Course Data

Class number:	25737/11702
Days and Time:	M/W 3:00pm - 4:15pm
Location:	Tempe - SCOB210
Online:	Blackboard class site
Instructor:	Marcus Herrmann
Office:	ERC 311
Phone:	965-7291
E-mail:	marcus.herrmann@asu.edu
Office hours:	M/W: 9:30-10:30am; T/Th: 3-4pm; Fr: 11am-12pm no office hours on: 1/19; 3/9-18; 3/20
Textbook:	- Computational Fluid Dynamics (Vol. 1) by K. A. Hoffmann & S. T. Chiang - class notes (available on Blackboard for each class)
Optional Reading:	- Computational Methods for Fluid Dynamics by J. H. Ferziger & M. Peric - Engineering Numerical Analysis by P. Moin
Prerequisites:	- AEE471: Engineering BS/BSE students: MAE 340/AEE360 (C or better), MAE 384 (C or better) - MAE561: Graduate Engineering student

Course Outline

1. Introduction
2. Governing equations and boundary conditions; Classification of PDEs
3. Finite difference formulas; Truncation error
4. Elliptic equations and solution of linear systems using iterative methods
5. Accuracy, consistency, and stability of differencing schemes
6. Solving parabolic equations
7. Solving hyperbolic equations
8. The incompressible Navier-Stokes equations
9. Verification & validation; Method of Manufactured Solutions
10. Advanced topics

Course Objectives and Expected Learning Outcomes

Students should be able to classify PDEs, derive finite difference formulas, determine truncation errors, code solvers for and solve elliptic, parabolic, and hyperbolic equations and systems of linear equations using time advancement explicit and implicit schemes and iterative methods, determine the accuracy, consistency and stability of differencing schemes, code solvers for and solve the incompressible Navier-Stokes equations in two dimensions using a fractional step method, perform code verification using the Method of Manufactured Solutions, perform solution verification, and perform code validation.

Grade Policies

Grades are calculated using the following weighted sum of scores obtained in three assignment categories:

Homework:	50%
In-Semester Project:	20%
Final Project:	30%

Each homework assignment is not necessarily weighted equal. The letter grade associated with the final weighted score will be determined depending on the overall performance of the classes.

Homework and project assignments will be posted on the class' Blackboard site only, within the Content section. Please ensure that you have access to the class' Blackboard site. Due dates for each homework, in-semester project, and the final project as well as submission instructions will be announced in class and are stated on the posted assignment sheets.

All requested programs/codes have to be submitted through Blackboard's SafeAssign mechanism. They must be well documented by comments in the source code. Using these comments, one must be able to understand the purpose of individual code segments, i.e. for loops, if statements, subroutines, etc. Uncommented code will incur grading penalties. Failure to submit your code through Blackboard's SafeAssign mechanism may result in no credit given for the respective entire assignment.

All homework/projects must be submitted on the due date. No credit will be given for work submitted late. If valid circumstances (see also absence policy) prevent you from submitting your work on time, please talk to me BEFORE the submittal deadline for a potential extension of the due date.

Some homework and projects may contain sections mandatory for students taking MAE561 which are optional for extra credit for students taking AEE471. In addition, some homework and projects may contain bonus assignments for extra credit for students taking either MAE561 or AEE471. These bonus assignments can be used to increase your score in each category (homework, in-semester project, final project) separately. Bonus points do not carry over from one category to another. Category scores are capped at 100%.

Quizzes for bonus points may be given in class without prior notice.

Students taking AEE471 have to meet several core course outcomes outlined in a separate document students need to sign at the beginning of the semester. A grade of C or better, regardless of the overall course score, can only be assigned if all the core outcomes have been met. Assignments will contain problems marked as pertaining to individual core course outcomes. To meet an outcome, AEE471 students must score 70% or better in each course core outcome category. A makeup problem may be assigned to AEE471 students failing to meet core course outcomes near the end of the semester. AEE471 students can track their core outcome demonstration status on Blackboard.

If you have questions or complaints about the grading of individual homework or projects, please come see me during my office hours to discuss your concerns.

Course Outcome	ABET (a-k)	Level of Mastery
Students will derive finite difference approximations for first and second order derivatives for given stencil points and identify associated truncation errors	a	application
Students will write solvers for elliptic, parabolic and hyperbolic PDEs using explicit, implicit, and iterative schemes	a, e, k	application
Students will code a 2D Burgers equation flow solver and apply it to the solution of a flow problem.	a, e, k	application
Students will code a 2D incompressible Navier-Stokes solver and apply it to the solution of a flow problem.	a, e, k	application
Students will perform solution verification and identify strategies to identify sources of error in CFD results	a, e, k	analysis

Application	
Student applies equations and scientific principles learned in a previous course to a current problem without being directed to do so. Student correctly solves basic calculus, linear algebra or differential equation without reference. a	Student selects appropriate computer application. Student writes code to solve problem k
Student identifies appropriate models for solving problem – *e.g.*, using beam model for bending of airplane wing. e	

Analysis
Student derives equations from first principles. a
Student explains restrictions, assumptions and basis for simple and complex models used in solving problems. e
Student explains limitations and restrictions of various computational tools. k

Absence Policy

Attending classes is mandatory, although no checks/sign-in sheets will be used.

Excused absences related to religious observances/practices are in accord with ACD 304–04, "Accommodation for Religious Practices".

Excused absences related to university sanctioned events/activities are in accord with ACD 304–02, "Missed Classes Due to University-Sanctioned Activities".

Expected Classroom Behavior Policy

The use of pagers, cell phones, and recording devices is **not** allowed in the classroom during class times.

Policy against Threatening Behavior

The policy against threatening behavior, per the Student Services Manual, SSM 104–02, "Handling Disruptive, Threatening, or Violent Individuals on Campus" applies.

Accommodation for a Disability

Students requesting accommodation for a disability must be registered with the Disability Resource Center (DRC) and submit appropriate documentation from the DRC.

