

Computational  
Fluid  
Dynamics

Brady  
Metherall,  
100516905

Introduction

Theory

SU<sup>2</sup> Code

Scripting and  
Automation

Results

Conclusion

# Computational Fluid Dynamics using the SU<sup>2</sup> Code

Brady Metherall    100516905

Thursday April 7, 2016

# Introduction

Computational  
Fluid  
Dynamics

Brady  
Metherall,  
100516905

Introduction

Theory

SU<sup>2</sup> Code

Scripting and  
Automation

Results

Conclusion

- Over the past few decades, a new discipline of fluid dynamics has emerged, computational fluid dynamics.
- Computational fluid dynamics is essentially a perfect wind tunnel.
- Advantages include cost, speed, ease of use, and range of operation.
- Furthermore, computational fluid dynamics has the ability to provide a broader range of data, as well as simulate impractical situations to test in a wind tunnel.
- Applications in many areas of physics and engineering.

# Laminar Flow Equations

Computational  
Fluid  
Dynamics

Brady  
Metherall,  
100516905

Introduction  
Theory  
SU<sup>2</sup> Code  
Scripting and  
Automation  
Results  
Conclusion

- Incompressibility Condition:  $\nabla \cdot \mathbf{u} = 0$
- Euler's Equations:  $\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \cdot \nabla) \mathbf{u} = -\frac{1}{\rho} \nabla p + \mathbf{g}$
- Cauchy's Equation:  $\rho \left( \frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \cdot \nabla) \mathbf{u} \right) = \nabla \cdot \mathbf{T} + \rho \mathbf{g}$
- Stress Tensor:  $T_{ij} = -p\delta_{ij} + \mu \left( \frac{\partial u_j}{\partial x_i} + \frac{\partial u_i}{\partial x_j} \right)$
- Navier-Stokes Equation:

$$\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \cdot \nabla) \mathbf{u} = -\frac{1}{\rho} \nabla p + \nu \nabla^2 \mathbf{u} + \mathbf{g}$$

# Reynolds-Averaged Navier-Stokes (RANS)

Computational  
Fluid  
Dynamics

Brady  
Metherall,  
100516905

Introduction  
Theory  
SU<sup>2</sup> Code  
Scripting and  
Automation  
Results  
Conclusion

To make solving the Navier-Stokes equations easier, especially for turbulent flows, we can assume the flow is the superposition of the laminar, and turbulent flows, and then take the time average.

$$\alpha_i \equiv \overline{\alpha_i} + \alpha'_i,$$

$$\text{where } \overline{\alpha_i} = \frac{1}{\tau} \int_0^\tau \alpha_i dt$$

$$\overline{\alpha'_i} = 0, \quad \frac{\partial \overline{\alpha_i}}{\partial t} = 0, \quad \overline{\overline{\alpha_i}} = \overline{\alpha_i}, \quad \text{and} \quad \frac{\partial \overline{\alpha_i}}{\partial x_i} = \frac{\partial \overline{\alpha_i}}{\partial x_i}$$

Making the appropriate substitutions, taking the time average, and using the properties we obtain the Reynolds-Averaged Navier-Stokes equation,

$$(\bar{\mathbf{u}} \cdot \nabla) \bar{\mathbf{u}} + \overline{(\mathbf{u}' \cdot \nabla) \mathbf{u}'} = -\frac{1}{\rho} \nabla \bar{p} + \nu \nabla^2 \bar{\mathbf{u}} + \bar{\mathbf{g}}$$

# Spalart-Allmaras Turbulence

Computational  
Fluid  
Dynamics

Brady  
Metherall,  
100516905

Introduction

Theory

SU<sup>2</sup> Code

Scripting and  
Automation

Results

Conclusion

- A model is still needed for  $\nu$
- According to the Boussinesq hypothesis, an increase in the viscosity gives the effect of turbulence
- Therefore  $\nu = \bar{\nu} + \nu'$
- One such way to model the turbulence, is the Spalart-Allmaras turbulence model

$$\nu' = \hat{\nu} f_{v1}; \quad f_{v1} = \frac{\chi^3}{\chi^3 + c_{v1}^3}; \quad \chi \equiv \frac{\hat{\nu}}{\bar{\nu}}.$$

