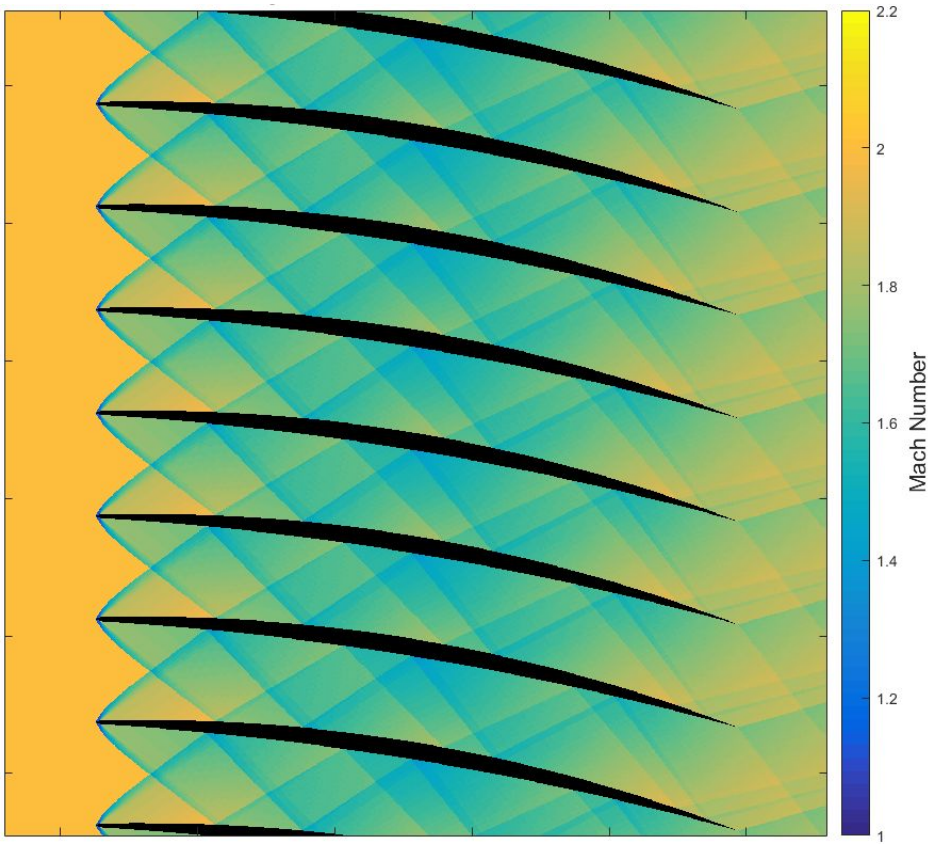

Figure 4. Mach number surrounding a series of supersonic turning veins, with an inlet Mach number of 2 and an inlet pressure of 10000 Pa.