Academic Integrity Policy

Academic integrity is mandatory, and any form of plagiarism will not be tolerated. Please see ASU's Student Academic Integrity Policy for details (https://provost.asu.edu/academicintegrity). As a reminder, it is a violation of the Academic Integrity Policy if one

- refers to materials or sources or uses devices (e.g., computer disks, audio recorders, camera phones, text messages, crib sheets, calculators, solution manuals, **materials from previous classes**, or commercial research services) not authorized by the instructor for use during the Academic Evaluation or assignment;
- possesses, reviews, buys, sells, obtains, or uses, without appropriate authorization, any materials intended to be used for an Academic Evaluation or assignment in advance of its administration.

For example, you are **not** authorized to use any solutions from homework/project assignments from prior years, including but not limited to course posted solutions and solutions of prior year students.

Although you are encouraged to talk to other students about the topics covered in class, you have to solve all homework problems and projects on your own. You are **not** permitted to collaborate with other students on **any** assignment, unless specifically authorized to do so in the written homework/project assignment.

All requested programs/codes have to be submitted through Blackboard's SafeAssign mechanism and may be checked for plagiarism using additional tools.

Some Important Dates

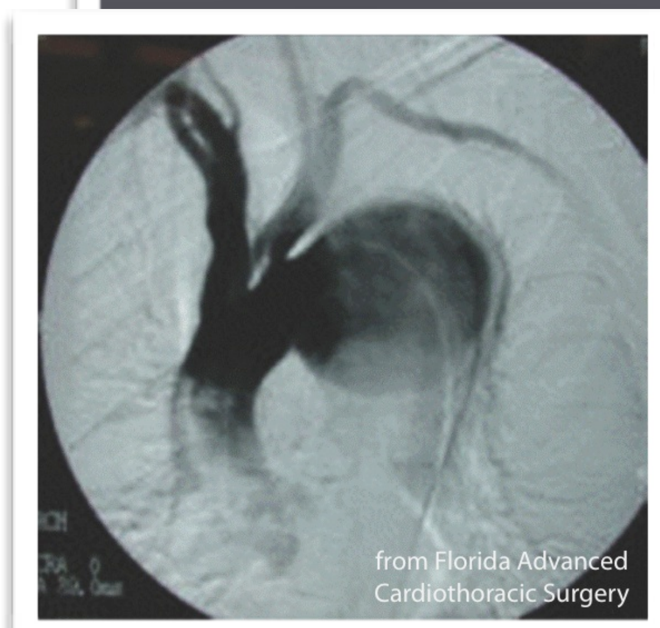
First day of class:	01/12/2015
Last day of class:	04/29/2015
Final exam:	none
No class on:	01/19 (MLK Day); 03/09-11 (Spring Break); 04/16-18

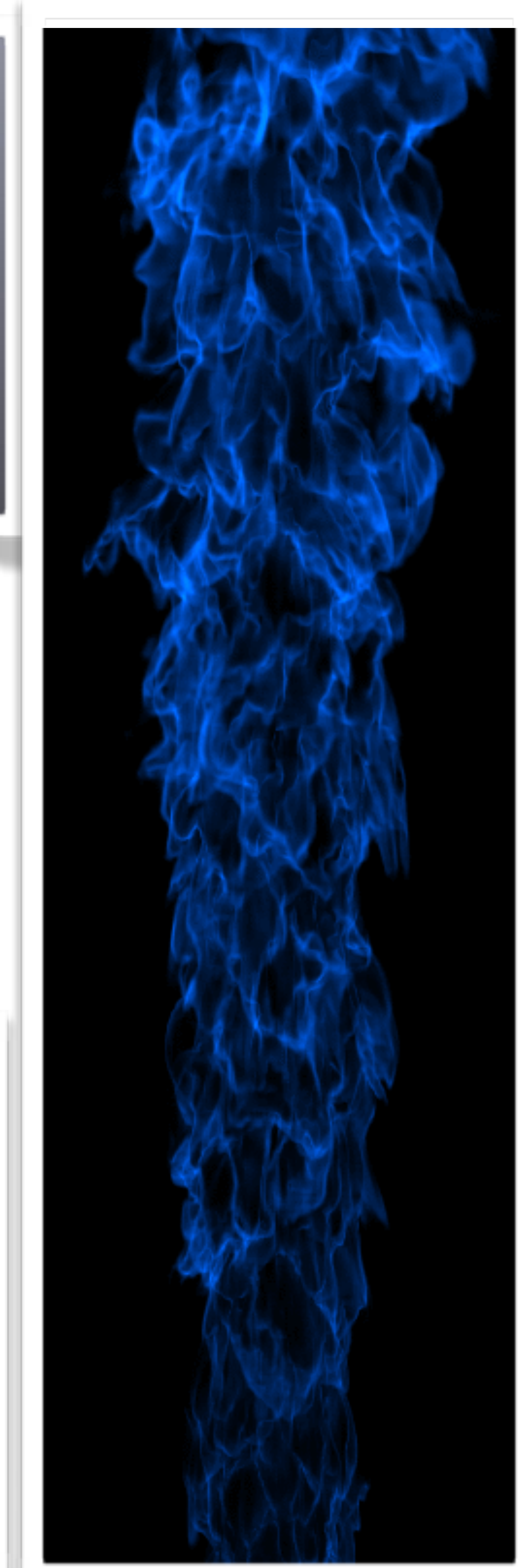
Additional Information

- On Blackboard, please complete the "Assignment Return Preference" task located in "Course Information". If you do not complete the assignment by Jan. 21st, I presume you do not wish to have your graded assignments returned to you during class. They then have to be picked up personally during office hours.
- This course requires you to write your own codes to compute various model problems. This means that everyone needs a programming environment to work in. The specific choice of programming language/environment is up to you. Matlab is one possible option, although I would strongly encourage you to use Fortran (or C/C++), since compiler based languages are more efficient the more complex the application problems become. Thus Fortran90 continues to be the programming language of choice in many CFD research codes.
- Matlab is available as a student license, or can be used free of charge on many ASU computers.
- Fortran (and C) compilers are available on the web at gcc.gnu.org. You can also use the compilers and development environments available at ASU's Online Applications (http://apps.asu.edu).
- Unix and Linux (and to some extend OS X) are the operating systems of choice for doing most non-commercial computational fluid dynamics. If you have access to these, I strongly encourage you to use them instead of Windows. OS X is available on many Apple ASU computers.