Assuming the flow is incompressible, and the diffusivity is constant,  $\hat{\nu}$  is obtained by solving the convection-diffusion equation,

$$\frac{\partial \hat{\nu}}{\partial t} = D \nabla^2 \hat{\nu} - \bar{\mathbf{u}} \nabla \hat{\nu},$$

where  $D$  is the diffusivity.

# SU<sup>2</sup> Code Development

Computational  
Fluid  
Dynamics

Brady  
Metherall,  
100516905

Introduction

Theory

SU<sup>2</sup> Code

Scripting and  
Automation

Results

Conclusion

- The Stanford University Unstructured (SU<sup>2</sup>) code is a relatively new, open source, constrained partial differential equation solver, optimized for computational fluid dynamics problems.
- It was initially developed by the Aerospace Design Laboratory of the Department of Aeronautics and Astronautics at Stanford University.
- SU<sup>2</sup> is written mostly in C++; and because it is open source, the capabilities have since been extended by many teams of developers from around the world.
- The developers believe that it should remain open source, portable, and highly encapsulated while operating at a high performance standard.

# SU<sup>2</sup> Code Capabilities

Computational  
Fluid  
Dynamics

Brady  
Metherall,  
100516905

Introduction

Theory

SU<sup>2</sup> Code

Scripting and  
Automation

Results

Conclusion

Some of the features of the SU<sup>2</sup> code include, but are not limited to:

- Reynolds-Averaged Navier-Stokes equations
- Spalart-Allmaras turbulence model ( $c_{v1} = 7.1$ )
- Menter Shear Stress Transport model (an alternative to the SA model)
- Compressible flows
- Wave equation, heat equation, and plasma equations
- Optimal shape design (i.e. Supersonic aircraft)
- Dynamic meshes

# SU<sup>2</sup> Code

Computational  
Fluid  
Dynamics

Brady  
Metherall,  
100516905

Introduction

Theory

SU<sup>2</sup> Code

Scripting and  
Automation

Results

Conclusion

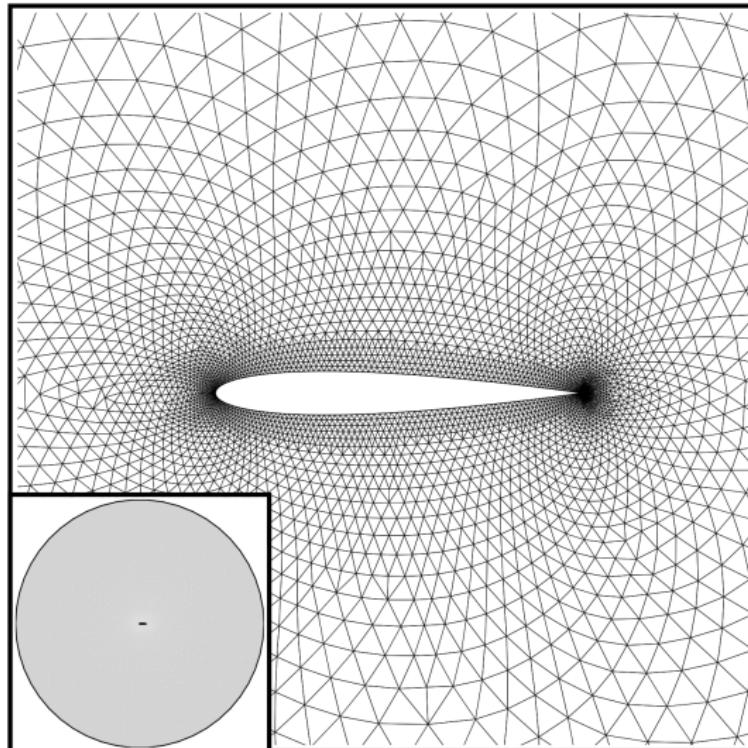
- All of the parameters such as the fluid speed, Reynolds number, and governing equations are specified in the configuration file.
- The shape geometry of the problem is stored as an unstructured mesh – hence the name – in the mesh file.
- The mesh is unstructured in the sense that there is no algebraic relation between nodes.
- Due to the way the solution is discretized, the mesh can dynamically be updated; this is a key feature in the shape optimization process of SU<sup>2</sup>.

