SU2 Overview and Installation

POINTWISE® AND SU2 JOINT WORKSHOP

SEPT 29TH-30TH, 2014

Thomas D. Economon, Francisco Palacios
Aeronautics & Astronautics Department (Stanford University)

SU2
The Open-Source CFD Code

aerospacedesignlab

I) WHAT IS SU2?
II) PROGRESS TO DATE
III) A NEW ERA
IV) GETTING STARTED

I) WHAT IS SU2?
II) PROGRESS TO DATE
III) A NEW ERA
IV) GETTING STARTED

What is SU2?

The SU2 suite is an **open-source** collection of C++ based software for multi-physics simulation and design (i.e., CFD!).

SU2 is under active development at Stanford University in the Aerospace Design Lab (ADL) of the Department of Aeronautics and Astronautics and **now in many places around the world**.

http://su2.stanford.edu/

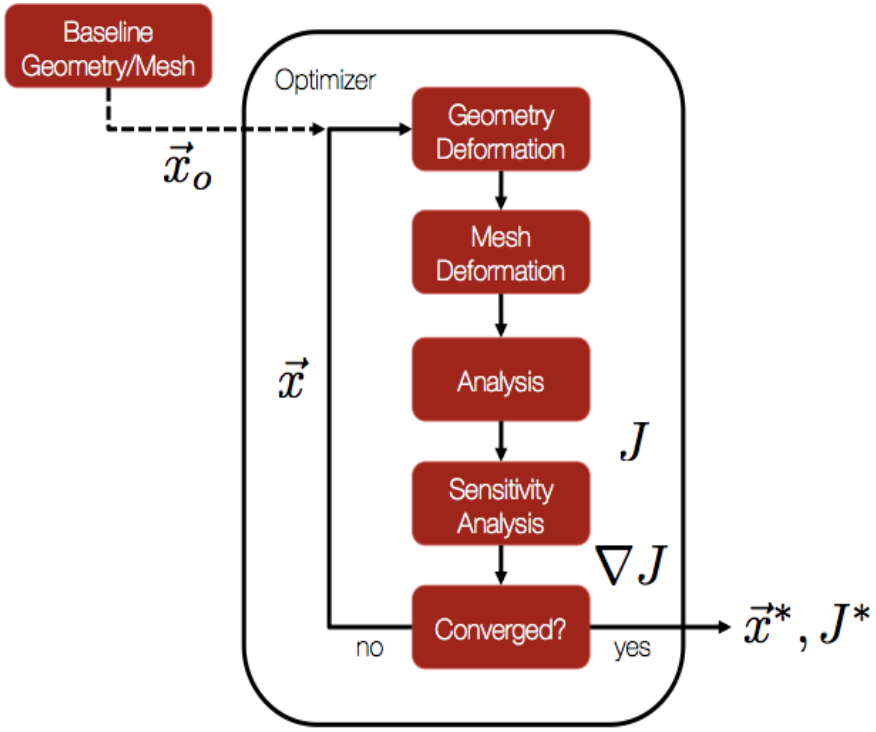
2012 SU² team, "Stanford University Unstructured (SU2): An open-source integrated computational environment for multi-physics simulation and design", AIAA Paper 2013-0287.
2013 SU² team, "Stanford University Unstructured (SU2): Open-source analysis and design technology for turbulent flows", AIAA Paper 2014-0243.

What is SU2?

SU2
The Open-Source CFD Code

$\vec{x}$ vector of design variables

J objective/constraint function(s)



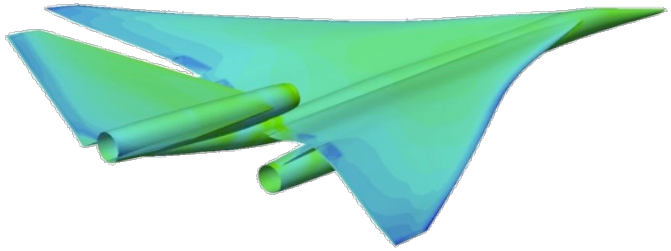
CFD gives us J (lift, drag, etc.),
but how do we get ∇J efficiently?

C++ tools are embedded in a **Python framework** to automate complex tasks, such as optimal shape design.

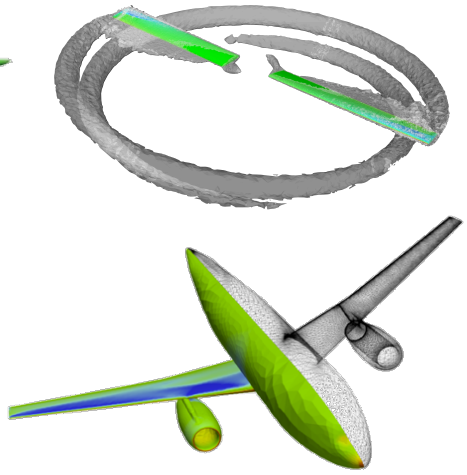
- Adjoint gradient-based optimization:
  - **Continuous** adjoint methodology to achieve cost-independence wrt number of design parameters
  - Analytic expression for the **gradient as a surface integral**.
  - No memory or time overhead compared with the direct solver.

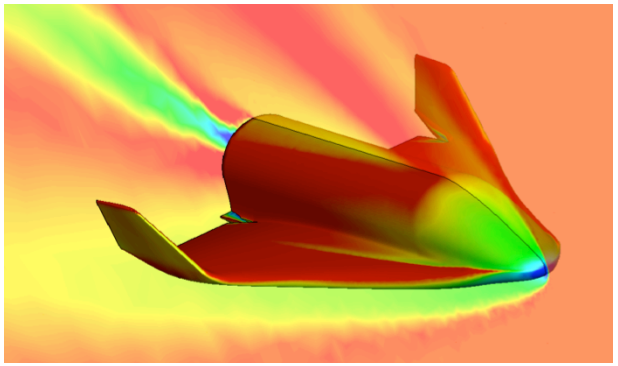
Our Objective

- **Computational analysis tools have revolutionized the way we design aerospace systems**.
- Unfortunately, **most established codes are closed, proprietary, unavailable for some potential users, or prohibitively expensive**.
- The SU2 team looks forward to making **CFD analysis and design freely available as open-source software, involving everyone in its creation, enhancement, and rapid development.**



