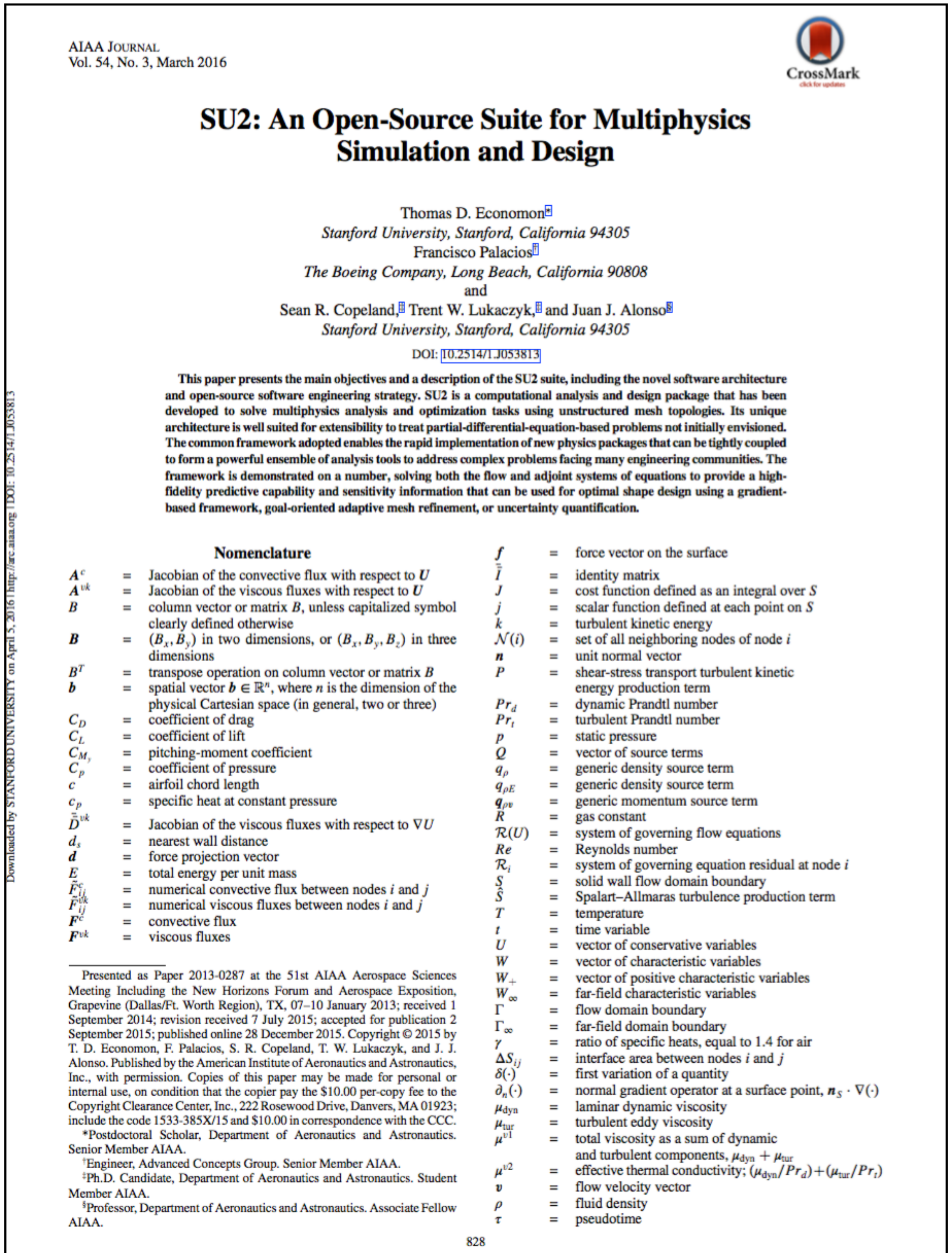
Welcome and Introduction to SU2

SU2 Winter Workshop
February 3rd, 2017

Dr. Thomas D. Economon
Department of Aeronautics & Astronautics
Stanford University

The Open-Source CFD Code

What is SU2?



AIAA Journal
Vol. 54, No. 3, March 2016

CrossMark

SU2: An Open-Source Suite for Multiphysics Simulation and Design

Thomas D. Economon*
Stanford University, Stanford, California 94305
Francisco Palacios†
The Boeing Company, Long Beach, California 90808
and
Sean R. Copeland,‡ Trent W. Lukaczyk,‡ and Juan J. Alonso§
Stanford University, Stanford, California 94305

DOI: 10.2514/1.J053813

This paper presents the main objectives and a description of the SU2 suite, including the novel software architecture and open-source software engineering strategy. SU2 is a computational analysis and design package that has been developed to solve multiphysics analysis and optimization tasks using unstructured mesh topologies. Its unique architecture is well suited for extensibility to treat partial-differential-equation-based problems not initially envisioned. The common framework adopted enables the rapid implementation of new physics packages that can be tightly coupled to form a powerful ensemble of analysis tools to address complex problems facing many engineering communities. The framework is demonstrated on a number, solving both the flow and adjoint systems of equations to provide a high-fidelity predictive capability and sensitivity information that can be used for optimal shape design using a gradient-based framework, goal-oriented adaptive mesh refinement, or uncertainty quantification.

Nomenclature

$\boldsymbol{A}^c$ = Jacobian of the convective flux with respect to $\boldsymbol{U}$
$\boldsymbol{A}^{vk}$ = Jacobian of the viscous fluxes with respect to $\boldsymbol{U}$
B = column vector or matrix B, unless capitalized symbol clearly defined otherwise
$\boldsymbol{B}$ = (B_x, B_y) in two dimensions, or (B_x, B_y, B_z) in three dimensions
B^T = transpose operation on column vector or matrix B
$\boldsymbol{b}$ = spatial vector $\boldsymbol{b} \in \mathbb{R}^n$, where n is the dimension of the physical Cartesian space (in general, two or three)
C_D = coefficient of drag
C_L = coefficient of lift
C_{M_y} = pitching-moment coefficient
C_p = coefficient of pressure
c = airfoil chord length
c_p = specific heat at constant pressure
$\bar{\bar{D}}^{vk}$ = Jacobian of the viscous fluxes with respect to ∇U
d_s = nearest wall distance
$\boldsymbol{d}$ = force projection vector
E = total energy per unit mass
$\tilde{F}^c_{ij}$ = numerical convective flux between nodes i and j
$\tilde{F}^{vk}_{ij}$ = numerical viscous fluxes between nodes i and j
$\boldsymbol{F}^c$ = convective flux
$\boldsymbol{F}^{vk}$ = viscous fluxes
$\boldsymbol{f}$ = force vector on the surface
$\bar{\bar{I}}$ = identity matrix
J = cost function defined as an integral over S
j = scalar function defined at each point on S
k = turbulent kinetic energy
$\mathcal{N}(i)$ = set of all neighboring nodes of node i
$\boldsymbol{n}$ = unit normal vector
P = shear-stress transport turbulent kinetic energy production term
Pr_d = dynamic Prandtl number
Pr_t = turbulent Prandtl number
p = static pressure
Q = vector of source terms
q_ρ = generic density source term
$q_{\rho E}$ = generic density source term
$\boldsymbol{q}_{\rho v}$ = generic momentum source term
R = gas constant
$\mathcal{R}(U)$ = system of governing flow equations
Re = Reynolds number
$\mathcal{R}_i$ = system of governing equation residual at node i
S = solid wall flow domain boundary
$\hat{S}$ = Spalart–Allmaras turbulence production term
T = temperature
t = time variable
U = vector of conservative variables
W = vector of characteristic variables
W_+ = vector of positive characteristic variables
W_∞ = far-field characteristic variables
Γ = flow domain boundary
Γ_∞ = far-field domain boundary
γ = ratio of specific heats, equal to 1.4 for air
ΔS_{ij} = interface area between nodes i and j
$\delta(\cdot)$ = first variation of a quantity
$\partial_n(\cdot)$ = normal gradient operator at a surface point, $\boldsymbol{n}_S \cdot \nabla(\cdot)$
μ_{dyn} = laminar dynamic viscosity
μ_{tur} = turbulent eddy viscosity
μ^{v1} = total viscosity as a sum of dynamic and turbulent components, $\mu_{\mathrm{dyn}} + \mu_{\mathrm{tur}}$
μ^{v2} = effective thermal conductivity; $(\mu_{\mathrm{dyn}}/Pr_d)+(\mu_{\mathrm{tur}}/Pr_t)$
$\boldsymbol{v}$ = flow velocity vector
ρ = fluid density
τ = pseudotime

Presented as Paper 2013-0287 at the 51st AIAA Aerospace Sciences Meeting Including the New Horizons Forum and Aerospace Exposition, Grapevine (Dallas/Ft. Worth Region), TX, 07–10 January 2013; received 1 September 2014; revision received 7 July 2015; accepted for publication 2 September 2015; published online 28 December 2015.