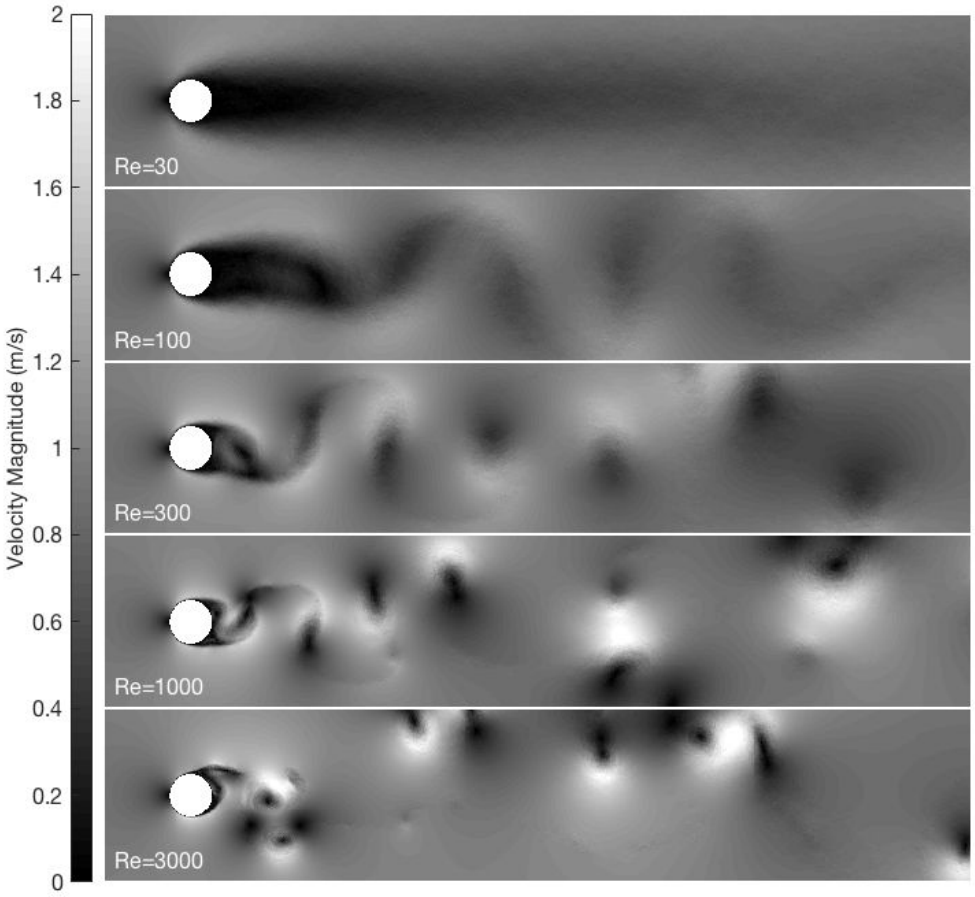

Figure 5. Instantaneous velocity magnitude surrounding circular cylinders at various Reynolds numbers.

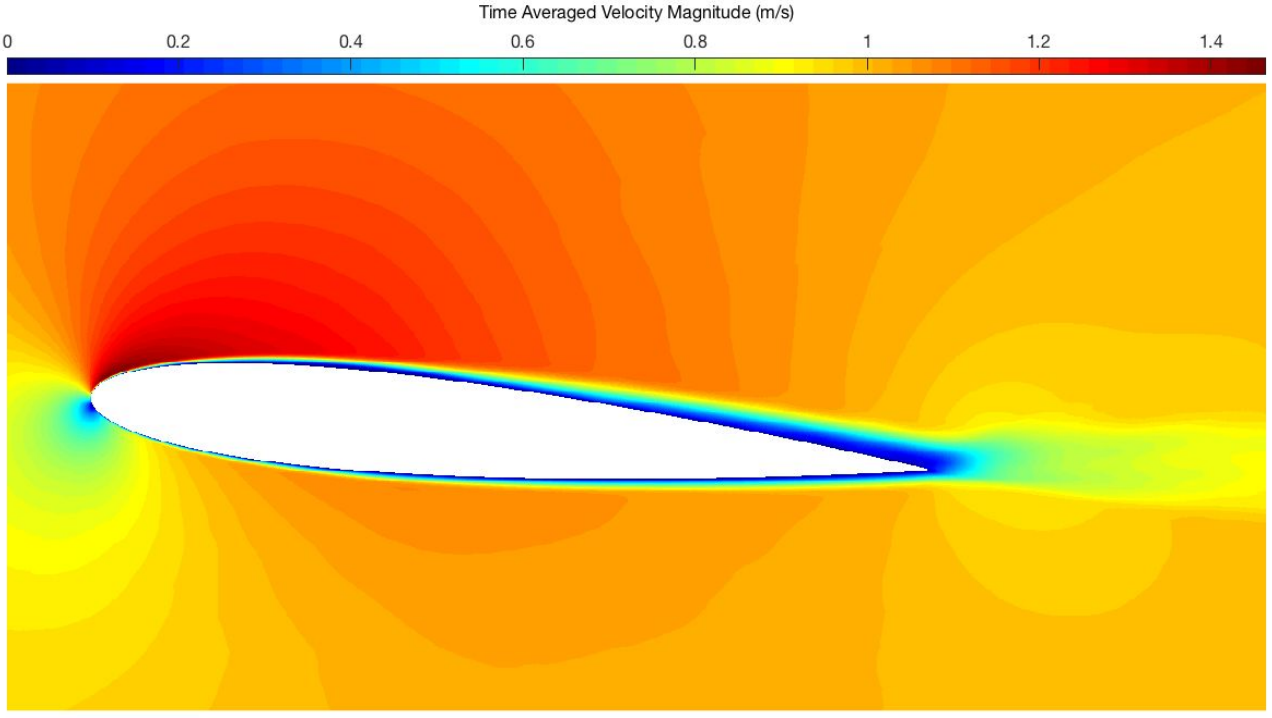

Figure 6. Time-averaged velocity magnitude surrounding a NACA 0012 airfoil at 5° angle of attack, with a Reynolds number of 50000.