- Unlike in previous years, the course will not include the use of commercial CFD solvers. Commercial CFD solvers will be covered in a new AEE/MAE course “Applied Computational Fluid Dynamics” to be offered starting in the Fall 2015 semester.
- Homework and project assignments will be posted on the course’s Blackboard site. Please make sure that you can access the course web pages.
- If an assignment requires coding multiple functions/programs, combine these into one single text file and upload this combined file to Blackboard’s Safe Assign mechanism.
- It is CRUCIAL that you start all of your assignments as soon as possible. Do not wait until the last minute to start. No questions concerning homework or projects will be answered in the respective due date’s office hours.
- Use offered opportunities to obtain extra-credit. You will be able to learn more (and enhance your grade while doing so).
- And finally: do not hesitate to ask questions and make good use of the available office hours!

- Atomization of Liquids
- Combustion
- Fluid/Solid Interactions
- Numerical Algorithms for Massively Parallel Computer Systems





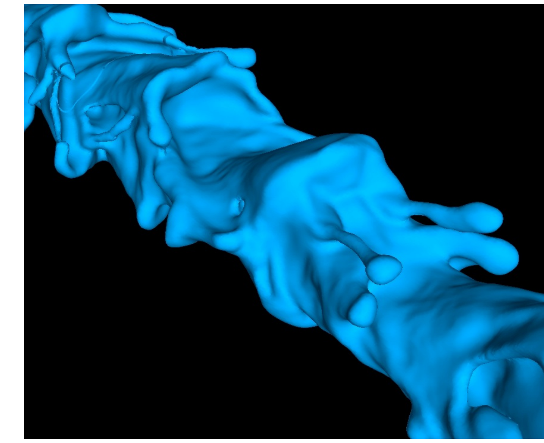
- Atomizing liquids have many applications
  - pharmaceutical sprays
  - gas exchange with the ocean
  - truck tire splash
  - fire fighting
  - reciprocal engines
  - gas turbines
  - RAM/SCRAM jets


Teva Pharmaceuticals

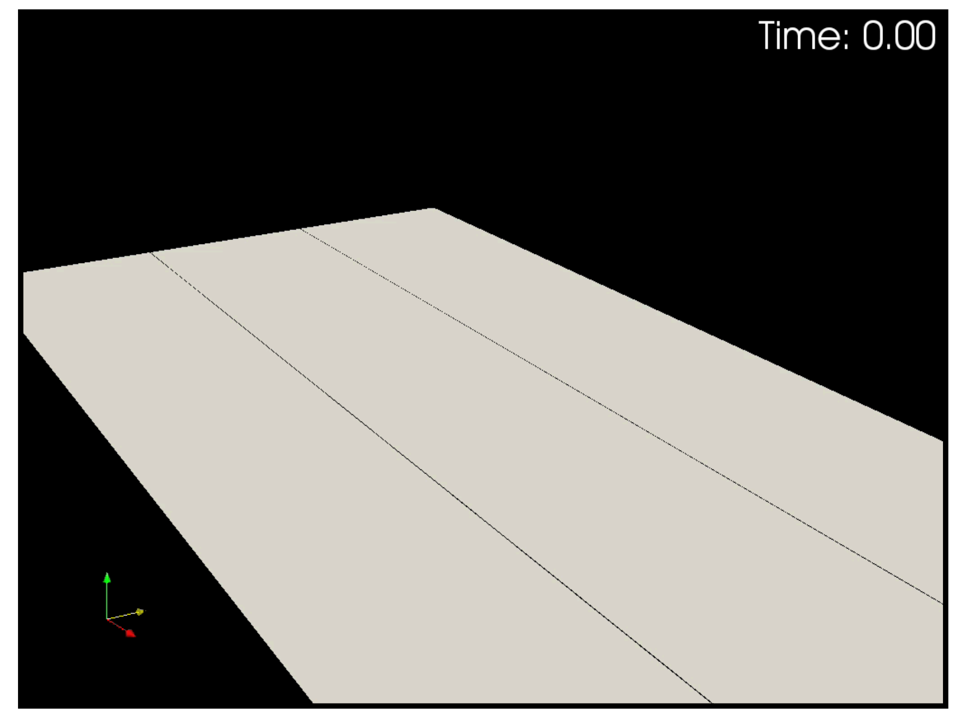

B

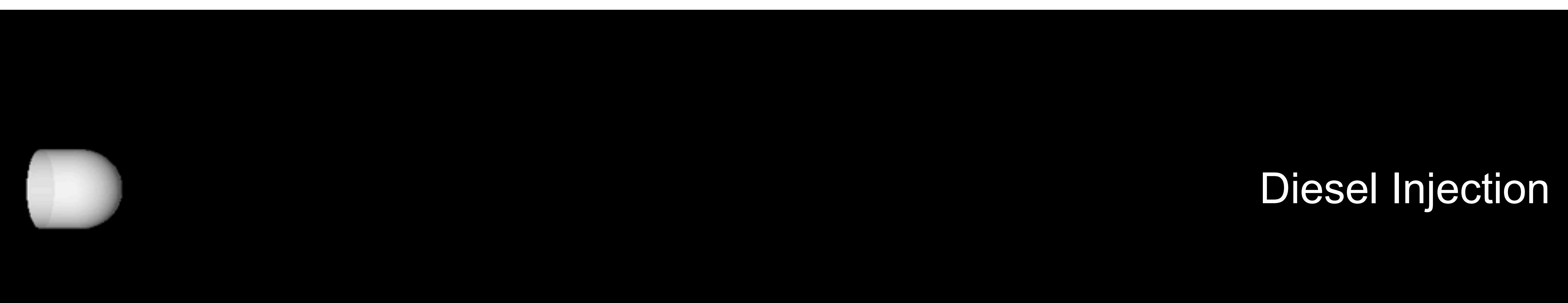
- Challenge: Predict Atomization
  - What are the details of atomization? What's the impact of temperature fluctuations or turbulence?
  - Experiments are hard and very costly
  - Perform high fidelity detailed simulations of atomization
  - Large scale simulations: billions of grid points on thousands of processors