*Postdoctoral Scholar, Department of Aeronautics and Astronautics. Senior Member AIAA.
†Engineer, Advanced Concepts Group. Senior Member AIAA.
‡Ph.D. Candidate, Department of Aeronautics and Astronautics. Student Member AIAA.
§Professor, Department of Aeronautics and Astronautics. Associate Fellow AIAA.

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

SU2 is under active development at Stanford University in the Department of Aeronautics and Astronautics and **now in many places around the world**.

SU2
The Open-Source CFD Code

https://github.com/su2code/SU2

http://su2.stanford.edu

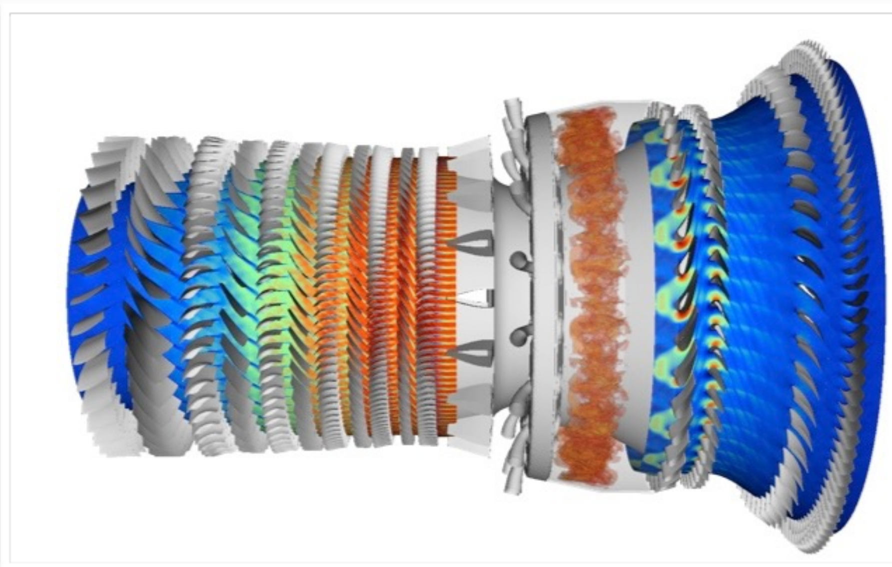
Our Guiding Principles

1. Open-source (LGPL 2.1)!
2. Portability and easy installation.
3. Readability, reusability, and encapsulation (C++).
4. Flexibility and automation (Python).
5. High performance.
6. Gradient availability for design, mesh adaptation, UQ, etc.

We believe that an open-source code supported by a large group of developers working in concert has **tremendous potential**…

- Technical excellence: experts all around the world contribute to produce new research and capabilities not previously envisioned.
- Open, web-based platform encourages global collaboration without geographic limitations.
- Increases the pace of innovation in computational science.

The SU2 Timeline



SUmb solver
developed
@ ADL

June 2008
Francisco Palacios
completes PhD with
Juan Alonso
on committee

Summer/Fall 2009
Francisco spends
3 months at Stanford

2010
Work on
CADES (predecessor
to SU2) begins

Jan 2011
Francisco joins
ADL @ Stanford

Summer/Fall
Preparations for
releasing SU2 as
open source

2003-2008

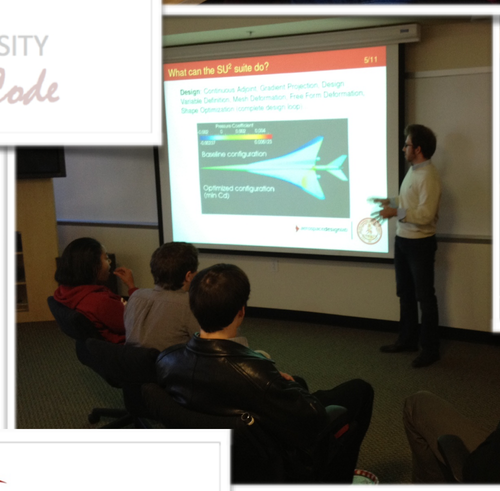
2009

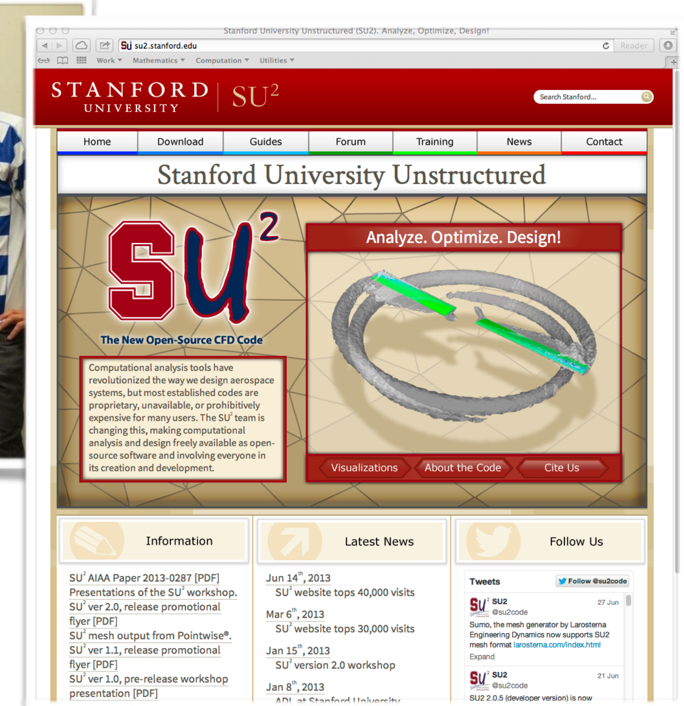
2010

2011

“We must think big... on Jan 20th everybody in the aeronautical community must know that there is a new player in the CFD open-source community.”

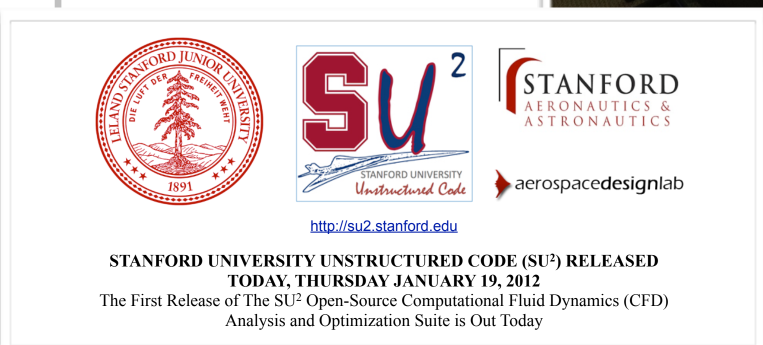
- Dr. Francisco Palacios, January 9 2012