NASA N+2 Supersonics Program (2013)





SOAR - S3 Swiss Space Systems (2014)

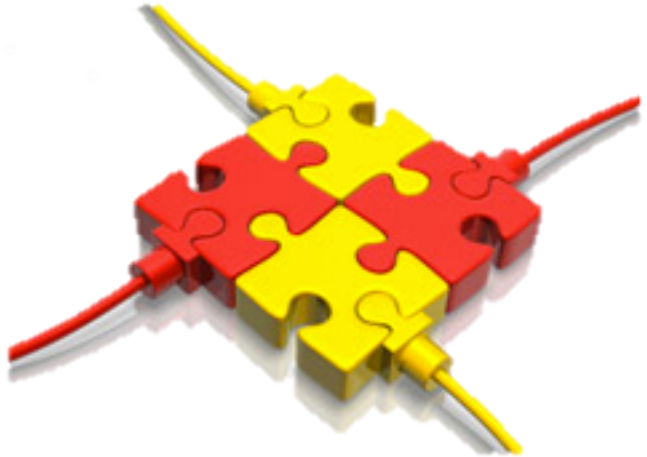
Why SU2?
Software integration in aerodynamic shape design

SU2
The Open-Source CFD Code

Integrating existing software packages into coupled multi-physics analysis and design optimization solvers is a challenge.

The variety of approaches chosen for the independent components of the overall problem make it difficult to:

- Expand the range of applicability to situations not originally envisioned.
- Reduce the overall burden of creating integrated applications.
- Exploit fully-coupled approaches.



Why SU2?
Lasting infrastructure for future efforts

SU2
The Open-Source CFD Code

1. **An open-source model:** basic formulation with a reasonable set of initial capabilities and room for contributions from the community!
2. **Portability:** SU2 has been developed using ANSI C++ and only relies on widely-available, well-supported, open-source software.
3. **Reusability and encapsulation:** SU2 is built so that the main concepts (geometry, sol. algorithms, num. algorithms, etc.) are abstracted to a very high-level. This abstraction promotes reusability of the code and enables modifications without adversely affecting other portions of the suite.
4. **Flexibility:** required to re-purpose existing software for new and different uses. Enabling a common interface for all the necessary components.
5. **Performance:** we have attempted to develop numerical solution algorithms that result in high-performance convergence of the solver in SU2. Striving to strike balance between performance and flexibility.
6. **Gradient availability:** for many applications it is important to obtain sensitivities of the responses computed by SU2 to variations in design parameters.

Why SU2?
Open-Source Philosophy

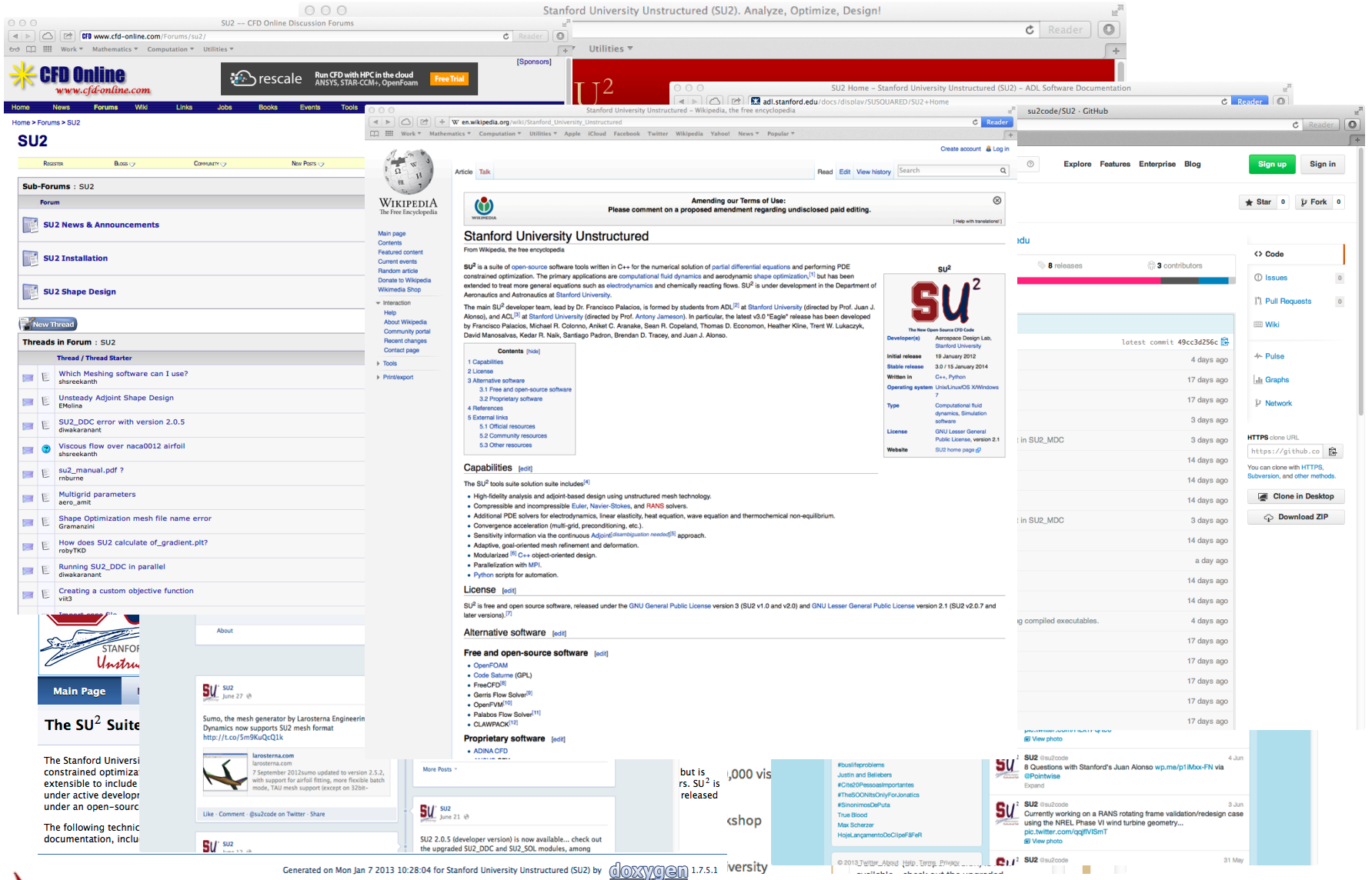
SU2
The Open-Source CFD Code

- We **fully endorse the spirit of the open-source movement**
- We happily support the development efforts in the community
- Avoid “starting from scratch” and focus on research
- We believe that an open-source code supported by a large group of developers working in concert has tremendous potential…
  - Technical excellence: draw from expertise all around the world
  - Open platform encourages collaboration
  - Increase the pace of innovation in computational science