# Example Mesh File

Computational  
Fluid  
Dynamics

Brady  
Metherall,  
100516905

Introduction  
Theory  
**SU<sup>2</sup> Code**  
Scripting and  
Automation  
Results  
Conclusion



# Scripting and Automation

Computational  
Fluid  
Dynamics

Brady  
Metherall,  
100516905

Introduction  
Theory  
SU<sup>2</sup> Code  
Scripting and  
Automation  
Results  
Conclusion

- Wolfram Mathematica 10.1.0 for Linux x86 was used to sift through the vast data files and extract the relevant information to produce the plots.
- A Wolfram function was needed to convert the data to a form Mathematica could use.
- Mathematica's ListDensityPlot function was used to create the images.
- The function to tidy the data along with the plotting function were combined into a Wolfram script which can be executed from the terminal.
- To fully automate the image generation, a shell script was written to iterate over each data file.

# Airfoil

Computational  
Fluid  
Dynamics

Brady  
Metherall,  
100516905

Introduction

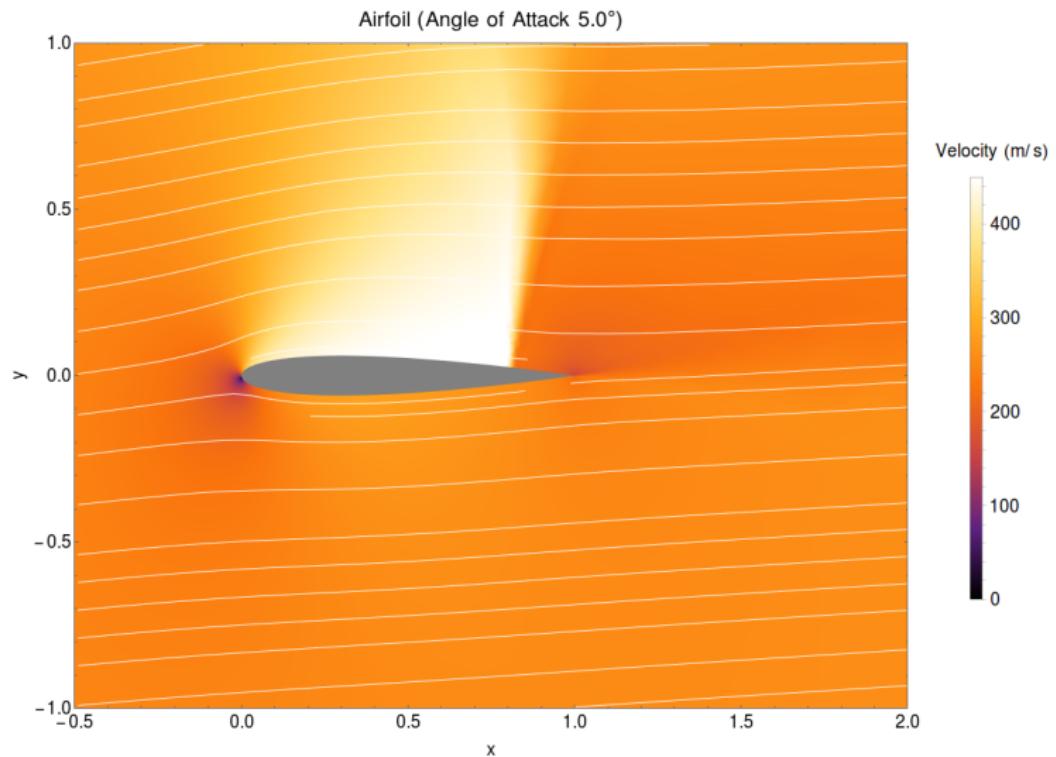
Theory

SU<sup>2</sup> Code

Scripting and  
Automation

Results

Conclusion



# Pitching Airfoil

Computational  
Fluid  
Dynamics

Brady  
Metherall,  
100516905

Introduction

Theory

SU<sup>2</sup> Code

Scripting and  
Automation

Results

Conclusion

# Laminar Flow Around a Static Cylinder

## Euler

Computational  
Fluid  
Dynamics

Brady  
Metherall,  
100516905

Introduction

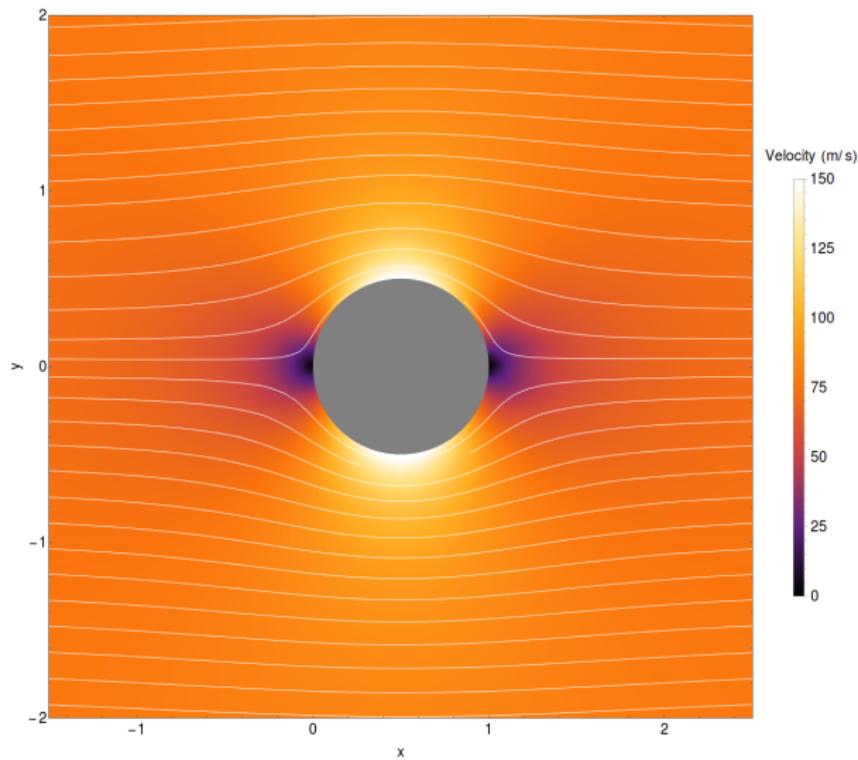
Theory

SU<sup>2</sup> Code

Scripting and  
Automation

Results

Conclusion



# Laminar Flow Around a Static Cylinder

## Euler Boundary

Computational  
Fluid  
Dynamics

Brady  
Metherall,  
100516905

Introduction

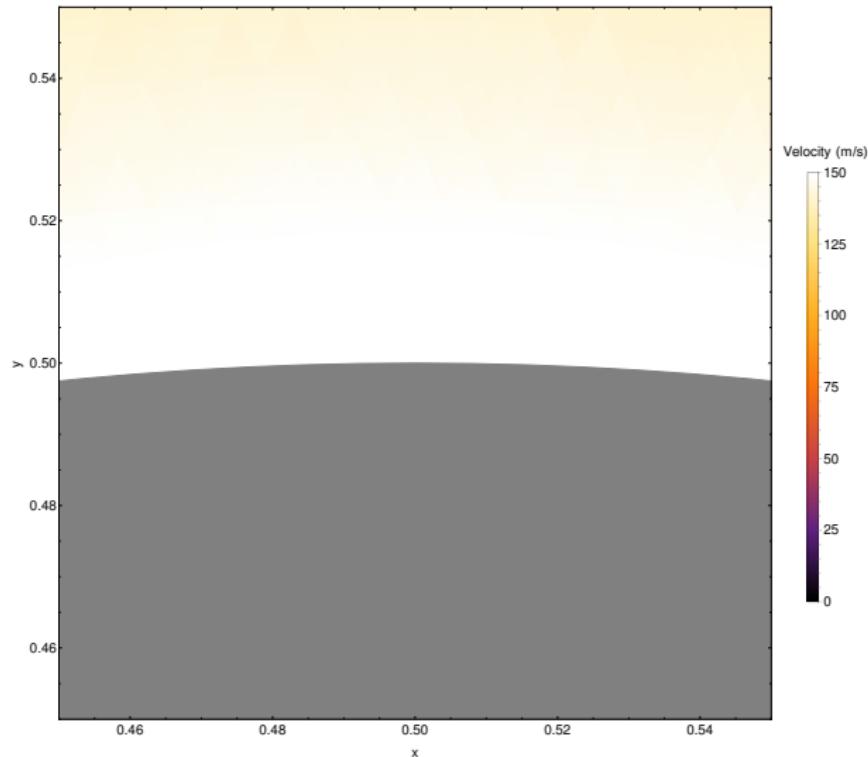