Experiment

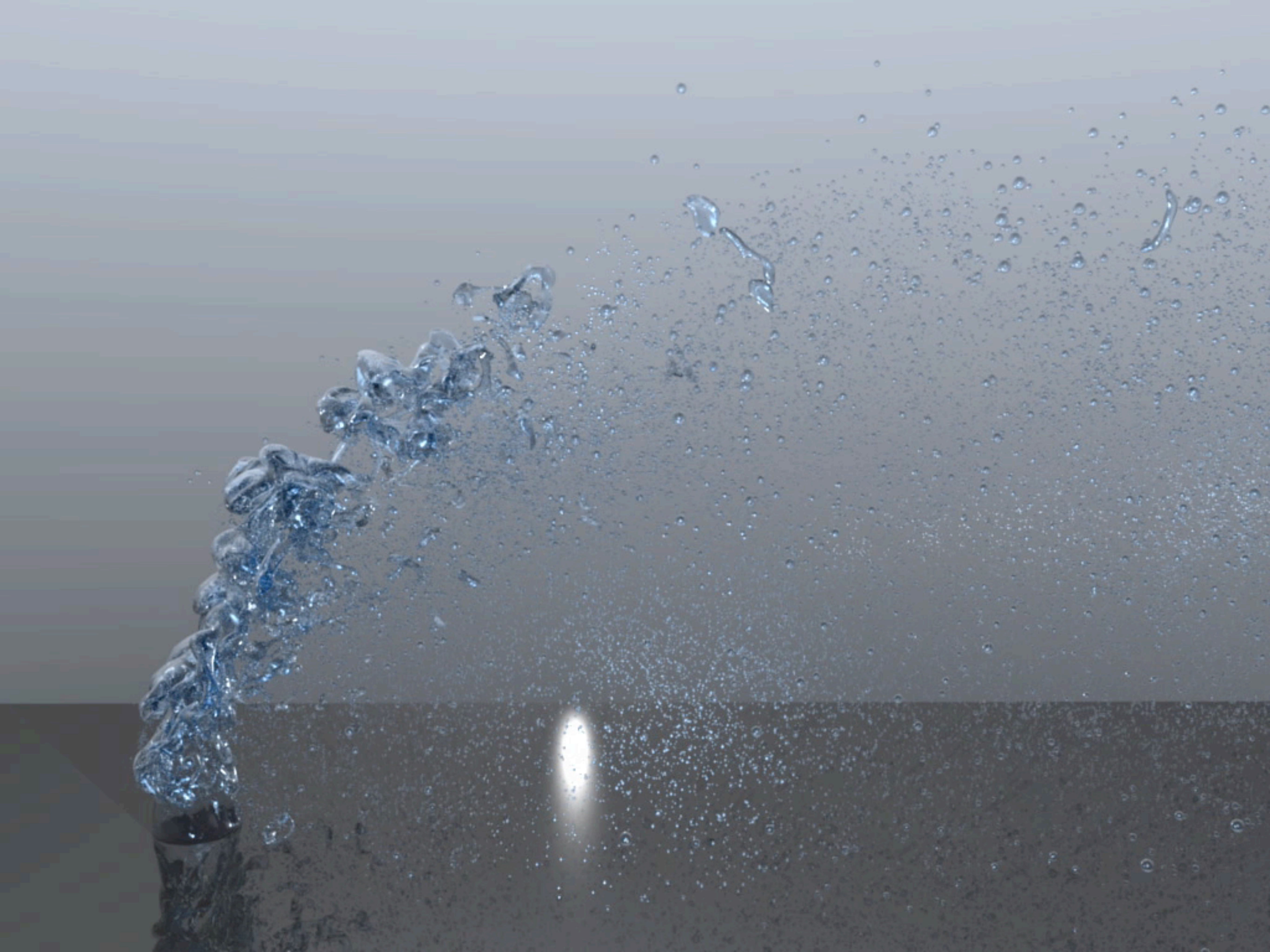


Simulation

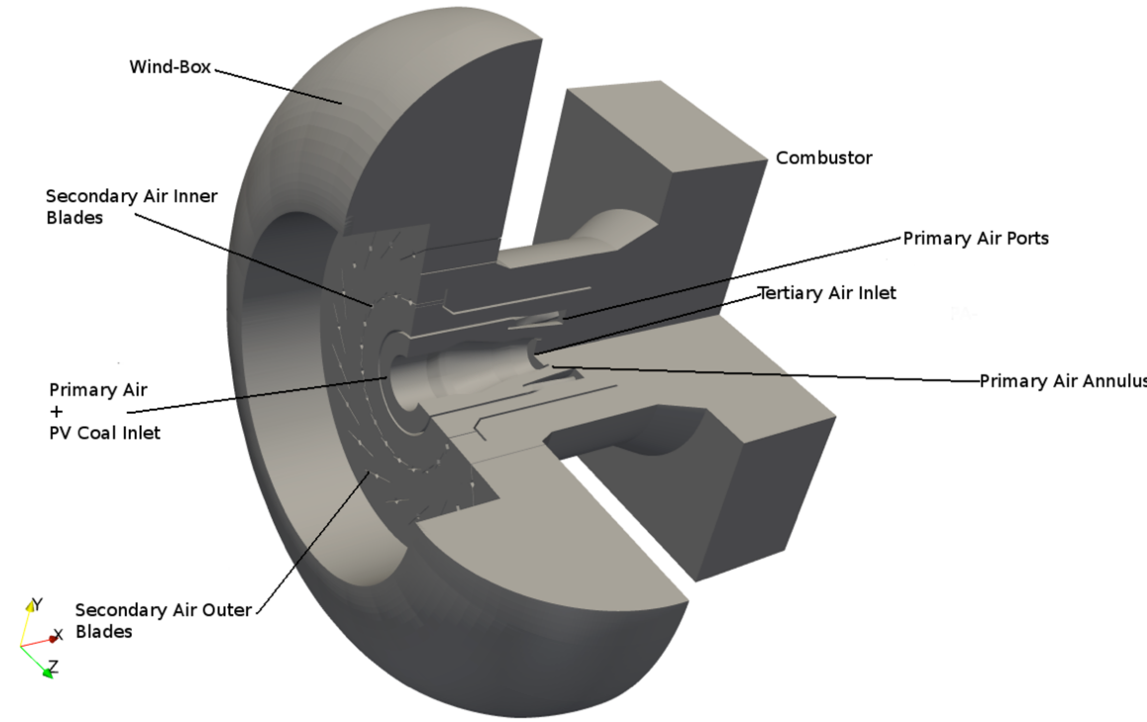


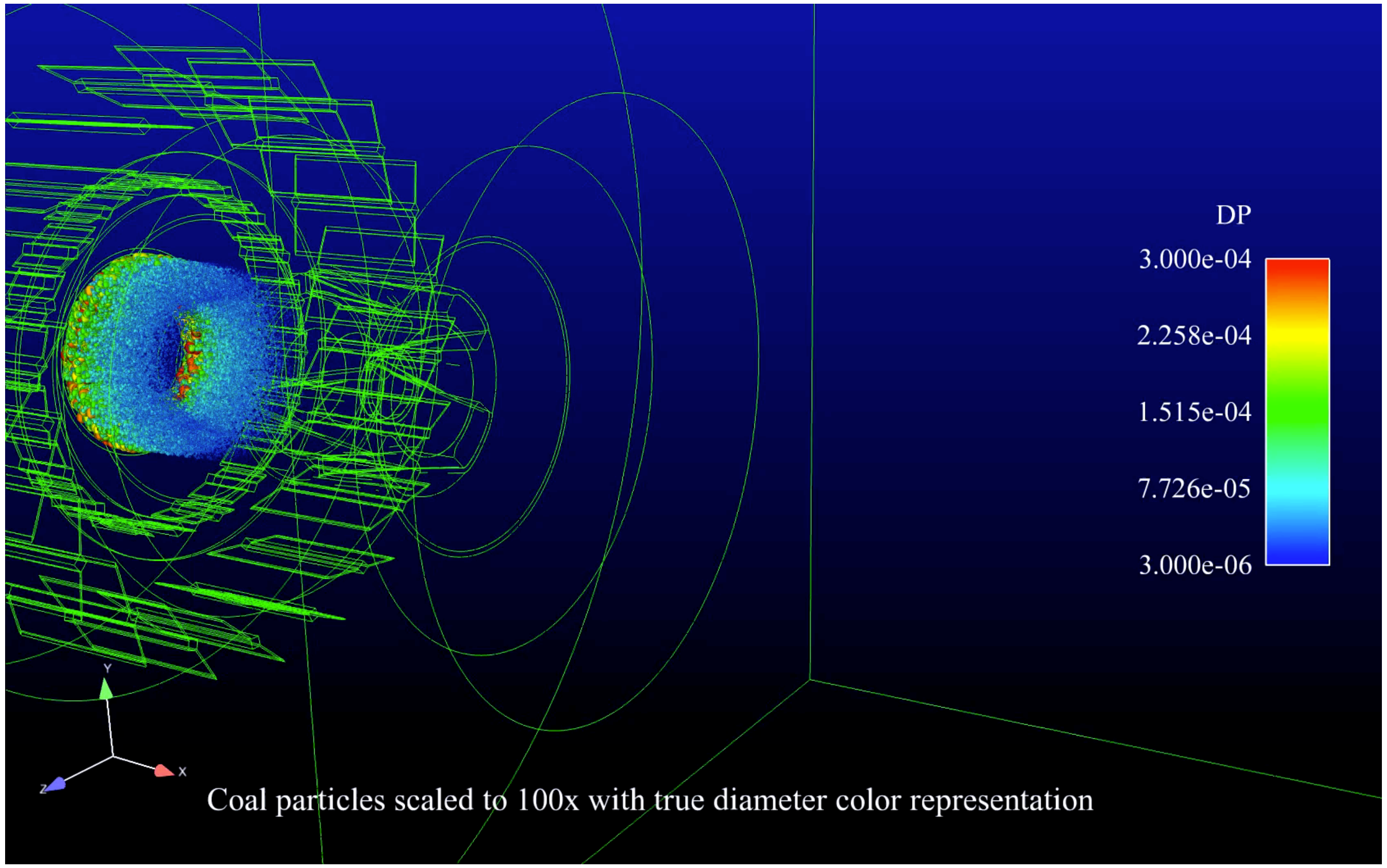


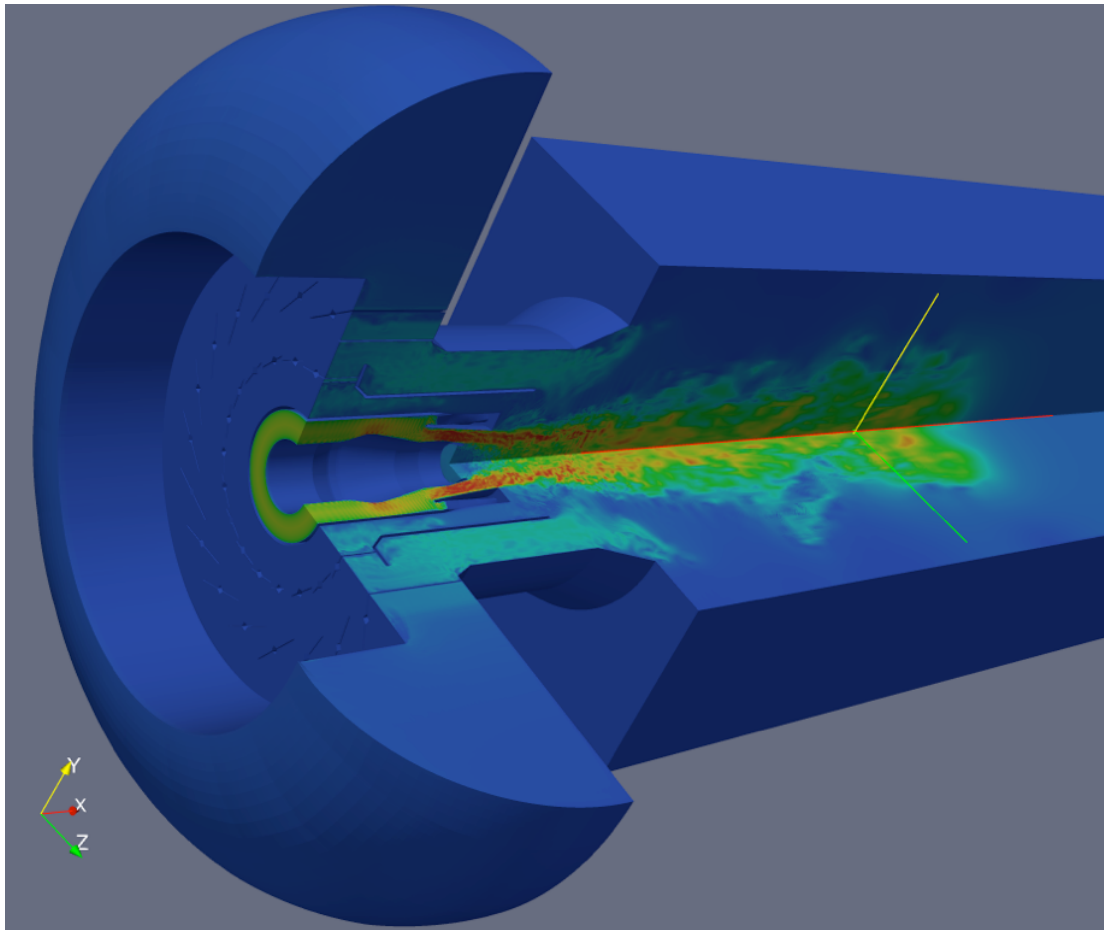
Time = 0.00 μs



- Turbulent combustion in complex geometries
  - flamelet models
  - Lagrangian particle models