I) WHAT IS SU2?
II) PROGRESS TO DATE
III) A NEW ERA
IV) GETTING STARTED

SU2 Infrastructure

SU2
The Open-Source CFD Code



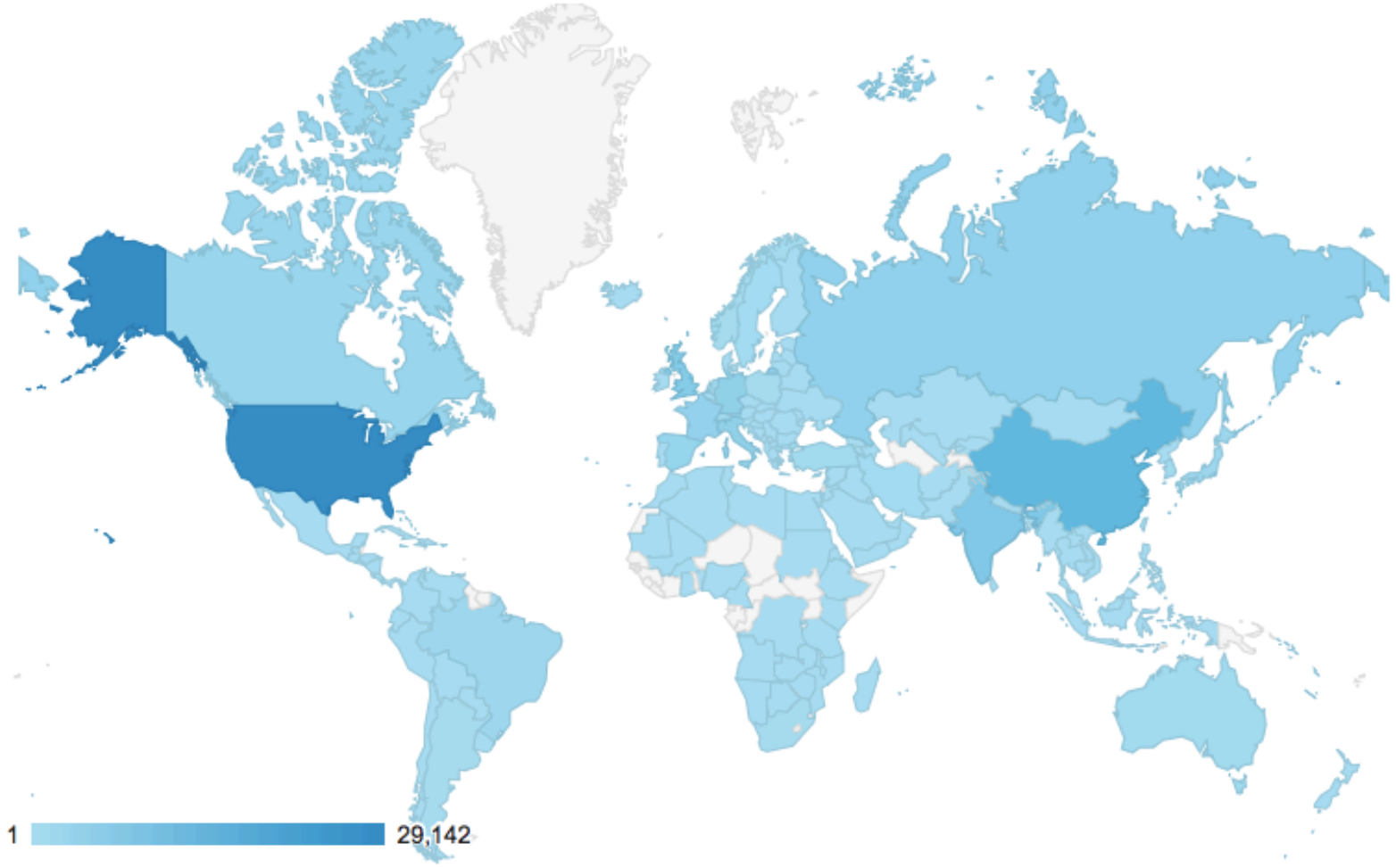
aerospacedesignlab

Stanford University

Global Reach

SU2
The Open-Source CFD Code

Since release in Jan. 2012: 102,000+ web visits across 160 countries



Global Reach

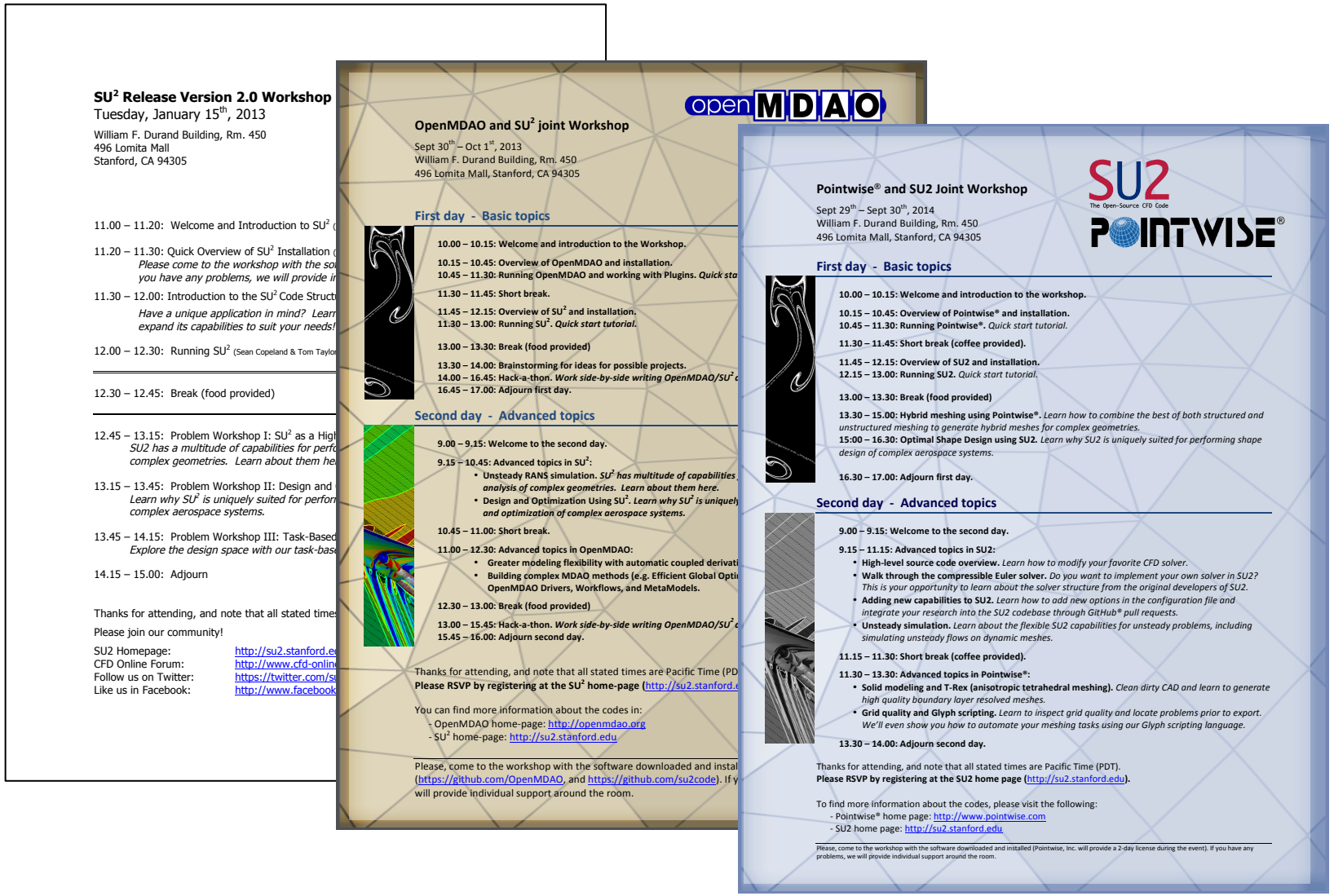
SU2
The Open-Source CFD Code

1.	United States	29,142 (28.40%)
2.	China	12,538 (12.22%)
3.	India	6,970 (6.79%)
4.	United Kingdom	6,499 (6.33%)
5.	Italy	4,343 (4.23%)
6.	Germany	4,323 (4.21%)
7.	Russia	4,020 (3.92%)
8.	France	3,503 (3.41%)
9.	Spain	3,237 (3.15%)
10.	Canada	2,788 (2.72%)

10,000+ downloads, 6,200+ email addresses on the user list

3rd Major Workshop

SU2
The Open-Source CFD Code



SU² Release Version 2.0 Workshop
Tuesday, January 15th, 2013

William F. Durand Building, Rm. 450
496 Lomita Mall
Stanford, CA 94305

11.00 – 11.20: Welcome and Introduction to SU²

11.20 – 11.30: Quick Overview of SU² Installation
Please come to the workshop with the so
you have any problems, we will provide i

11.30 – 12.00: Introduction to the SU² Code Struct
Have a unique application in mind? Learn
expand its capabilities to suit your needs!

12.00 – 12.30: Running SU² (Sean Copeland & Tom Taylo

12.30 – 12.45: Break (food provided)

12.45 – 13.15: Problem Workshop I: SU² as a High
SU2 has a multitude of capabilities for perfo
complex geometries. Learn about them he

13.15 – 13.45: Problem Workshop II: Design and
Learn why SU² is uniquely suited for perfor
complex aerospace systems.

13.45 – 14.15: Problem Workshop III: Task-Based
Explore the design space with our task-base