Static Aeroelasticity Code

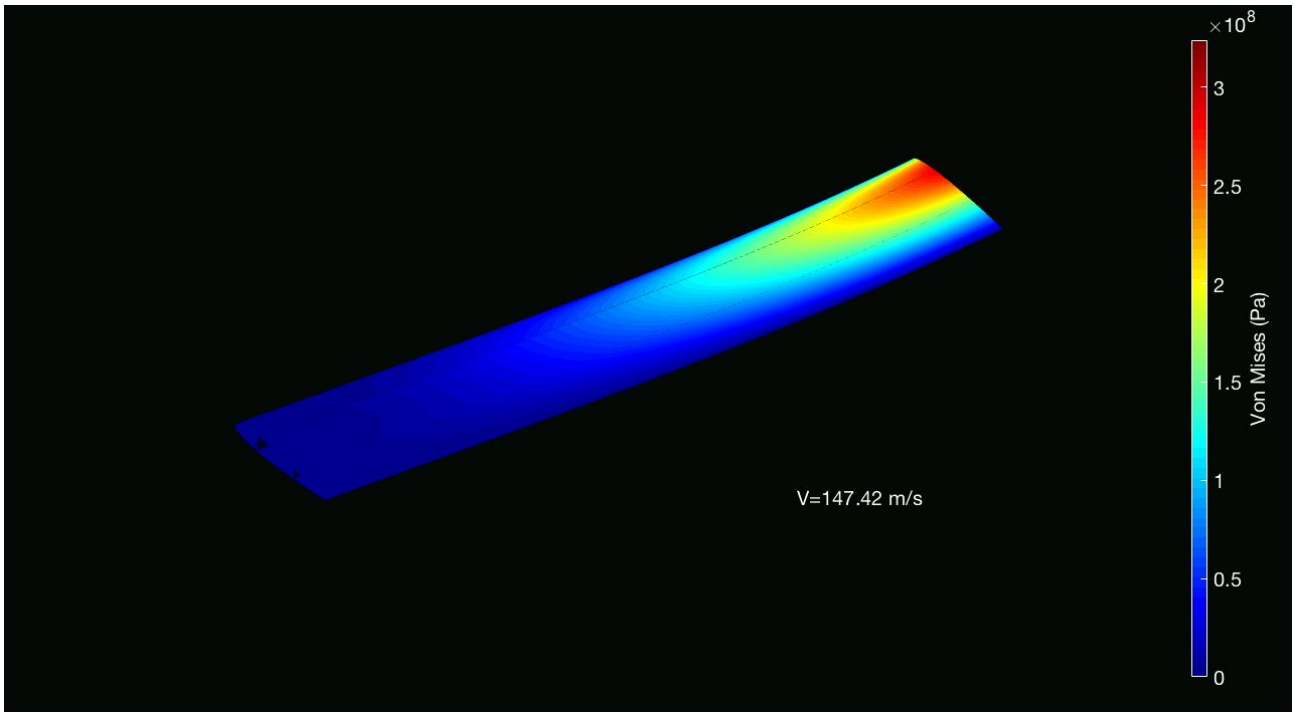

Figure 7. Von Mises stress within a two-spar torsionally compliant wing structure near the aeroelastic divergence dynamic pressure.