  - Large Eddy Simulations







Visualization of a Non-Premixed Turbulent Flame

Sandia Flame-E

Matthias Ihme
Marcus Herrmann
Heinz Pitsch

Center for Integrated Turbulence Simulations

Stanford University

- Wax deposition processes in deep sea oil pipelines
  - Multi-scale modeling
    - micro-scale wax crystal growth
    - meso-scale wall crystal deposition
    - macro-scale pipeline blockage

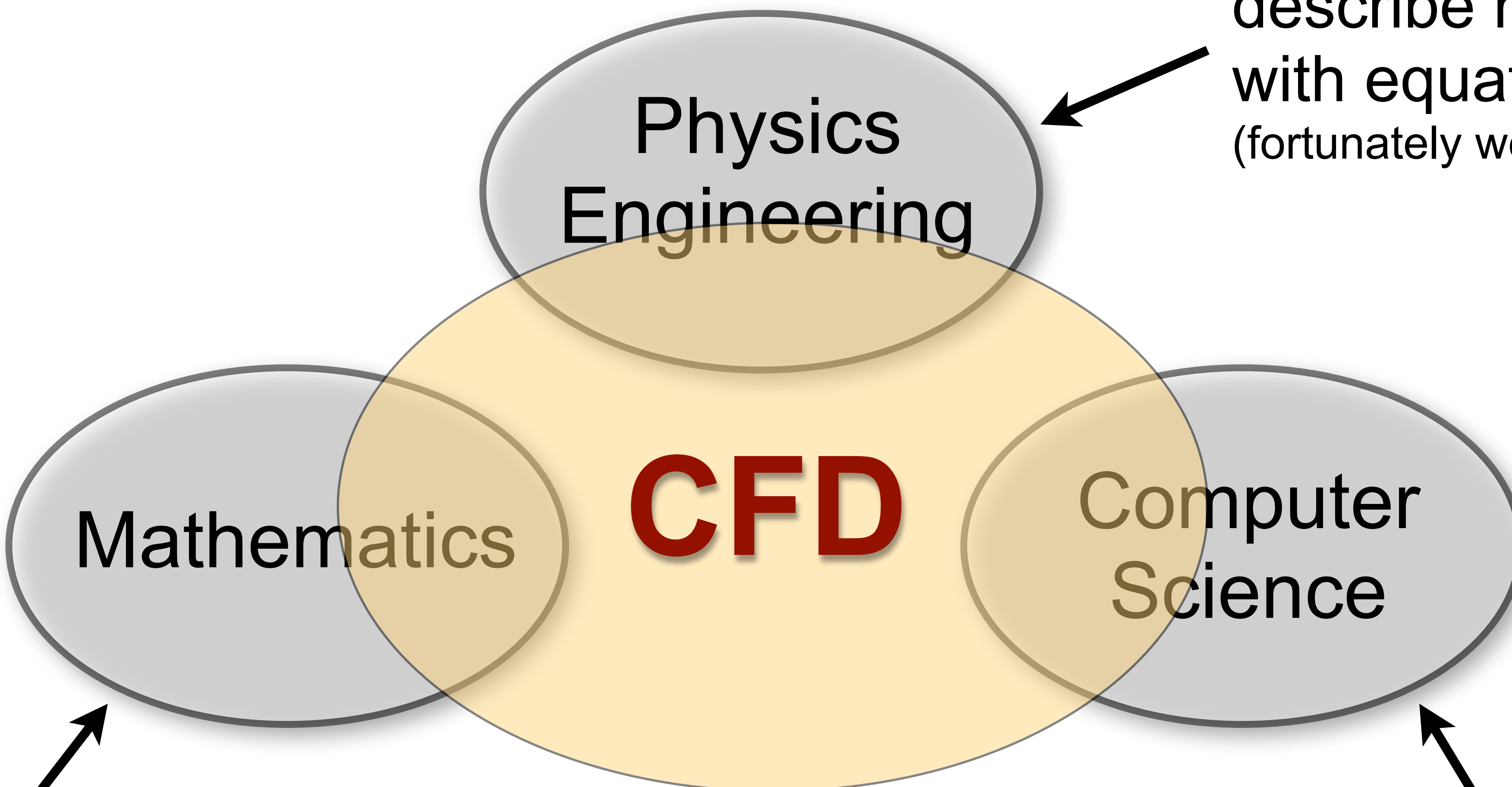
- Complex geometries using Immersed Boundary methods
  - unstructured Cartesian solvers
  - adaptive mesh refinement
  - parallel dynamic load-balancing

H. Sodano

What is CFD?