14.15 – 15.00: Adjourn

Thanks for attending, and note that all stated time

Please join our community!

SU2 Homepage: http://su2.stanford.e
CFD Online Forum: http://www.cfd-onlin
Follow us on Twitter: https://twitter.com/s
Like us in Facebook: http://www.facebook

openMDAO

OpenMDAO and SU² joint Workshop

Sept 30th – Oct 1st, 2013
William F. Durand Building, Rm. 450
496 Lomita Mall, Stanford, CA 94305

First day - Basic topics

10.00 – 10.15: Welcome and introduction to the Workshop.
10.15 – 10.45: Overview of OpenMDAO and installation.
10.45 – 11.30: Running OpenMDAO and working with Plugins. ***Quick sta***
11.30 – 11.45: Short break.
11.45 – 12.15: Overview of SU² and installation.
11.30 – 13.00: Running SU². *Quick start tutorial.*
13.00 – 13.30: Break (food provided)
13.30 – 14.00: Brainstorming for ideas for possible projects.
14.00 – 16.45: Hack-a-thon. *Work side-by-side writing OpenMDAO/SU² c*
16.45 – 17.00: Adjourn first day.

Second day - Advanced topics

9.00 – 9.15: Welcome to the second day.
9.15 – 10.45: Advanced topics in SU²:
- **Unsteady RANS simulation. *SU² has multitude of capabilities analysis of complex geometries. Learn about them here.***
- **Design and Optimization Using SU². *Learn why SU² is uniquely and optimization of complex aerospace systems.***

10.45 – 11.00: Short break.
11.00 – 12.30: Advanced topics in OpenMDAO:
- **Greater modeling flexibility with automatic coupled derivati**
- **Building complex MDAO methods (e.g. Efficient Global Opti OpenMDAO Drivers, Workflows, and MetaModels.**

12.30 – 13.00: Break (food provided)
13.00 – 15.45: Hack-a-thon. *Work side-by-side writing OpenMDAO/SU² c*
15.45 – 16.00: Adjourn second day.