Gravitational N-Body Code with Collisions

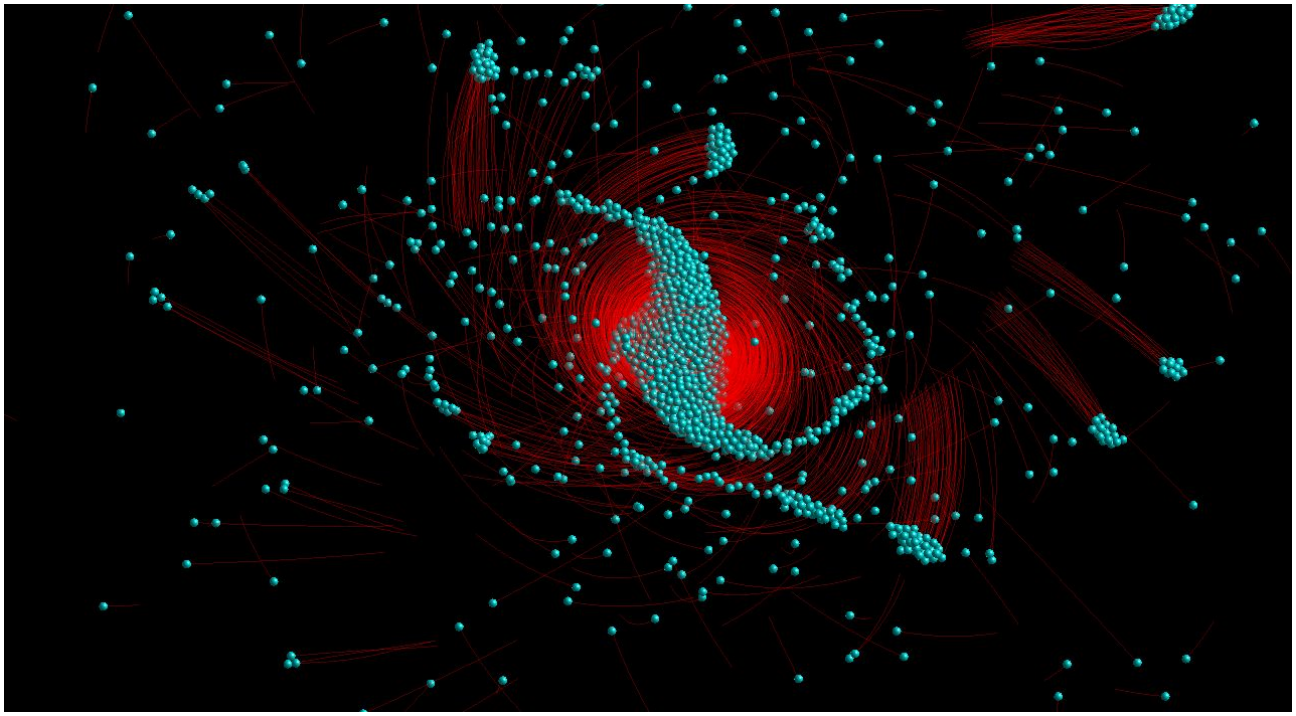

Figure 8. Accretion structures within a system of 2000 point masses subject to Newtonian gravity with inelastic collisions. Particles initialized in a uniform random spherical distribution with random velocities tangential to their position vectors.

Event-Driven Hard Sphere Molecular Dynamics Code

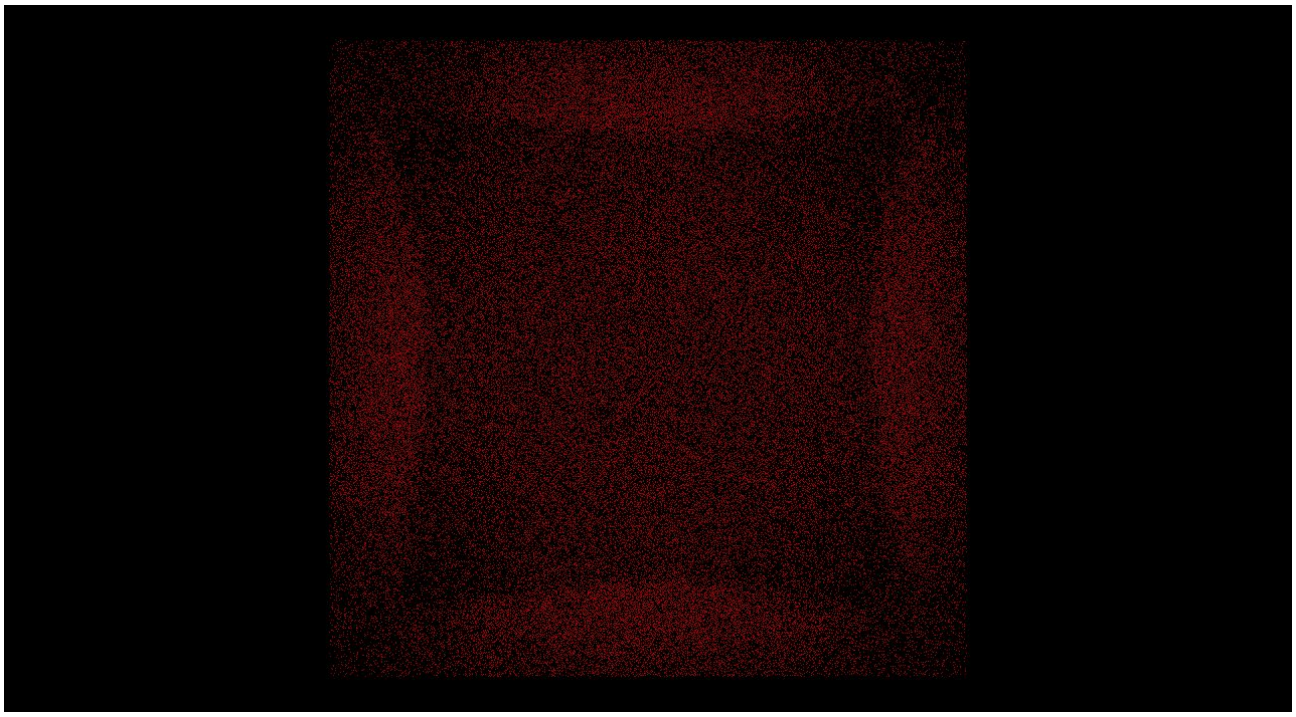

Figure 9. Simulation of an ideal gas in non-equilibrium within a box with adiabatic frictionless walls. Particles initialized within a smaller box in the center, with random velocities. Shockwaves are clearly visible, reflecting off the box walls. This simulation contained 90000 molecules and computed approximately 30 million collisions.