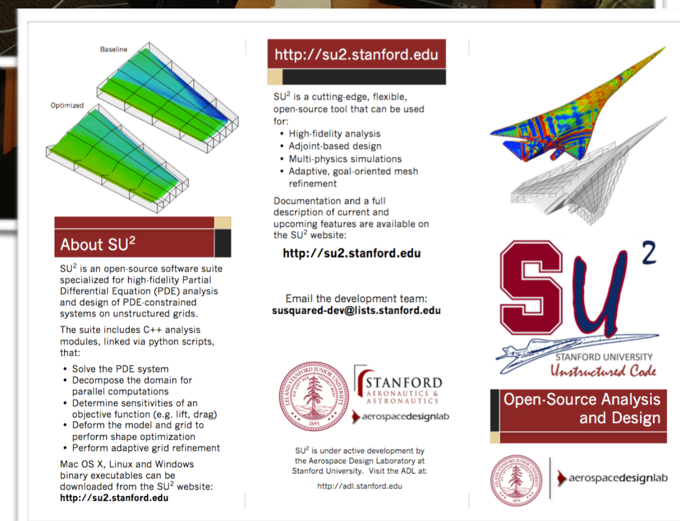




Jan 17
Pre-release
Workshop

Oct 31
su2.stanford.edu update





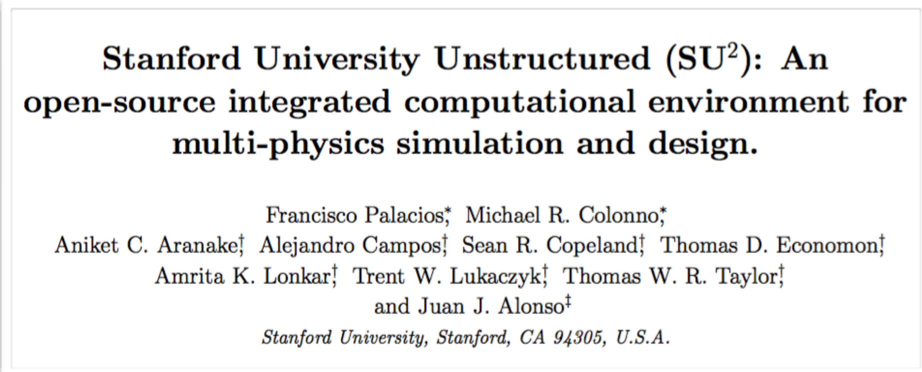
Jan 19
SU2 v1.0
SU2 is born!

Jun 25
SU2 v1.1

Sept 25
First tweet
@su2code

Winter Spring Summer Fall

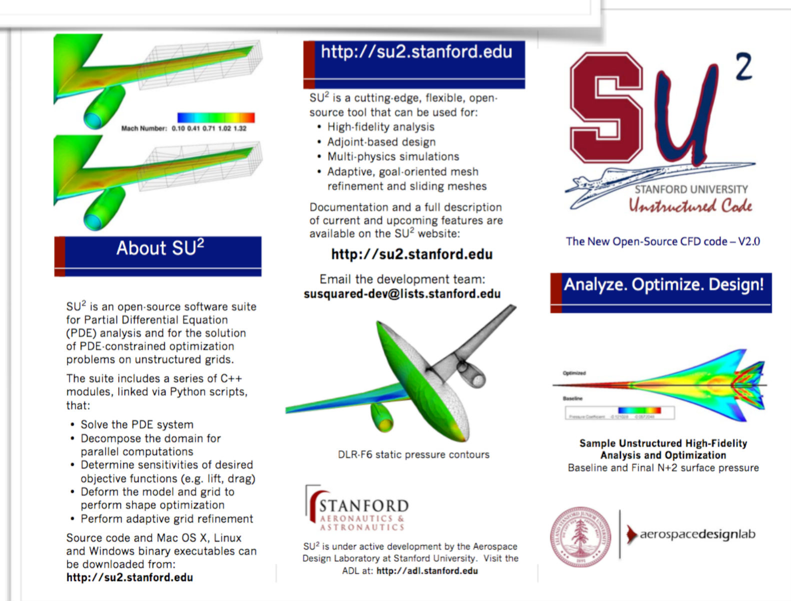
2012


Stanford University Unstructured (SU^2): An open-source integrated computational environment for multi-physics simulation and design.

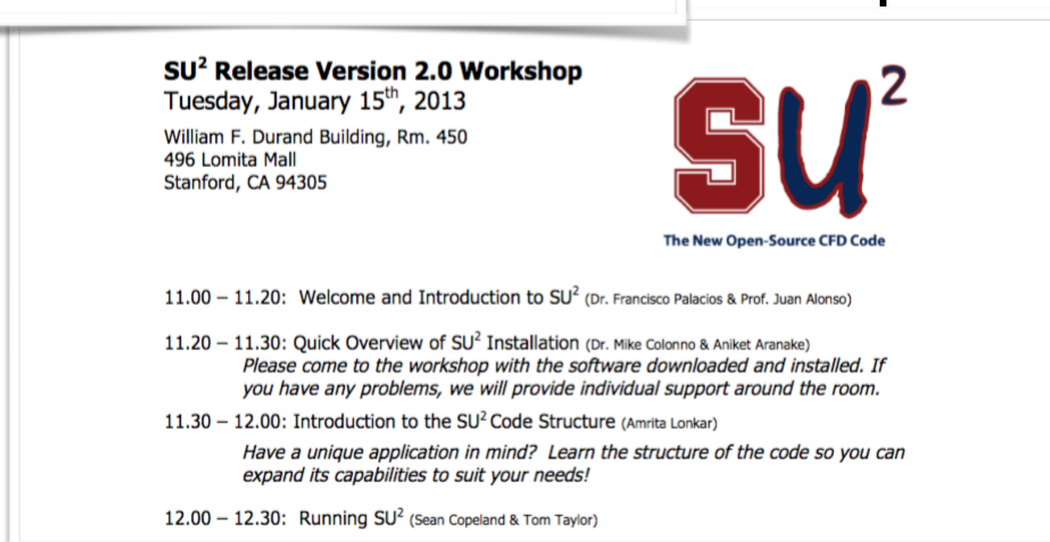
Francisco Palacios, Michael R. Colonno, Aniket C. Aranake, Alejandro Campos, Sean R. Copeland, Thomas D. Economon, Amrita K. Lonkar, Trent W. Lukaczyk, Thomas W. R. Taylor, and Juan J. Alonso

Stanford University, Stanford, CA 94305, U.S.A.

Jan 7
AIAA SciTech Presentation



Jan 8
SU2 v2.0,
CFD Online Forum Open


SU^2 Release Version 2.0 Workshop
Tuesday, January 15th, 2013
William F. Durand Building, Rm. 450
496 Lomita Mall
Stanford, CA 94305

The New Open-Source CFD Code

11.00 – 11.20: Welcome and Introduction to SU^2 (Dr. Francisco Palacios & Prof. Juan Alonso)

11.20 – 11.30: Quick Overview of SU^2 Installation (Dr. Mike Colonno & Aniket Aranake)
Please come to the workshop with the software downloaded and installed. If you have any problems, we will provide individual support around the room.

11.30 – 12.00: Introduction to the SU^2 Code Structure (Amrita Lonkar)
Have a unique application in mind? Learn the structure of the code so you can expand its capabilities to suit your needs!

12.00 – 12.30: Running SU^2 (Sean Copeland & Tom Taylor)

Jan 15
SU2 v2.0
Workshop

May 17 & 25
SU2's first two PhDs

Aug 10
SU2 on GitHub


OpenMDAO and SU^2 joint Workshop
Sept 30th – Oct 1st, 2013
William F. Durand Building, Rm. 450
496 Lomita Mall, Stanford, CA 94305

openMDAO
SU^2 The Open-Source CFD Code

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 start tutorial.*
11.30 – 11.45: Short break.
11.45 – 12.15: Overview of SU^2 and installation.
11.30 – 13.00: Running SU^2. *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/SU^2 applications.*
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 SU^2:
- Unsteady RANS simulation. *SU^2 has multitude of capabilities for performing high-fidelity analysis of complex geometries. Learn about them here.*
- Design and Optimization Using SU^2. *Learn why SU^2 is uniquely suited for performing design 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 derivatives in OpenMDAO.
- Building complex MDAO methods (e.g. Efficient Global Optimization, StackMC) with 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/SU^2 applications.*
15.45 – 16.00: Adjourn second day.

Thanks for attending, and note that all stated times are Pacific Time (PDT).
Please RSVP by registering at the SU^2 home-page (http://su2.stanford.edu).