Thanks for attending, and note that all stated times are Pacific Time (PD
Please RSVP by registering at the SU² home-page (http://su2.stanford.e

You can find more information about the codes in:
- OpenMDAO home-page: http://openmdao.org
- SU² home-page: http://su2.stanford.edu

Please, come to the workshop with the software downloaded and instal
(https://github.com/OpenMDAO, and https://github.com/su2code). If y
will provide individual support around the room.

Pointwise® and SU2 Joint Workshop

SU2 The Open-Source CFD Code
POINTWISE®

Sept 29th – Sept 30th, 2014
William F. Durand Building, Rm. 450
496 Lomita Mall, Stanford, CA 94305

First day - Basic topics

10.00 – 10.15: Welcome and introduction to the workshop.
10.15 – 10.45: Overview of Pointwise® and installation.
10.45 – 11.30: Running Pointwise®. *Quick start tutorial.*
11.30 – 11.45: Short break (coffee provided).
11.45 – 12.15: Overview of SU2 and installation.
12.15 – 13.00: Running SU2. *Quick start tutorial.*
13.00 – 13.30: Break (food provided)
13.30 – 15.00: Hybrid meshing using Pointwise®. *Learn how to combine the best of both structured and unstructured meshing to generate hybrid meshes for complex geometries.*
15:00 – 16.30: Optimal Shape Design using SU2. *Learn why SU2 is uniquely suited for performing shape design of complex aerospace systems.*
16.30 – 17.00: Adjourn first day.

Second day - Advanced topics

9.00 – 9.15: Welcome to the second day.
9.15 – 11.15: Advanced topics in SU2:
- **High-level source code overview.** *Learn how to modify your favorite CFD solver.*
- **Walk through the compressible Euler solver.** *Do you want to implement your own solver in SU2? This is your opportunity to learn about the solver structure from the original developers of SU2.*
- **Adding new capabilities to SU2.** *Learn how to add new options in the configuration file and integrate your research into the SU2 codebase through GitHub® pull requests.*
- **Unsteady simulation.** *Learn about the flexible SU2 capabilities for unsteady problems, including simulating unsteady flows on dynamic meshes.*

11.15 – 11.30: Short break (coffee provided).
11.30 – 13.30: Advanced topics in Pointwise®:
- **Solid modeling and T-Rex (anisotropic tetrahedral meshing).** *Clean dirty CAD and learn to generate high quality boundary layer resolved meshes.*
- **Grid quality and Glyph scripting.** *Learn to inspect grid quality and locate problems prior to export. We'll even show you how to automate your meshing tasks using our Glyph scripting language.*

13.30 – 14.00: Adjourn second day.

Thanks for attending, and note that all stated times are Pacific Time (PDT).
Please RSVP by registering at the SU2 home page (http://su2.stanford.edu**).**

To find more information about the codes, please visit the following:
- Pointwise® home page: http://www.pointwise.com
- SU2 home page: http://su2.stanford.edu

Please, come to the workshop with the software downloaded and installed (Pointwise, Inc. will provide a 2-day license during the event). If you have any problems, we will provide individual support around the room.

aerospacedesignlab

Stanford University

Verification & Validation

SU2
The Open-Source CFD Code

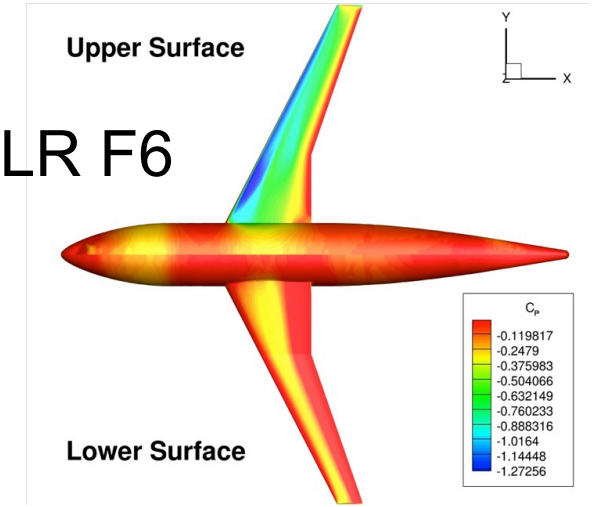
Verification and Validation is critical. It should address **the consistency of the numerical methods**, a solid **accuracy** assessment for different critical application cases, and **sensitivity studies** with respect to parameters.

1. **Unit/Test problems:** zero pressure gradient flat plate boundary layer, bump in a channel, and unsteady flow around a square cylinder.
2. **Subsonic airfoil geometries:** NACA 0012 with attached flow, NACA 4412 with a recirculation bubble, and the McDonnell-Douglas 30P30N three-element high-lift configuration.
3. **Subsonic wing and rotorcraft configurations:** delta wing at a high angle of attack, and the Caradonna & Tung rotor.
4. **Transonic airfoil geometries:** NACA 0012 with attached flow after a shock, RAE 2822 with flow separation.
5. **Transonic wing and full aircraft configurations:** ONERA M6, and the DLR F6 model.

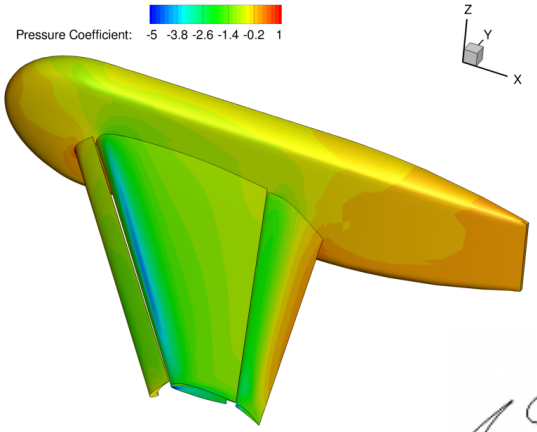
2013 SU² team, "Stanford University Unstructured (SU2): Open-source analysis and design technology for turbulent flows", AIAA Paper 2014-0243.