3D Unsteady Panel Code

Aerodynamic flows of sufficiently high Reynolds numbers contain very low viscous stress and vorticity outside of a thin boundary layer and wake. At sufficiently low Mach numbers where density can be assumed constant, the governing equations of the irrotational flow field can be simplified to a Laplace equation of the velocity potential, a scalar field whose gradient gives the velocity field. This velocity field and its potential can be decomposed into a freestream component and a perturbation component due to immersed boundaries such as a body surface and wake. The perturbation velocity must go to zero moving away from these immersed boundaries. It can be demonstrated through the divergence theorem that a perturbation velocity potential field satisfying these criteria can be represented as a non-unique doublet and source singularity distribution on the immersed boundaries. The problem is then closed by assuming some general form of these singularity distributions and ensuring that the circulation produced by the doublet distribution is equivalent to the circulation contained within the boundary layer and wake of the real flow.

This is the foundation of the panel method, also known as the boundary element method, which discretizes the source and doublet distributions using panels and selects a circulation using a Kutta condition. It was considered the state of the art in external flow modeling for several decades, and while higher-fidelity methods have now become practical with advances in computing, panel methods still occupy an important niche in fast aerodynamic analysis, particularly in early design phases where optimization involving multiple disciplines may be required.

My panel method is based on the numerical recommendations of Katz and Plotkin in the book “Low-Speed Aerodynamics”. It uses uniformly distributed sources and doublets on triangular panels. The source strengths are set to the component of the freestream velocity normal to the surface, which implies that the perturbation velocity inside the body must be zero. A Kutta condition is enforced by introducing wake doublet panels in the plane of the freestream velocity with a strength equal to the difference across the trailing edge. This leaves a unique solution to the doublet distribution. Pressure is computed using the unsteady bernoulli equation.

The paneling of the wake in this way is acceptable for thin wings at low angles of attack in a steady-state flight condition, but in reality the wake must deform so that it remains force-free. Another way to think of this is that the wake panels carry some amount of free vorticity that, according to the incompressible vorticity equation, must advect and deform with the flow. This can be accomplished using a wake advection scheme. The nodes of the wake panels are moved according to computed velocities at discrete time steps. A treatment of the wake in this way not only extends the method to unsteady flight conditions, but also provides a much more accurate representation of the wake structure, including the phenomenon of wing tip vortex rollup.

My code could be optimized and improved in many ways. The greatest increase in computing time would be achieved by using the Fast Multipole Method, especially if parallel computing was utilized. Higher-order panels would also help to reduce the number of panels, and a more accurate Kutta condition would yield more accurate pressure distributions for edge-case geometries.

Additionally, the wake model could be improved in several ways. First of all, a proper mollifier should be used to eliminate velocity singularities caused by the wake panels. Currently, I only use a cutoff distance. Secondly, the strengths of the panels should be adjusted as they stretch to conserve angular momentum. This is particularly important if the wake contains large deformations. Thirdly, the wake will often develop curvature singularities at the panel edges that destabilize the solution if no wake refinement is introduced. Some of these wake modeling difficulties can be substituted for others by using a particle wake instead of a panel wake. The application of the Kutta condition to fuselages and their interface with lifting surfaces is still an active area of research.

Shock-Expansion Analysis Code with Wave Interactions

High Reynolds number supersonic flows can be described accurately in many cases using the Full Potential Equation, a partial differential equation that is hyperbolic, and therefore solvable using the Method of Characteristics, where flow properties are constant along discrete characteristic lines. A variant of this method that is used often in supersonic nozzle analysis assumes flow properties are instead constant between characteristic lines. The characteristic lines are Mach waves and they are often produced by Prandtl-Meyer expansion fans. The reason the method is often limited to supersonic nozzles is because while expansion Mach waves spread out, compression Mach waves converge to form a shockwave. The Full Potential Equation assumes the flow is isentropic and does not properly account for the change in entropy when calculating the flow properties across shockwaves. For this reason, attached shockwaves are evaluated using the Rankine-Hugoniot relations instead.