You can find more information about the codes in:
- OpenMDAO home-page: http://openmdao.org
- SU^2 home-page: http://su2.stanford.edu

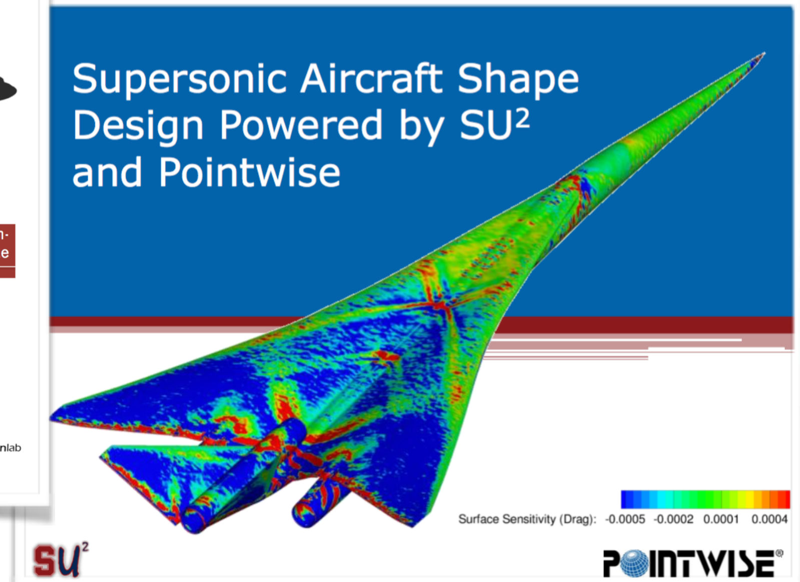
Please, come to the workshop with the software downloaded and installed (https://github.com/OpenMDAO, and https://github.com/su2code). If you have any problems, we will provide individual support around the room.

Sept 30
OpenMDAO / SU2
Joint Workshop

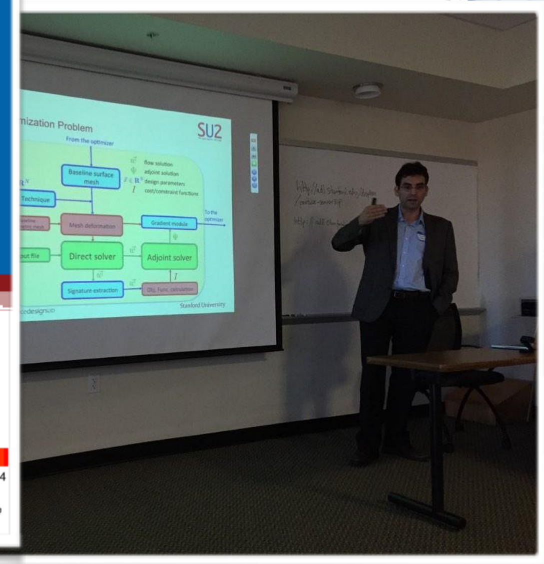
Winter | Spring | Summer | Fall

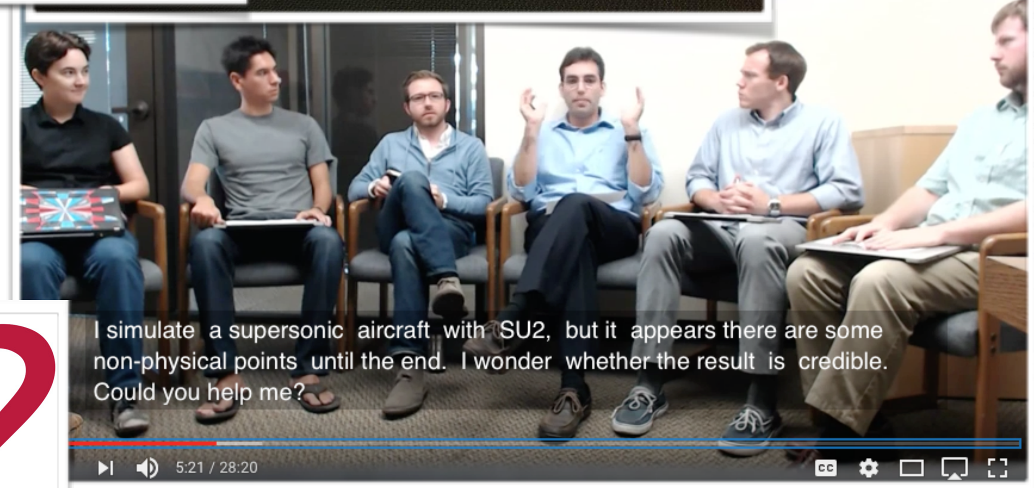
2013

Jan 15
SU2 v3.0
SU2_EDU v1.0



Apr 29
Pointwise-SU2
Webinar





SU2
The Open-Source CFD Code

July 29 - New Logo

July 31
TU Delft & Polimi
Visit Stanford

Sept 29
Pointwise-SU2
Joint Workshop

Sept 24
"This Week in SU2"
Youtube Season 1
Premiere

Apr 14
SU2 v3.1

May 7
Intel Parallel
Comp. Center

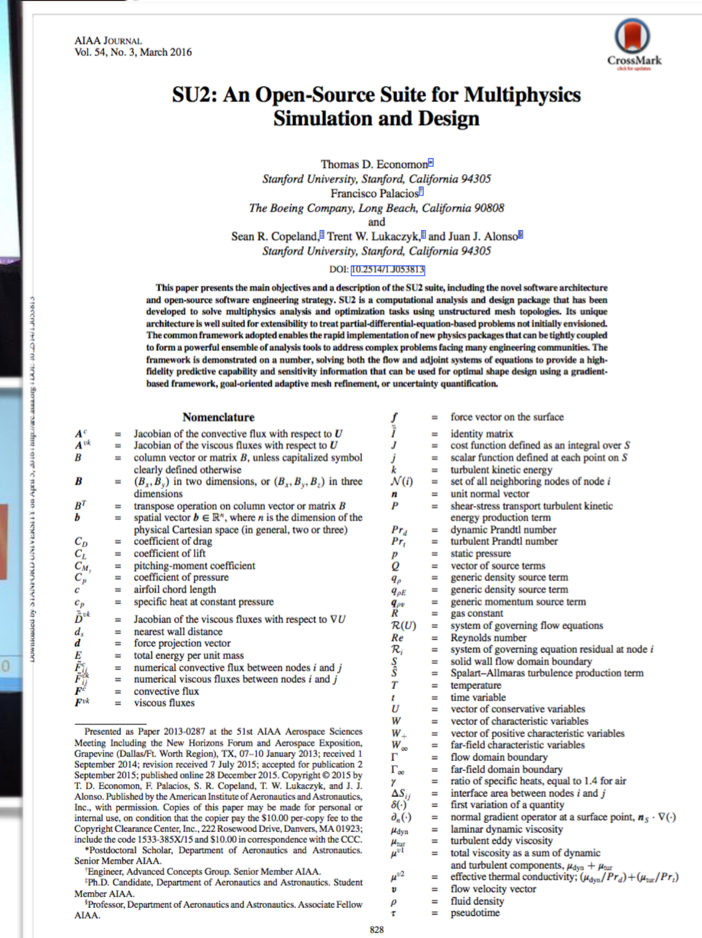
Jun 17
SU2 v3.2

Sept 16
Dev
Email List

Winter | Spring | Summer | Fall

2014

Mar 14
Francisco's farewell
from Stanford



December 29
SU2 in the
AIAA Journal

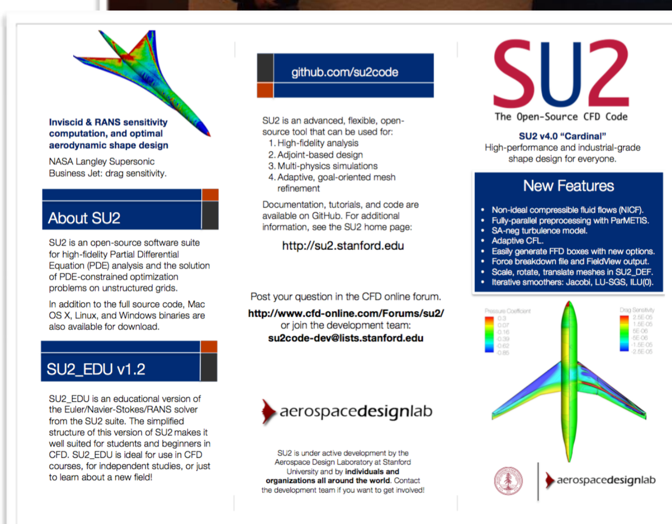
Wednesday, March 18

MS302
PDE-constrained Optimization using the Open-source Code SU2

2:00 PM - 3:40 PM
Room: 151 AB

Most established codes for PDE-constrained optimization are proprietary, unavailable, or prohibitively expensive for many users. The SU2 code is freely available as open-source and features a complete computational analysis framework for multidisciplinary design in applications such as, but not limited to, aerospace technology. This minisymposium will cover up-to-date topics within the SU2 framework related to its continuous and discrete adjoint capabilities, the application to large-scale aerodynamic design, and the utilization of many-core architectures. Each of the topics covered involve the combination of multiple research areas of interest to the CS&E community.