V & V Exercises (RANS)

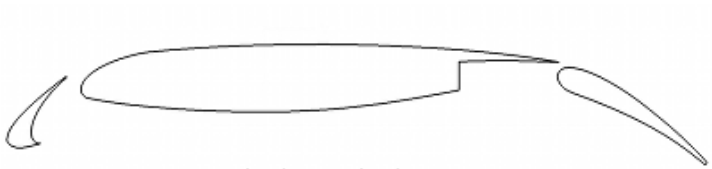
SU2
The Open-Source CFD Code



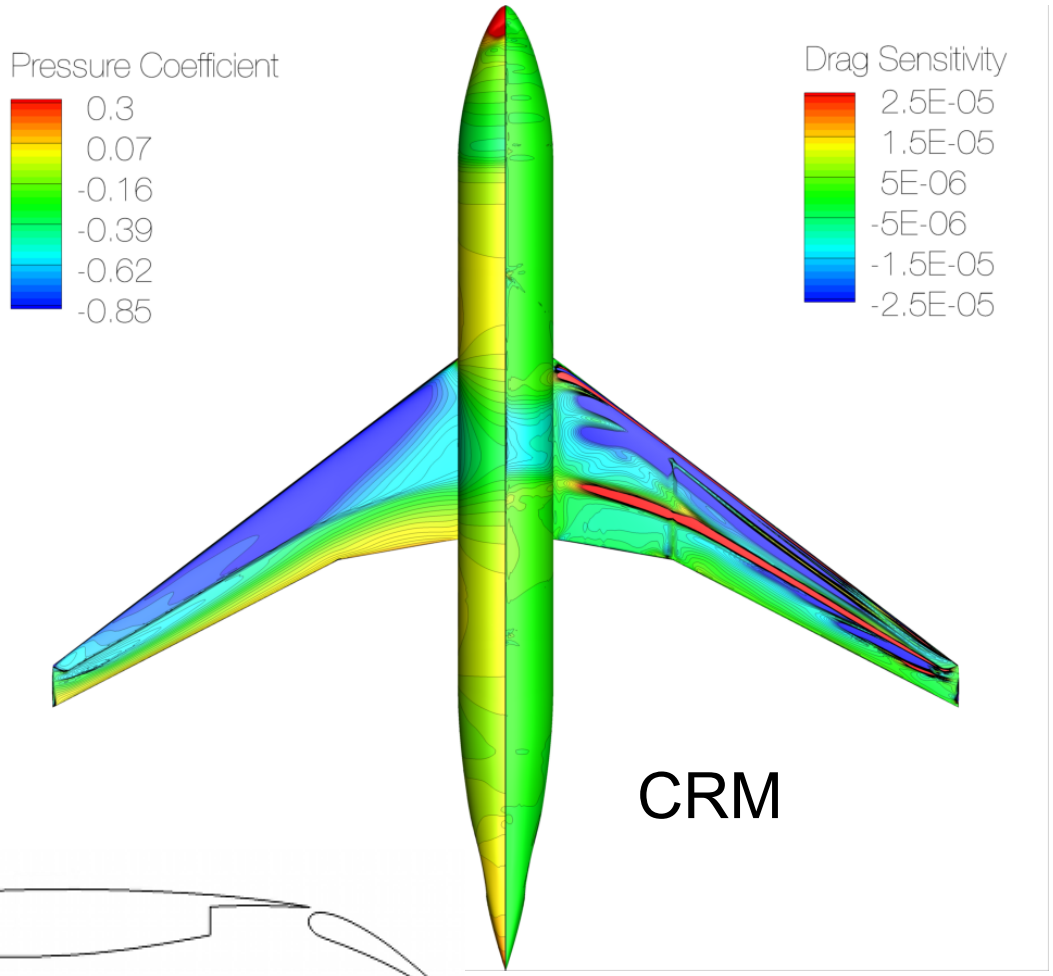
DLR F6



Trap Wing



30P30N



CRM

Ongoing Collaborations

SU2
The Open-Source CFD Code

- Pointwise!
- Intel Parallel Computing Center: SU2_PHI, improving performance and scalability while retaining flexibility. Porting to advanced Xeon and Phi architectures with high performance? Design in a box?
- U. Braunschweig: improving turbulence models
- Univ. of Aachen (Prof. Nico Gauger): automatic differentiation for steady and unsteady flow control
- Delft University: real gas models
- VKI: mutation++ library for non-eq. flows and combustion
- NASA: Integration with OpenMDAO, advanced propulsion architectures
- Others: supersonics, transition models, UAVs, learning CFD, etc.

I) WHAT IS SU2?
II) PROGRESS TO DATE
III) A NEW ERA
IV) GETTING STARTED

A New Era of Development

SU2
The Open-Source CFD Code

Welcome to the team.