This is the foundation of Shock-Expansion Theory. Concave turning of the flow produces shockwaves evaluated by the Rankine-Hugoniot relations and convex turning of the flow produces Prandtl-Meyer expansion fans evaluated using the Method of Characteristics. This description of supersonic external flows can be effective but raises one significant question. What happens when, perhaps some distance away from the body surface, these waves intersect?

The Method of Characteristics is capable of evaluating the interaction of expansion waves, which should only happen in flows with multiple surfaces that are not connected. But the shockwave interactions would be especially important if the concave corners in the flow are stretched out into surfaces of finite negative curvature. The shockwaves will converge and combine. The naive approach to a shock-shock interaction would be to set the strength of the combined shockwave so that the flow is turned the same amount that the individual shockwaves turned it. Unfortunately, this results in a discrepancy in the pressure above and below the point of intersection. Pressure can only vary across waves, so this straight-forward combination is not permitted. Similarly, if the combined shock strength is set so that the pressure increase is equal to the pressure increase across the individual shocks, there will be a discrepancy in flow direction. The solution is to introduce a “reflected” wave, which can either be a weak shockwave or expansion wave, from the point of intersection. The strength of the combined shock and the strength of the reflected wave are solved numerically, where a flow direction is assumed and refined until pressure is matched. Often, a discrepancy in Mach number will remain, which is permitted in inviscid flows and is represented as a slip-surface. This description of shock-shock interactions is detailed in John Anderson’s book, “Modern Compressible Flow”, but this method of solution seems to be valid for all interactions, including shock-expansion interactions, shock-slip

surface interactions, etc. If applied to expansion-expansion interactions, it gives the same result as the method of characteristics. Since every interaction typically results in at least three new lines to keep track of, the number of interactions grows exponentially, and sufficiently weak ones should be omitted using averaging.

This extended shock-expansion theory is extremely powerful, because as long as the entire flow field is supersonic, it can be solved for an arbitrary number of bodies in an infinite fluid domain. Only the body surfaces must be discretized, and no volume mesh is required. All flow discontinuities are captured naturally with no need for refinement. The solution fails if the fluid turning at any point results in a detached shock, because then the flow becomes locally subsonic and can no longer be represented in this discrete wave form. However, relatively blunt leading edges can still be evaluated at high Mach numbers by setting the tip angle to a value just below the threshold of detachment. Bodies with blunted bases can be evaluated if they are assumed to emit a supersonic jet, like a rocket nozzle. All kinds of interesting shock structures can be evaluated this way, including curvature, reflection, and far-field attenuation.

The biggest bottle neck of the method is calculating and recalculating all of the possible interactions. Everytime an interaction is handled, the interaction list must be updated and sorted. Significant speedups could be introduced here using many of the techniques in event-driven Molecular Dynamics for collision detection. Only a small number of interactions need to be recalculated every time the interaction list is updated, and a cell structure could be used so that interactions are only checked locally. This alone would likely accelerate the method by several orders of magnitude.

Vortex Particle Code

Taking the curl of the incompressible Navier-Stokes equations results in the vorticity transport equation, where vorticity is defined as the curl of the velocity. The equation describes how the vorticity changes within a fluid parcel advecting with the local flow velocity. The first term on the right-hand side describes vortex stretching, which only occurs in three dimensions. The second term describes vorticity diffusion and is the only viscous term in the equation. Given some initial vorticity field, the equation can be solved by clumping the vorticity into discrete lagrangian particles and moving the particles according to the velocities at their centers. The diffusion is superimposed onto the particle advection, and there are several techniques for doing this. The first is the random-walk algorithm described by Chorin in the paper “Numerical Study of Slightly Viscous Flows”. Other common methods for handling diffusion include core-spreading, particle-splitting, particle strength exchange, and a number of grid based methods. The velocities at the particle centers are computed using the Biot-Savart law.

Representing flow fields in terms of the vorticity instead of the velocity has some advantages because, especially for high Reynolds number flows, vorticity is often concentrated in relatively thin regions, such as boundary layers and wakes. The lagrangian particle representation also avoids the numerical diffusion problem that grid methods encounter, and is generally able to resolve vortex structures at higher resolutions and frequencies.

I have based much of my implementation off of Spalarts paper, “Vortex Methods for Separated Flows”. The impermeability and no slip boundary conditions are enforced using a boundary element method. The vorticity contained within the elements is then released into the flow each time step. My particular innovation is that I have used linear vortex panels to enforce the boundary conditions instead of point vortex panels. The distribution of vortices is then converted into discrete point vortices of equal strength and a random-walk algorithm is used for viscous diffusion. Since each particle requires the same amount of computational resources to keep track of, it makes intuitive sense to give them equal strengths and distribute them non-uniformly to account for the changes in vorticity. This also benefits the random-walk algorithm, which becomes a better approximation as the number of particles increases. Particles are naturally allocated exactly where they are needed most.