Theory

SU<sup>2</sup> Code

Scripting and  
Automation

Results

Conclusion



# Laminar Flow Around a Static Cylinder

## Navier-Stokes

Computational  
Fluid  
Dynamics

Brady  
Metherall,  
100516905

Introduction

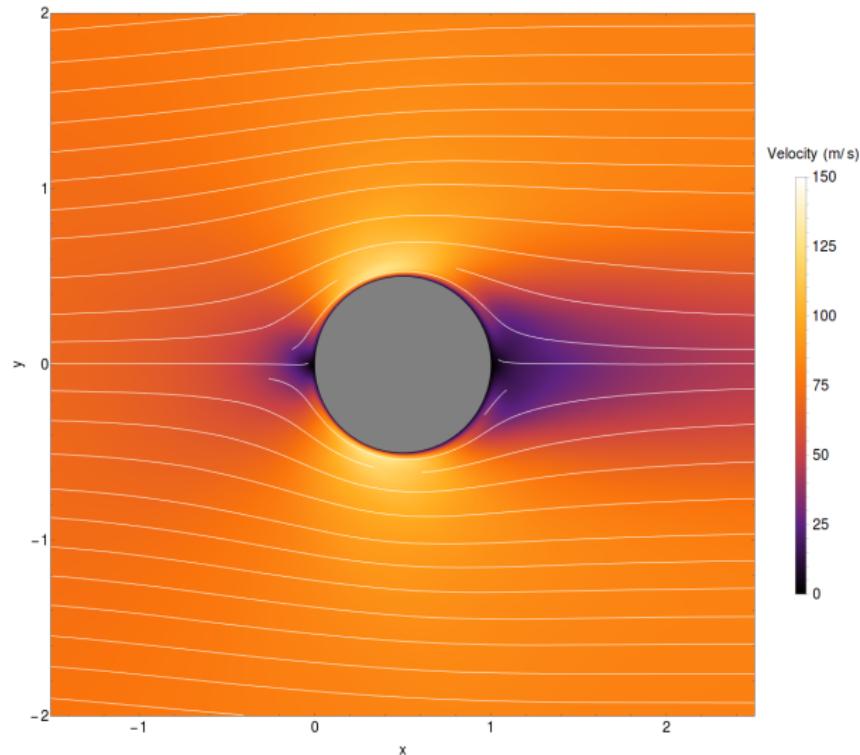
Theory

SU<sup>2</sup> Code

Scripting and  
Automation

Results

Conclusion



# Laminar Flow Around a Static Cylinder

## Navier-Stokes Boundary

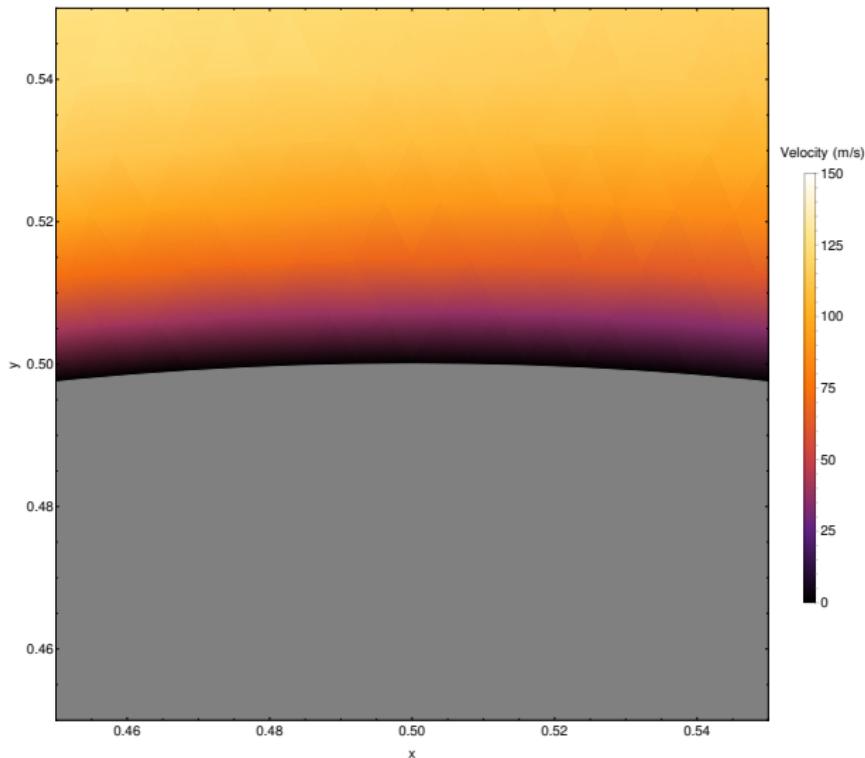
# Computational Fluid Dynamics

Brady  
Metherall,  
100516905

Introduction

SU<sup>2</sup> Code

## Results



# Turbulent Flow Around a Static Cylinder

## Vortex Shedding

## Computational Fluid Dynamics

Brady  
Metherall,  
100516905

## Results

# Conclusion

Computational  
Fluid  
Dynamics

Brady  
Metherall,  
100516905

Introduction  
Theory  
SU<sup>2</sup> Code  
Scripting and  
Automation  
Results  
Conclusion

This project examined the theory behind fluid dynamics, as well as numerical simplifications that can be made to optimize the computation. To acquire the visualizations, a Wolfram script along with Wolfram Mathematica was used to parse the data files and create the graphics. The preliminary results in this project are in agreement with the theoretical predictions. SU<sup>2</sup> is a viable computational fluid dynamics simulator – however, only a small set of its capabilities were used in this project. We were able to simulate a von Kármán vortex street behind a static cylinder using the Navier-Stokes equations, and the Spalart-Allmaras turbulence model. Computational fluid dynamics is now a critical discipline of fluid dynamics, and it is only going to become more indispensable as computations become faster, and the numerical algorithms become further optimized.