- Definition: ***Computational Fluid Dynamics (CFD)*** is the numerical solution of the equations describing fluid flows

describe reality with equations
(fortunately we know the Eqs.)

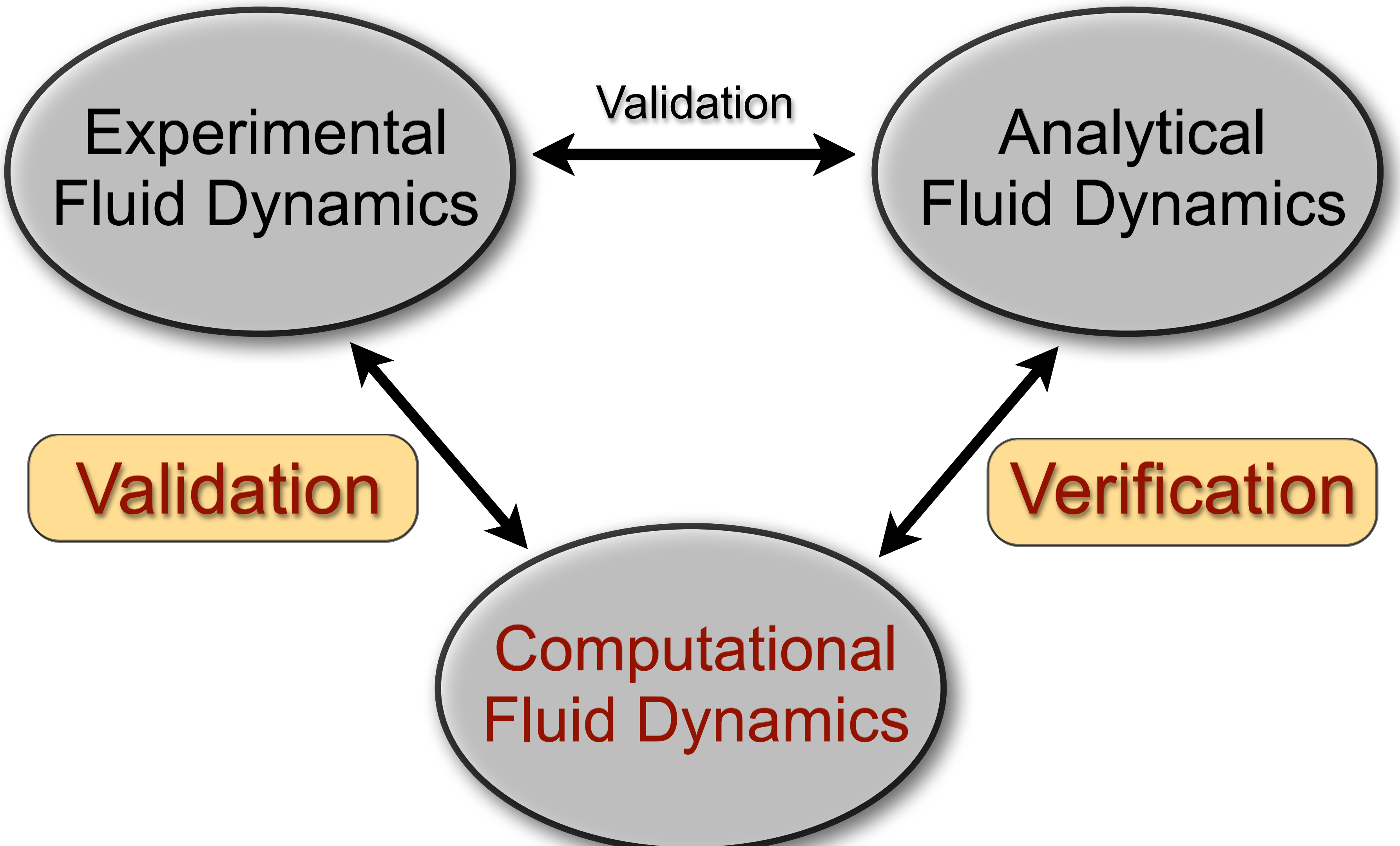


- properties of Eqs.
- numerical methods for solving Eqs.

- Code
- Hardware

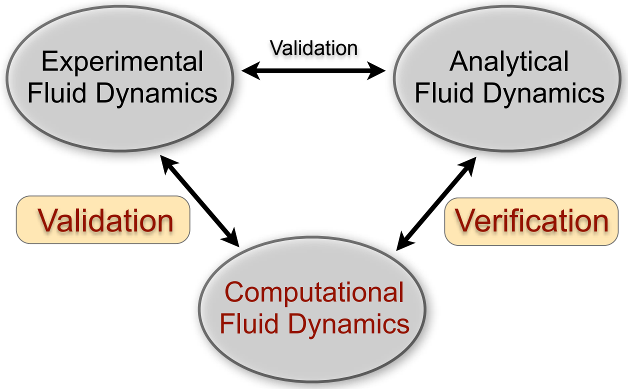
What is CFD?

- Definition: Computational Fluid Dynamics (CFD) is the numerical solution of the equations ***describing fluid flows***



  - Validation: Am I solving the correct equations?
  - Verification: Am I solving the equations correctly?

Why CFD?



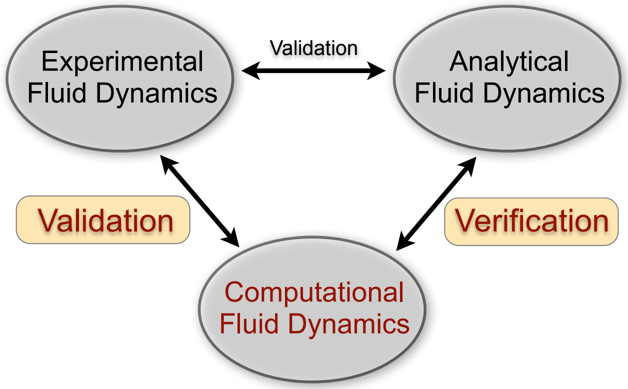
- Full 3D flow data ⇒ understand the flow
  (experiments usually yield just point or plane data)
- Complex flows ⇒ reality
  (analytical fluid dynamics limited to simple flows)

Integrated Simulation of Turbulent Flow in a Jet Engine

G.Medic, G.Kalitzin, D.You, M.Herrmann, E.van der Weide,
F.Ham, G.Iaccarino, H.Pitsch, J.J.Alonso and P.Moin

Center for Integrated Turbulence Simulations
Stanford University

Why CFD?