My approach does not seem to be common in the literature, and I suspect the reason is that it results in significant core overlap between particles in regions of high vorticity. Any attempts to correct this leaves significant gaps between vortex cores. Both core overlap and core gaps lead to errors in the velocity calculation using Biot-Savart kernels, but these errors seem to shrink as the number of particles increases. A more thorough review of the literature may help to further validate my implementation, but its advantages seem to outweigh its problems.

Each time step, the “contribution” of each particle to the velocity of every other particle must be computed. It’s effectively an N-body problem, similar to the simulation of point masses in astrophysics or point charges in electromagnetism. The time complexity of N-body problems is typically $O(N^2)$, but with the invention of the Fast-Multipole Method, this can be reduced dramatically to O(NlogN) or faster. The fast multipole method uses multipole expansions to clump particles together that are far away from a reference point, with a loss in accuracy determined by the number of truncated terms in the expansion. When applied to an adaptive quadtree, the method is particularly effective. After implementing this method, I have found that typical runs complete hundreds of times faster. There are two main variants of the Fast Multipole Method. I have used the variant that only uses outer expansions, but the second variant uses both outer and inner expansions and may result in significant improvements in computing time. The inner expansions variant is more complicated and currently eludes me, but it is something I would like to explore at some point.

One of the most difficult parts of vortex particle methods seems to be the pressure calculation. Surface pressures can be related to the vorticity released into the flow each time step, but depending on the diffusion algorithm used, this can be extremely noisy. For flow over a circular cylinder, I was able to obtain pressures comparable to the literature, but for airfoils, my results were generally poor. It seems my current implementation may only be useful for evaluating aerodynamic loads in cases of extreme flow separation. This may be remedied by either computing surface pressures using a different technique or using a different diffusion algorithm.

Static Aeroelasticity Code

In the design of aircraft, it is often assumed that the structural and aerodynamic properties are only loosely coupled. The aerodynamics determines the loading of the structure, but the structure does not deform enough to affect the aerodynamics. This assumption is rapidly becoming outdated as the industry moves towards lighter, more compliant structures. Therefore, it is becoming more and more valuable for the aerospace engineer to build intuitions about fluid-structure interaction.

Most aerospace structures are governed by the equations of elasticity. These consist of strain-displacement equations that geometrically relate displacement to strain, constitutive equations such as Hooke’s law that relate strain to stress, and equations of motion that relate the internal stresses and applied loads to the motion at every point throughout the structure. If the applied loads are not time dependent, the steady-state motion of the structure is negligible, and the equations of motion reduce to equations of static equilibrium. For structures of sufficiently large aspect ratio, the deformation can be decomposed into six deformation modes of the structure cross-section along its length, three translation modes and three rotation modes. This is the foundation of classical beam theory, an effective tool for the analysis of many structures.

Classical beam theory neglects shear deformations. A proper treatment of shear deformations leads to deformation that can not be decomposed into the existing deformation modes. Timeshenko beam theory attempts to correct for the shear deformation due to lateral loading by modifying the existing deformation modes so that the strain energy from shear deformations is accounted for. The theory of Saint Venant addresses the shear deformations due to torsional loads, but it includes a seventh deformation mode that must remain unconstrained. Vlasov beam theory properly accounts for torsional shear deformation without the limitations of Saint Venant.

The seventh deformation mode in the Saint Venant and Vlasov theories is an out-of-plane warping. While the integrated out-of-plane deformation over a warping cross-section will be zero, the local deformation will be some non-zero distribution derived from the shear stress distribution. If the cross-section is assumed to be made up of thin-walls, the concept of shear flow can be used to simplify the solution for the shear stress distribution. If this warping is constrained, the Vlasov theory predicts an added torsional stiffness called the warping stiffness. The warping stiffness has a particularly large effect for beams with open cross-sections, smaller aspect ratios, and smaller shear-to-elastic modulus ratios.

For my Static Aeroelasticity code, I have employed a Timeshenko-Vlasov beam theory coupled with a lifting-line aerodynamic model. This is more accurate than the classical aeroelasticity approach, which uses classical beam theory coupled with a strip theory aerodynamic model. In the limit of large aspect ratios, both models yield a similar value for the divergence dynamic pressure, suggesting that my code successfully describes the physical phenomenon. Additionally, my code yields results for the lift, induced drag, and internal stresses of the wing that are more accurate than a decoupled aerodynamic or structural analysis of similar fidelity.