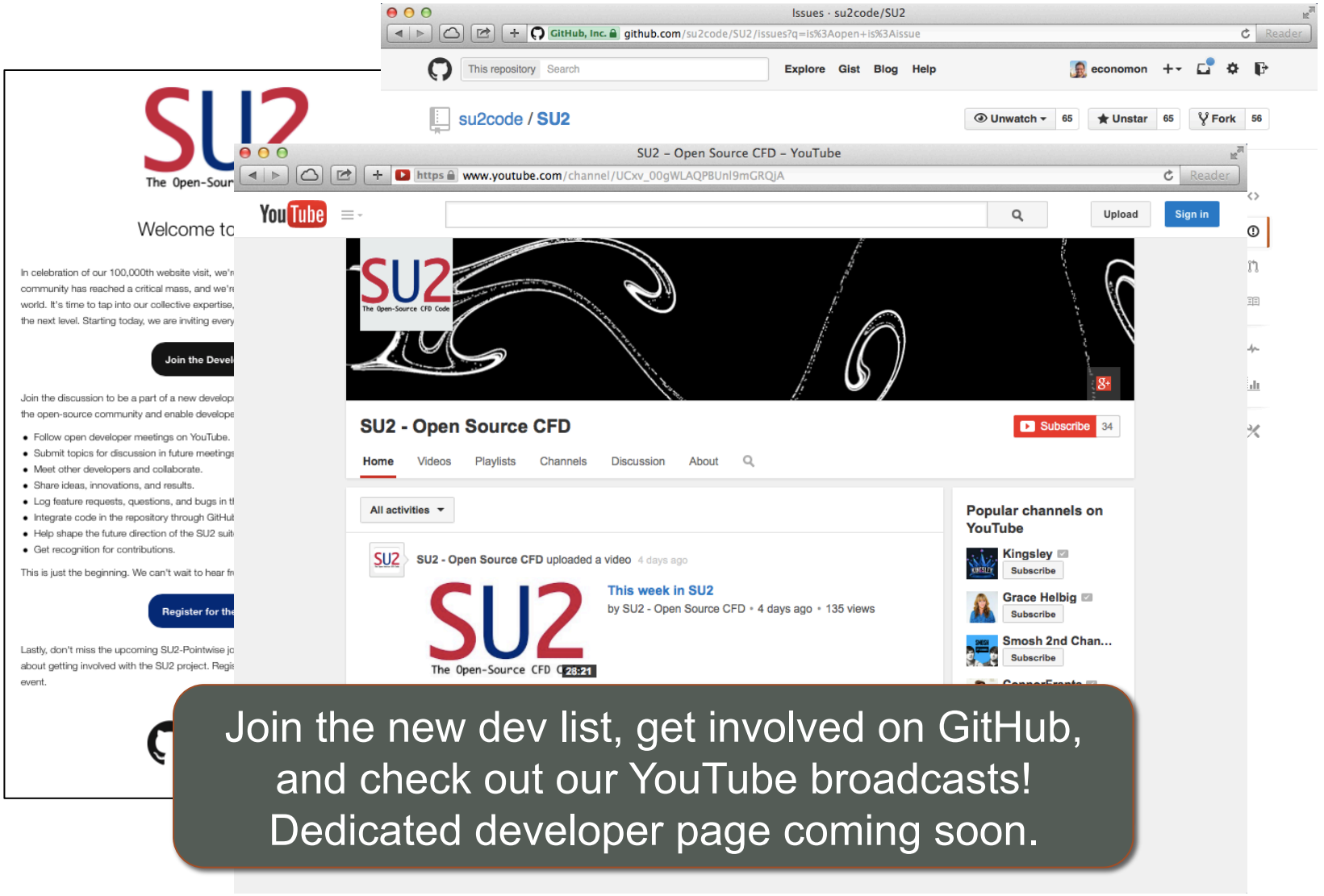
In celebration of our 100,000th website visit, we're thrilled to announce a new initiative. The SU2 community has reached a critical mass, and we're now seeing contributions from all over the world. It's time to tap into our collective expertise, creativity, and coding abilities to take SU2 to the next level. Starting today, we are inviting everyone to join the official developer team.

Join the Developer Team

Join the discussion to be a part of a new development campaign for SU2. We want to engage the open-source community and enable developers to do the following:

New Initiatives

SU2
The Open-Source CFD Code



Join the new dev list, get involved on GitHub, and check out our YouTube broadcasts! Dedicated developer page coming soon.

Flexible Class Structure

SU2
The Open-Source CFD Code

1a) Read Input
Class: CConfig
- Read the config file.

1b) Read Mesh
Class: CGeometry
- Read the mesh file.
- Set up multigrid meshes.

2) Solve Equations
Class: CSolution
- Euler Equations: CEulerSolution
- Plasma Equations: CPlasmaSolution
- Euler Adj. Eq.: CEulerAdjSolution
- Heat Equations: CHeatSolution
- And others…

3) Write Output
Class: COutput
- Print on screen
- Write solution file
- Write restart file
- Write history file

Store Flow Variables
Class: CVariable
- Stores variables at every mesh node.
- Declare & store all flow variables
  - CEulerVariable: Density, energy etc.
  - CNSVariable: CEulerVariable + Viscosity
  - CAdjVariable: Adjoint variables
  - And others…

Discretization
Class: CNumerics

Spatial Discretization
- Convective Flux, Jacobian
  - CNumerics:: Roe/JST/etc.
- Viscous Flux, Jacobian
  - CNumerics:: Avg_Grad/etc.
- Source Terms, Jacobian
  - CNumerics:: PieceWiseConst.

Temporal Discretization
- Implicit Time Integration
- Explicit Euler/ Runge-Kutta
- Dual Time

Solve Linear System
Class: CSparseMatrix
- Linear solvers
  - BiCSTAB
  - GMRES
- Preconditioners
  - Linelet
  - LU-SGS
  - Low Mach number
- Update solution vector

I) WHAT IS SU2?
II) PROGRESS TO DATE
III) A NEW ERA
IV) GETTING STARTED

Getting the Code

SU2
The Open-Source CFD Code

There are a number of ways to get your hands on SU2:

1. Register and download from http://su2.stanford.edu/download.html

2. Download a tagged release from GitHub: https://github.com/su2code/SU2/releases.

3. Fork the SU2 repository to your account in GitHub and start developing in SU2!

Recommended: Forking is recommended for folks that are interested in developing SU2. Once you have a local copy, you can make changes to the code and submit pull requests to get your work into the official repo. Submit your pull requests to the SU2 develop branch for consideration.

Building SU2
Our Open-Source Build Philosophy

SU2
The Open-Source CFD Code

- As an open-source code, care has been taken to **simplify the build process** wherever possible:
  - Use standard C++
  - Leverage standard build tools, i.e., autoconf and automake
  - Avoid or automatically include external dependencies

- To build the vanilla version, all you should need is a C++ compiler.

- However, we maintain the **flexibility** to build with additional features:
  - Fully parallel with MPI, specify your MPI flavor with configure
  - METIS ships with SU2 and is automatically built for parallel runs
  - Tecio for binary Tecplot output also ships as an optional feature
  - CGNS mesh input capability (unstructured, single block)

Building SU2
Simple Build from Source

SU2
The Open-Source CFD Code

1. After getting the code, move into the source directory.

   ```
   $ cd SU2
   ```

2. Run configure to prepare the build.

   ```
   $ ./configure
   ```

3. Make and install the code (installed in /usr/local/bin/ by default)

   ```
   $ make install
   ```

4. Update your PATH variables appropriately before executing.

   ```
   $ export SU2_RUN="/usr/local/bin”
   $ export SU2_HOME="/home/economon/SU2”
   $ export PATH=$SU2_RUN:$PATH
   $ export PYTHONPATH=$SU2_RUN:$PYTHONPATH
   ```

Building SU2
Customizing the Build

SU2
The Open-Source CFD Code

1. After getting the code, move into the source directory.

   ```
   $ cd SU2
   ```

2. Specify various options, compilers, etc., and run configure. The following options are typical for high-performance/parallel operation.