Mar 18
SIAM CSE
Mini



TECHNISCHE UNIVERSITÄT KAISERSLAUTERN

March
TU Kaiserslautern
Visits Stanford

Jun 23
SU2 v4.0@ AIAA
AVIATION
Travis CI

UNIVERSITY OF TWENTE.

Argonne NATIONAL LABORATORY

August 6
ALCF Theta
ESP Selection

Imperial College London

Aug - Sept
Imperial College
Visits Stanford

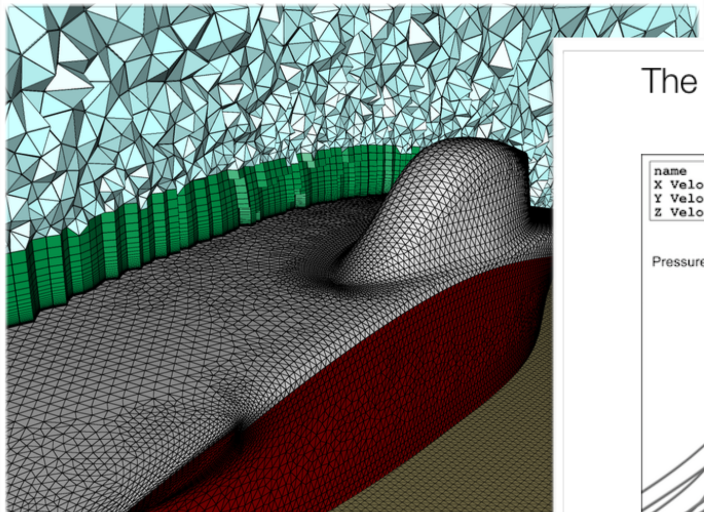
Winter

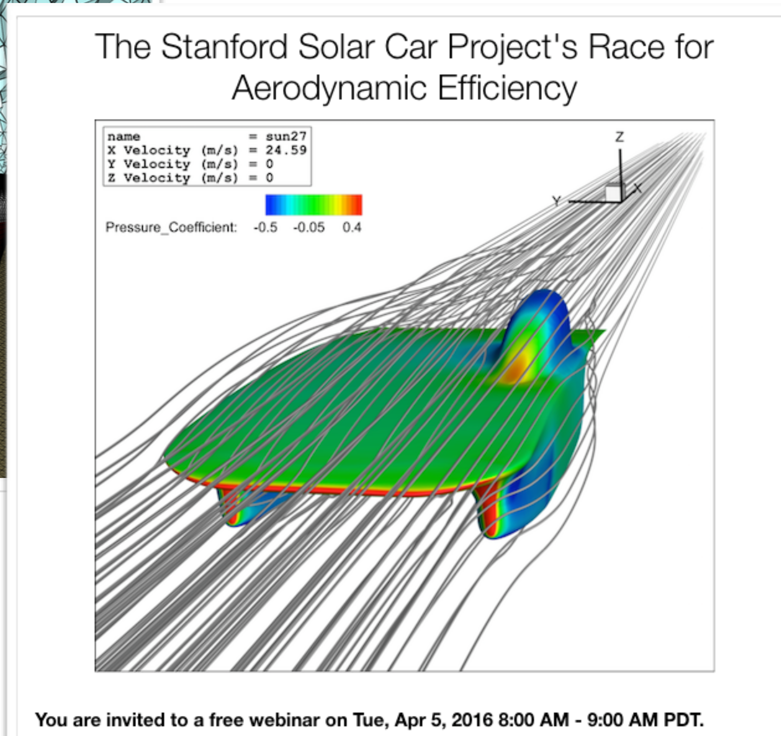
Spring

Summer

Fall

2015

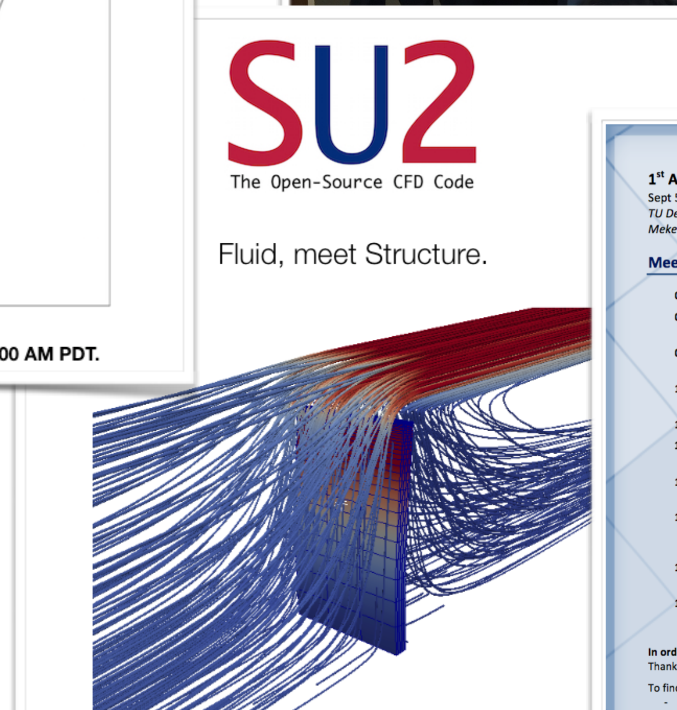




The open-source SU2 package for CFD analysis and design serves not only as a useable example to computational scientists, but also as a common baseline for future development by the entire community. The current open-source model has enabled the leading experts across many technical areas, anywhere in the world, to work together in creating new capabilities that would not have materialized in the absence of collaboration. Today, we demonstrate this once again with the release of SU2 version 4.1 "Cardinal."

Download SU2 v4.1

Through collaboration with the SciComp Team at TU Kaiserslautern, we are proud to introduce the support of Algorithmic Differentiation (AD). Based on ... interfaces, this enables exact derivative computations throughout ... implementation uses the recently released open-source library, C... derivatives of all occurring operations and to evaluate the final gra... C++ features, like static polymorphism and expression templates, ... advanced AD methods (preaccumulation, externally differentiated ... communication) result in a low memory footprint and fast evaluati...



The open-source SU2 package for CFD analysis and design was conceived as a ... for multi-physics research. We've been hard at work improving our C++ class arch... more easily support the addition of new physical models and their coupling at a high-level, and today we are releasing a powerful new example in the form of a fluid-structure interaction (FSI) capability embedded within SU2 version 4.2 "Cardinal."

Download SU2 v4.2

1st Annual SU2 Developers Meeting
Sept 5th, 2016
TU Delft, AULA Conference Center, Commissie Kamer 3
Mekelweg 5, 2628 CC Delft, Netherlands

SU2
The Open-Source CFD Code

Meeting Agenda

- **09.00 – 09.15: Welcome & Introduction**
- **09.15 – 09.35: SU2: Overview of History, Status, and Future Developments**
  Prof. Juan J. Alonso & Dr. Thomas D. Economon, Stanford University
- **09.35 – 10.00: NICFD (Non Ideal Compressible Fluid Dynamics) in the SU2 Framework**
  Prof. Alberto Guardone, Politecnico di Milano (presenter), Profs. Piero Colonna & Matteo Pini, TU Delft
- **10.00 – 10.25: Automatic Differentiation Discrete Adjoints Using SU2**
  Prof. Nicolas Gauger, TU Kaiserslautern
- **10.25 – 10.35: Coffee Break**
- **10.35 – 11.00: Development of a High-Order Discontinuous Galerkin Fluid Solver Within SU2**
  Prof. Edwin van der Weide, University of Twente
- **11.00 – 11.25: Fluid-Structure Interaction Problems Using Native and External Structural Solvers Coupled to SU2**
  Prof. Rafael Palacios (presenter) & Mr. Ruben Sánchez, Imperial College, Prof. Vincent Terrapon & Mr. David Thomas, Université de Liège
- **11.25 – 11.50: Turbomachinery Simulations Using SU2**
  Profs. Matteo Pini (presenter) & Piero Colonna, Mr. Salvatore Vitale, Mr. Antonio Rubino, TU Delft Prof. Alberto Guardone, Mr. Giulio Gori Politecnico di Milano
- **11.50 – 12.15: Mesh Adaptation for SU2 with the INRIA AMG Library**
  Prof. Juan J. Alonso & Dr. Thomas D. Economon, Stanford University
- **12.15 – 12.45: SU2 Development Priorities for the Next Year / Discussion**
  Prof. Juan J. Alonso (moderator), all attendees

In order to participate (in-person or virtually), please register for the meeting by following the link on the SU2 home page (http://su2.stanford.edu).
Thanks for your interest and note that all stated times are Central European Summer Time (CEST).