Gravitational N-Body Code with Collisions

A distribution of mass in space-time is governed by Einstein's theory of General Relativity. In the limit of small gravitational potential and small velocities, this reduces to Newtonian gravity. Discretizing the distribution of mass into a large number of point masses leads to a large number of coupled Ordinary Differential Equations known as the classical gravitational N-body problem. Each mass contributes to the gravitational potential energy throughout the entire domain. The gradient of this potential energy is the gravitational force-field, a function of space and time that defines the force on every mass. The force on each mass can also be described by directly summing the force contribution from every other mass. Once the force acting on each mass is known, the acceleration of each mass is found using Newton's second law. The acceleration is then numerically integrated to yield the velocity and position of each mass. In my code, I have employed a simple first-order Euler method for numerical integration.

One of the challenges of this problem is encountered when the point masses pass close to each other. The force of gravity between two masses is proportional to the inverse of the distance between them squared. This means that singularities exist in the solution where the force between two masses may go to infinity. If the time integration was exact, this would not be as much of an issue because the infinite force would only apply for an infinitesimal amount of time. The large velocity of the masses involved would almost immediately drop back to reasonable values as they moved through each other. Since the time integration is numerical, there is significant potential for large errors to accrue near these singular points, causing masses to launch each other apart at ridiculous speeds. Higher-order time integration and adaptive time-stepping are techniques that can partially help avoid this problem. Another solution might be to include a cutoff function or mollifier, so that masses within some small distance experience a dramatic dropoff in force between them. However, one might argue that the best solutions to this problem would be physically based, and not strictly numerical.

In reality, masses are not able to pass through each other so effortlessly. The masses take up space, and they will often collide if they get too close. Large scale N-body systems may employ the use of Smoothed-Particle Hydrodynamics to model these collisions. The point masses then behave much like a fluid in regions where the density is large, with a hydrodynamic pressure keeping them reasonably spaced as they clump together. A simpler approach is to treat the collisions as discrete instantaneous events where momentum is exchanged. This is the approach I have taken. A coefficient of restitution is also included to account for losses in kinetic energy due to heat and deformation. With a mechanism for energy dissipation, the masses accrete and form interesting structures, even with only a couple thousand masses.

Event-Driven Hard Sphere Molecular Dynamics Code (WIP)

The interaction of two neutral molecules is often described well with a Lennard-Jones potential. The gradient of this potential gives the force between the molecules. At close distances the molecules will strongly repel, and at farther distances the molecules will weakly attract. In the limit of large relative velocities and low density, these interactions only affect the trajectories of the molecules significantly during the short time of close proximity. The weak attraction can be approximated as no attraction, and the

strong repulsion can be approximated as an infinite discontinuous repulsion. These changes, which effectively morph the Lennard-Jones potential into a step function, describe the forces acting on hard spheres undergoing perfectly elastic collisions. The trajectory of a molecule from one collision to another can be determined exactly with no need for numerical time integration, so the simplified dynamics of hard spheres enables an extremely efficient and exact solution. This is known as event-driven molecular dynamics.

There are three key ingredients in a typical event-driven molecular dynamics solution. The first is the particle information array. This contains all of the information about the state of every particle in the system. The position and velocity vectors, the radius, and the mass are some examples of values stored for each particle. The most recent collision time for each particle may also be included here. The second ingredient is the cell structure. The cell structure partitions the domain so that particles are able to check for collisions with neighbors more quickly. Without the cell structure, each particle would have to check for many more collisions to ensure that none are missed. The final ingredient is the event queue. The event queue is an ordered list that holds information about possible upcoming events and ensures they occur in the correct order. Events can be collisions, transfers between cells, checks for new collisions, or any other event that might need to take place. Events are constantly updated as the states of the particles change and many events become quickly outdated, so each particle may have multiple events scheduled to save time.

My code is based on the particular method of Sigurgeirsson in the paper “Algorithms for Particle-Field Simulations with Collisions”. The event queue is a binary tree with the heap property. Each particle has two events scheduled at all times and the heap is keyed by the one with the smaller event time. For each iteration, the event at the top of the queue is executed, and any events affected by this are heapified at the end of the iteration. Thousands of events can be computed each second, even on mediocre hardware. This enables the solution of large numbers of molecules undergoing millions of collisions. These scales, while still only microscopic at best in most cases, are large enough to observe continuum behavior. My code is still a work-in-progress and does not currently support many of the kinds of experiments I would like to explore, but I have many plans for this tool. Some possible applications include micro/nanofluidics and the computing of atmospheric drag on satellites or other objects moving through rarefied media. Additionally, a series of animations demonstrating concepts in fluid mechanics and thermodynamics using the hard sphere model may provide a more fundamental intuition of these subjects for students.