   ```
   $ ./configure --with-MPI=mpicxx CXXFLAGS='-O3' --prefix=/home/economon/SU2 --with-CGNS-lib=/home/economon/cgns-3.1.4/lib --with-CGNS-include=/home/economon/cgns-3.1.4/include
   ```

3. Make and install the code (location specified by the prefix option above). Note that this could be done in two separate steps, as "make" and "make install" consecutively at the command line.

   ```
   $ make -j 8 install
   ```

4. Update your PATH variables appropriately before executing.

   ```
   $ export SU2_RUN="/home/economon/SU2/bin"
   $ export SU2_HOME="/home/economon/SU2"
   $ export PATH=$SU2_RUN:$PATH
   $ export PYTHONPATH=$SU2_RUN:$PYTHONPATH
   ```

Building SU2
Details in the Configure Output

SU2
The Open-Source CFD Code

```
Build Configuration Summary:

    Source code location: /Users/economon/SU2
    Install location:     /Users/economon/SU2
    Version:              3.2.1
    Compiler:             mpicxx
    Preprocessor flags:   -I/opt/X11/include
    Compiler flags:       -DHAVE_MPI -O3
    Linker flags:
    CGNS support:         yes
    MPI support:          yes
    Metis support:        yes
    TecIO support:        yes
    Mutation++ support:   no
    Jsoncpp support:      no

    External includes:    -DHAVE_METIS -I$(top_srcdir)/externals/metis/include -
DHAVE_TECIO -I$(top_srcdir)/externals/tecio/include
    External libs:        $(top_builddir)/externals/metis/libmetis.a $(top_build
dir)/externals/tecio/libtecio.a

    Build SU2_CFD:        yes
    Build SU2_PRT:        yes
    Build SU2_DOT:        yes
    Build SU2_MSH:        yes
    Build SU2_DEF:        yes
    Build SU2_SOL:        yes
    Build SU2_GEO:        yes

Please be sure to add the $SU2_HOME and $SU2_RUN environment variables,
and update your $PATH (and $PYTHONPATH if applicable) with $SU2_RUN.

Based on the input to this configuration, add these lines to your .bashrc file:

export SU2_RUN="/Users/economon/SU2/bin"
export SU2_HOME="/Users/economon/SU2"
export PATH=$PATH:$SU2_RUN
export PYTHONPATH=$PYTHONPATH:$SU2_RUN
```

Building SU2
Tips for Hassle-Free Builds

SU2
The Open-Source CFD Code

Recommended: CGNS V3.1.4 without HDF5

Tip: Use parallel make to accelerate the build, i.e., "$ make -j 24 install" to use 24 processes if you have a workstation with 24 cores.

Recommended: For advanced features in SU2, Python scripts are available. Try out the free Python distribution (V2.X distribution) with built-in packages by Enthought: https://www.enthought.com/products/epd/free/

Building SU2
Tips for Hassle-Free Builds

SU2
The Open-Source CFD Code

Tip: Add an alias for your configure command and set the environment variables in your ~/.bashrc file (steps 2 and 4 on the previous slide).

$ alias config_serial="./configure CXXFLAGS='-O3' --prefix=/Users/economon/SU2 --enable-tecio"

$ alias config_mpi="./configure --with-MPI=mpicxx --prefix=/Users/economon/SU2 --enable-tecio CXXFLAGS='-O3' --with-CGNS-lib=/usr/local/lib --with-CGNS-include=/usr/local/include"

Building SU2
Tips for Hassle-Free Builds

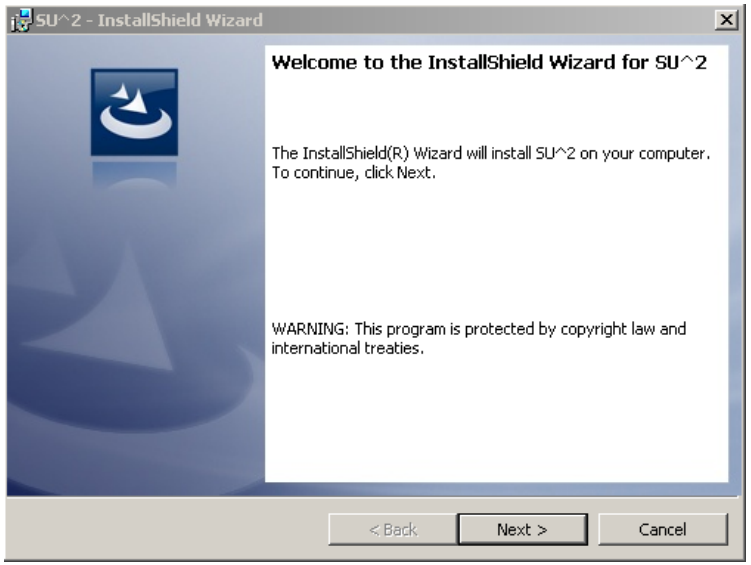
SU2
The Open-Source CFD Code

```
$ export SU2_RUN="/home/economon/SU2/bin”
$ export SU2_HOME="/home/economon/SU2”
$ export PATH=$SU2_RUN:$PATH
$ export PYTHONPATH=$SU2_RUN:$PYTHONPATH
```

Tip: Note the order of the PATH variable. $SU2_RUN appears before $PATH in this case. If $PATH comes before $SU2_RUN, make sure that there are not any previous versions of SU2 installed in other locations in your PATH, otherwise it may use old versions and cause headaches…

Windows Installation

- Download installer and follow wizard instructions
- Installer will place files in the correct locations
- Default install location: C:\SU2
- Command line syntax remains the same



Recommended: Set your WINDOWS PATH after install

Required: For parallel version, you must install and configure Microsoft HPC Pack (or just Microsoft MPI) *before* installing SU2.