To find more information about SU2 or to get involved, please visit the following pages:
- SU2 on GitHub: https://github.com/su2code/SU2
- SU2 Forum on CFD Online: http://www.cfd-online.com/Forums/su2/
- Follow SU2 on Twitter: https://twitter.com/su2code

Stanford ENGINEERING Aeronautics & Astronautics · Technische Universität Kaiserslautern · TU Delft · Imperial College London · University of Twente. · Argonne National Laboratory · Technion Israel Institute of Technology

Jan 7
SU2 v4.1

Feb 29
NASA LBFD
Announced

Apr 5
Pointwise,
Tecplot,
SU Solar Car,
SU2 Webinar

Jun 15
SU2 v4.2

July / August
U of Liege
Visits Stanford

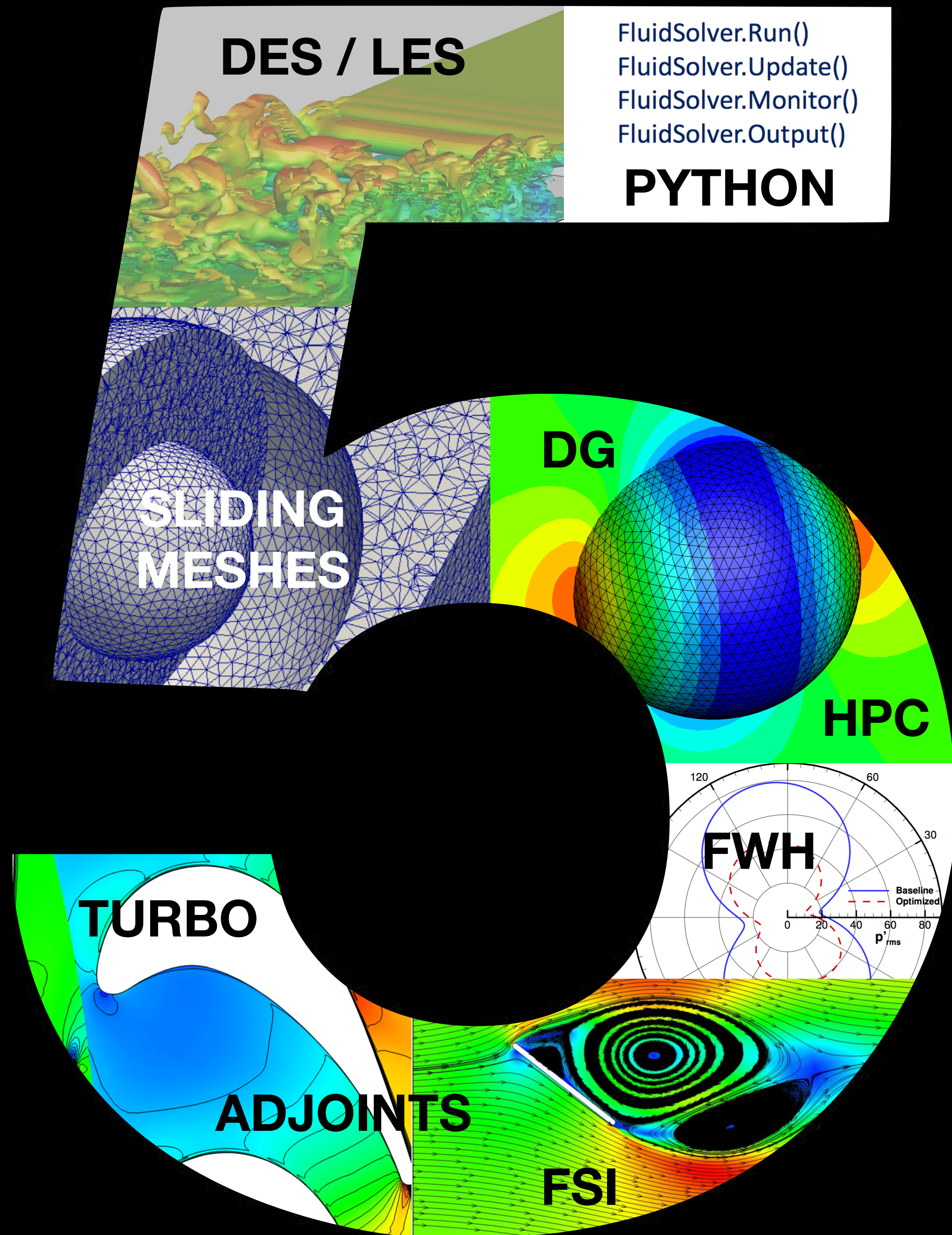
Université de Liège

Aug
SU2 v4.3

Sept 5
1st Annual SU2
Developers Meeting

Winter | Spring | Summer | Fall

2016



SU2 v5.0 (released on January 19, 2017) is just the beginning.
Here's a sneak peak at *just some* of the things we're working on for the future…

Research in SU2

AIAA Aviation
13-17 June 2016, Washington, D.C.
17th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference

17th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference, 13 – 17 June 2015, Washington, D.C.

Efficient Aerodynamic Design using the Discrete Adjoint Method in SU2

Tim Albring*, Max Sagebaum† and Nicolas R. Gauger‡
TU Kaiserslautern, Kaiserslautern, 67663, Germany