- Full 3D flow data ⇒ understand the flow
  (experiments usually yield just point or plane data)
- Complex flows ⇒ reality
  (analytical fluid dynamics limited to simple flows)
- Numerical experiments in design process
  (no need to build model ⇒ faster turnaround times than experiments)
- What if studies, even unphysical ones!
  (understand physics)

BUT:

- Can we trust CFD results?
  (Will you fly in a plane designed & tested purely on a computer?)
- Verification & Validation is crucial
- New disciplines: Uncertainty propagation & quantification

How is CFD done today?

- Mainly 2 different approaches:

Industrial

- Commercial Codes:
  - ANSYS Fluent/CFX
  - StarCD
  - etc.
- In-House Legacy Codes

Research

- Code your own, group codes
- CFD-Libraries:
  - OpenFoam
  - Gerris
  - etc.

Good Resource: www.cfd-online.com

The 4 Steps of Doing CFD

(I) Know the flow problem you are trying to solve

- Geometry, boundary conditions, initial conditions
- Flow regime, dominant physical mechanisms
  ⇒ potential simplifications

(II) Pre-processing

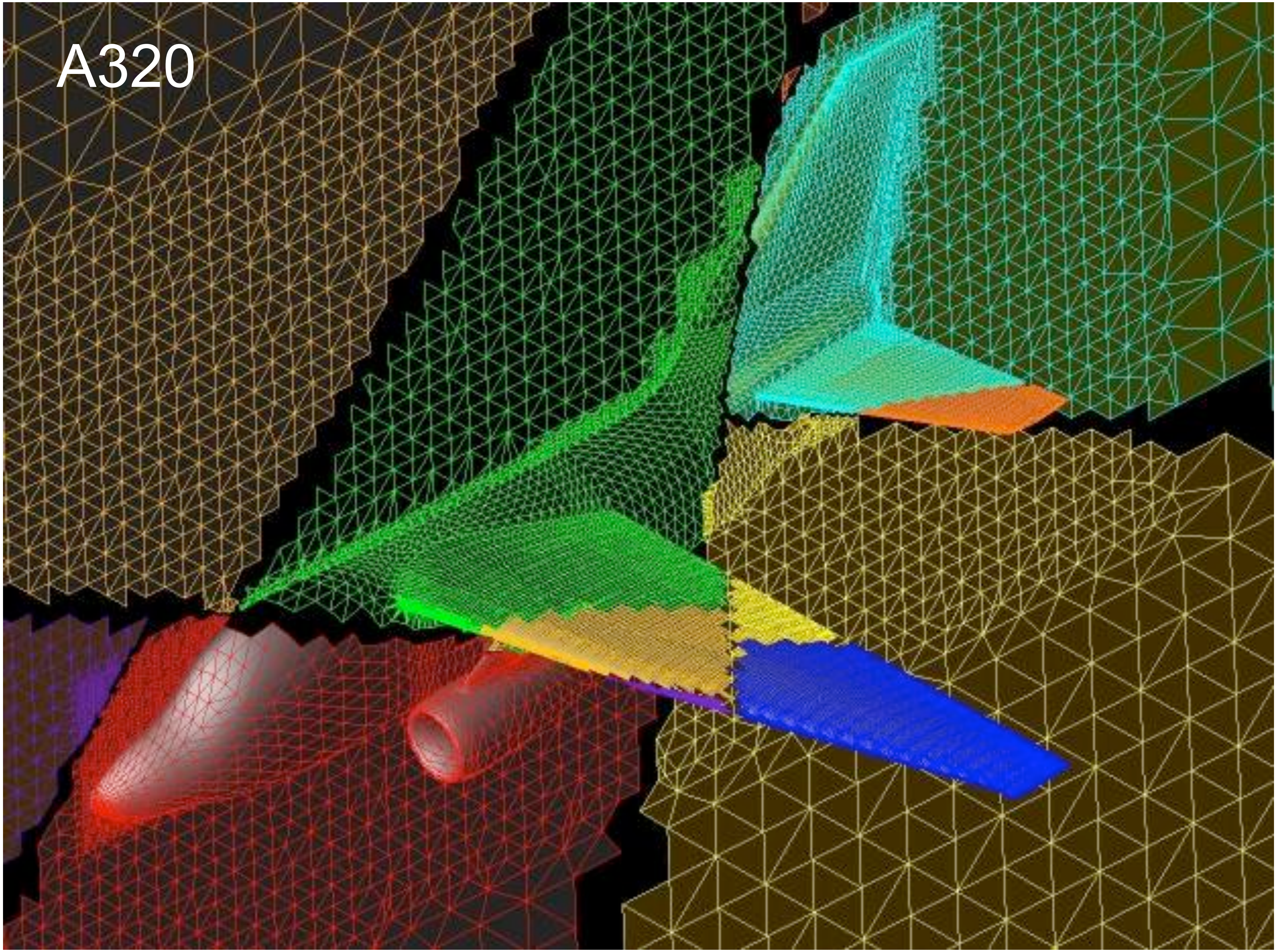
a) generate geometry (CAD or ANSYS DesignModeler)

b) generate a mesh (ANSYS Mesh)

  i. structured mesh (simple geometries)

  ii. unstructured mesh (complex geometries)

c) define initial & boundary conditions (ANSYS Fluent)



J. Alonso, Stanford

The 4 Steps of Doing CFD

(III) Processing

- numerical solution of equations (ANSYS Fluent / own code)
- Caveats:
  - must have equations
  - must have appropriate numerical methods to solve equations
  - must have sufficient computational power and storage

(IV) Post-processing

- Visualize flow
  (ANSYS CFD-Post/Gnuplot/Matlab;
  better: Paraview/Ensight/Tecplot)
- Data base & data mine results (can be Terabytes of data)
- Analyze & understand the flow