I. Introduction

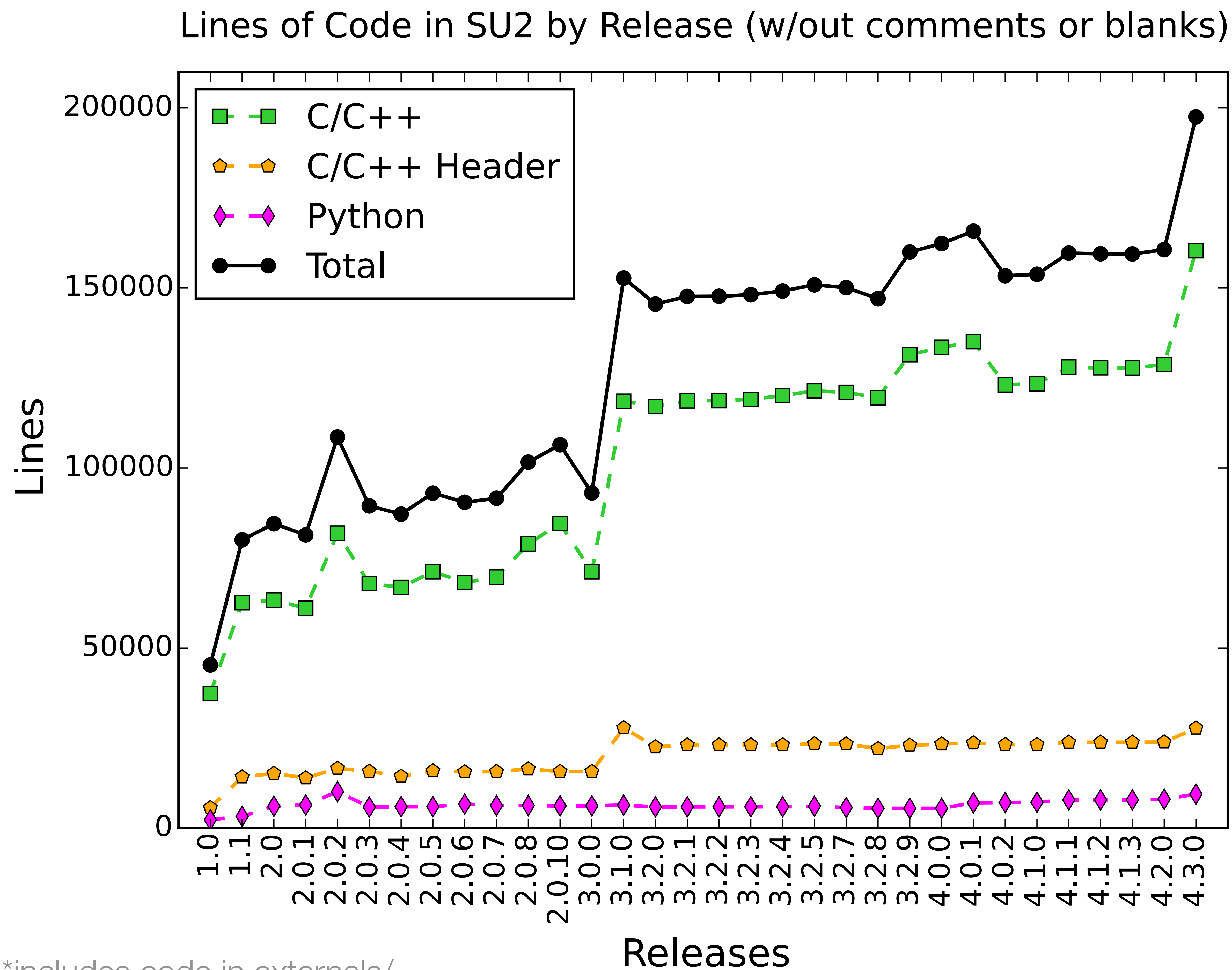
Almost 30 years have passed since Jameson[1] discussed the success and challenges in CFD and listed optimization and design as one the directions for future research. Unfortunately, computational methods for aerodynamic analysis are even today far from being incorporated in automatic design procedures. Although adjoint methods[2,3,4] have greatly decreased the computational effort and impressive results using the continuous[5] and discrete[8] adjoint approaches where published during the last few years, there are still open issues regarding the robustness, (discrete) consistency and generality for complex problems in turbulent flows. Typically, one has to make a compromise between efficiency and the latter properties while choosing an appropriate approach for a given problem. In general there exist different approaches to construct and solve the discrete adjoint system of equations. Most of them require the exact linearization of the flow residual, which is in contrast to the flow solver itself, where some approximation is most of the time sufficient to yield convergence. Depending on the complexity of the numerical methods the linearization by hand is time-consuming and error-prone. Furthermore it lacks the capability of adapting to changes in the flow solver. One way to circumvent this problems is the use of Algorithmic Differentiation (AD) applied to parts of the flow solver[6,7] to construct the Jacobian. Although it reduces the error-proneness, it still requires the manual application of AD to all subroutines involved in the computation of the residual. Even if we have the exact Jacobian, it is typically ill-conditioned so that applying a Krylov-method can be inefficient. This is often visible when including turbulence models. Here, *Duality-Preserving* methods can be useful as they guarantee to have the same convergence rate as the flow solver. These methods were originally suggested by Korivi and Newman,[9,10] albeit called *Incremental Iterative Form*. Until today they are only used by a manageable amount of people.[11,12,13]

In this paper we want to show that by exploitation of the fixed-point structure of the flow solver it is possible to derive a duality-preserving iteration to solve the adjoint system. All occurring gradients can be constructed by applying AD to the top-level routine of the flow solver, thereby eliminating the manual construction of the exact Jacobi matrix. Furthermore, we apply advanced AD techniques like expression templates[24,25] and local preaccumulation[14] to automatically generate a representation of the computational graph of each expression at compile-time. This results in competitive performance while still maintaining flexibility. Due to the use of AD the extension to new turbulence models, transitions models, fluid models or objective functions is straightforward. To contribute to the open-source idea, we therefore tightly integrated the discrete adjoint solver along with the AD features into the open-source framework SU2,[16,17] in order for the community to explore new and interesting optimization problems. For this reason we also dedicate a section to give a short overview on the implementation.

*PhD. Candidate, Chair for Scientific Computing, TU Kaiserslautern, 67663 Kaiserslautern, Germany, AIAA Student Member.
†PhD. Candidate, Chair for Scientific Computing, TU Kaiserslautern, 67663 Kaiserslautern, Germany.
‡Professor, Chair for Scientific Computing, TU Kaiserslautern, 67663 Kaiserslautern, Germany, AIAA Senior Member.

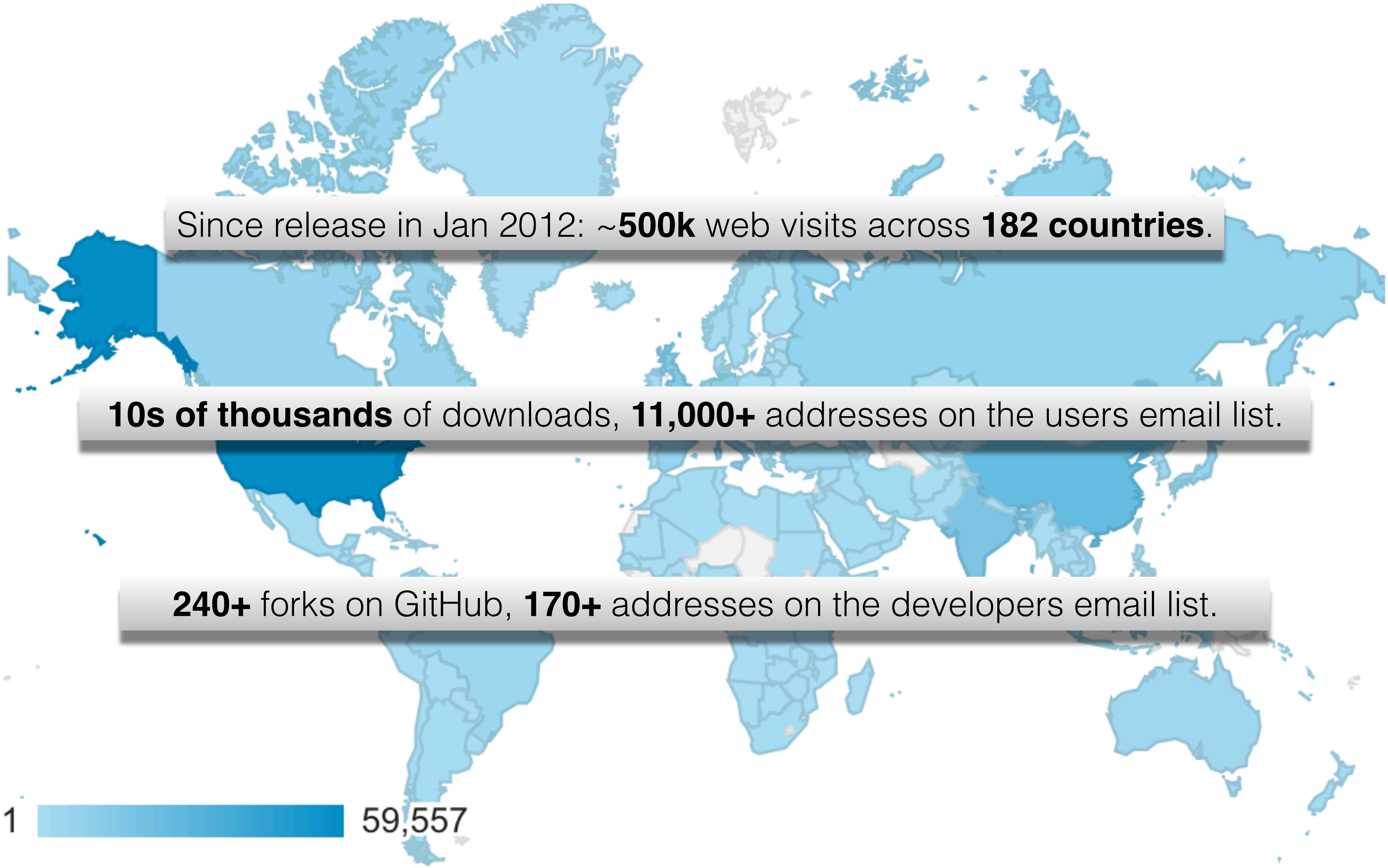
...ent Unsteady Aerodynamic and Aeroacoustic Design Framework Using Discrete Adjoint

Beckett Y. Zhou*, Tim Albring,† and Nicolas R. Gauger‡
Chair for Scientific Computing, TU Kaiserslautern
Bldg 34, Paul-Ehrlich-Strasse, 67663 Kaiserslautern, Germany
...arlos R. Ilario da Silva§, Thomas D. Economon,¶ and Juan J. Alonso‖
Department of Aeronautics and Astronautics, Stanford University
Stanford, CA 94305, U.S.A.



*includes code in externals/

Where are we today? Everywhere.



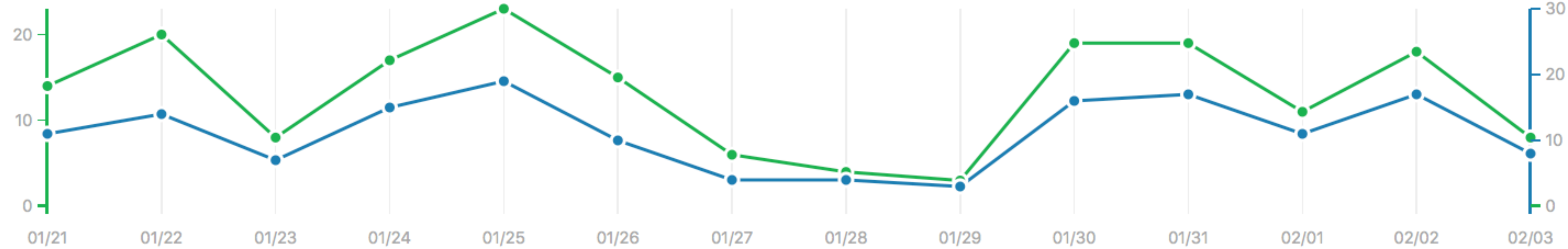
Since release in Jan 2012: ~**500k** web visits across **182 countries**.

10s of thousands of downloads, **11,000+** addresses on the users email list.

240+ forks on GitHub, **170+** addresses on the developers email list.

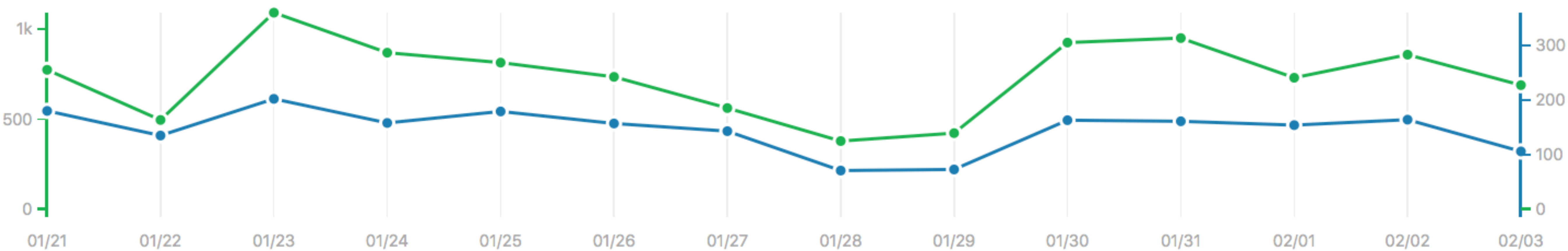
Webhits at su2.stanford.edu. Data as of 2017.02.03.

Git clones



185 Clones	**135** Unique cloners

Visitors



10,295 Views	**1,513** Unique visitors

Traffic data from the SU2 GitHub repository (accessed 2017.02.03).

Getting Started: Downloading SU2

Multiple options…

1. Check out the download portal on the main website to register and download binaries or source for v5.0.0:

   http://su2.stanford.edu/download.html

2. Download releases (latest and older) from GitHub here:

   https://github.com/su2code/SU2/releases

3. **Recommended**: Clone the open repository directly at the command line to get the latest release. Note: the master branch (default) is always stable:

   $ git clone https://github.com/su2code/SU2.git

SU2 v5.0.0 "Raven" was released on January 19, 2017, the 5th anniversary of SU2.

Documentation / Tutorials

- A large body of documentation is available!
- Main documentation found on the SU2 GitHub wiki linked on the main page under “Guides”:
    - https://github.com/su2code/SU2/wiki
    - Detailed information on installation, input and output files, etc.
    - Contains step-by-step tutorials.
    - Wiki-style docs that the developers maintain.
    - See current contents to the right…

- Home
- Quick Start
- User Docs
    - Software Components
    - Download
    - Installation
        - Build from Source
            - Simple Build
            - Parallel Build
            - AD Build
            - Python Wrapper Build
        - Windows Installation
        - Windows Demo
        - Cygwin Build for Windows
    - Input Files
        - Configuration File
        - Mesh File
        - Restart File
    - Execution
    - Post-processing
    - Test Cases

- User Tutorials
    - Inviscid Bump in a Channel
    - Inviscid Supersonic Wedge
    - Inviscid ONERA M6
    - Laminar Flat Plate
    - Laminar Cylinder
    - Turbulent Flat Plate
    - Transitional Flat Plate
    - Turbulent ONERA M6
    - Optimal Shape Design of a Transonic Airfoil
    - Constrained Optimal Shape Design of a Fixed Wing
- Developer Docs
    - Gitting Started
    - Developing SU2 on GitHub (Internal Developers)
    - Running Regression Tests
    - Code Review
    - Code Structure
    - Style Guide
    - Advanced AD Techniques
- FAQ
- License
- Contact

Documentation / Tutorials

- Additional training materials:
  http://su2.stanford.edu/training.html
  - Links to tutorials, presentations, files, and videos.

- Active forum on CFD Online:
  http://www.cfd-online.com/Forums/su2/

- AIAA publications with many technical details on physics, numerical methods, and V&V:
  - SU2: An Open-Source Suite for Multi-Physics Simulation and Design, AIAA Journal, Vol. 54, No. 3 (2016), pp. 828-846.
  - (SU2): An open-source integrated computational environment for multi-physics simulation and design, AIAA Paper 2013-0287.
  - Stanford University Unstructured (SU2): Open-source analysis and design technology for turbulent flows, AIAA Paper 2014-0243.

SU2 Winter Workshop
Feb 3rd, 2017
13:00 – 16:00, PST
Stanford, CA 94305

SU2
The Open-Source CFD Code

Meeting Agenda

Part I

13.00 – 13.15: Welcome & Introduction

13.15 – 13.35: Tutorial 1: Basic Analysis & Configuration Options
Running SU2 & familiarization with analysis options & capabilities.

13.35 – 13.45: Q&A

13.45 – 14.05: Tutorial 2: Python Scripts & Optimization Problems
Advanced features of SU2, inputs to the SU2 python scripts.

14.05 – 14.15: Q&A

14.15 – 14.30: Coffee Break

Part II

14.30 – 14.45: Code Structure & Locally Modifying the Code
Understanding how SU2 works & how to modify it.

14.45 – 14.50: Q&A

14.50 – 15.00: Introduction to Github and SU2 Development Best Practices
How to share your changes to SU2 with the world.

15.00 – 15.45: Interactive Exercise: Modifying a Python Script
***Recommended:** bring an idea of a problem that require running several small CFD solutions sequentially – for example a sweep of an input parameter or uncertainty quantification.*

15.45 – 16.00: Open Discussion

In order to participate (in-person or virtually), please register for the meeting by following the link included in the email announcement.
Thank you for your interest in SU2. Please make sure to install SU2 and run at least one tutorial prior to the workshop. (See https://github.com/su2code/SU2/wiki)

To find more information about SU2 or to get involved, please visit the following pages:
- SU2 on GitHub: https://github.com/su2code/SU2
- SU2 Forum on CFD Online: http://www.cfd-online.com/Forums/su2/
- Follow SU2 on Twitter: https://twitter.com/su2code

Stanford | ENGINEERING
Aeronautics & Astronautics

Institutions that have downloaded SU2. Sized by frequency.