

Welcome and Introduction to SU2

SU2 WINTER WORKSHOP
FEBRUARY 3RD, 2017

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



What is SU2?

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

 CrossMark
click for updates

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](https://doi.org/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

A^c	= Jacobian of the convective flux with respect to U	f	= force vector on the surface
A^{vk}	= Jacobian of the viscous fluxes with respect to U	\bar{I}	= identity matrix
B	= column vector or matrix B , unless capitalized symbol clearly defined otherwise	J	= cost function defined as an integral over S
B	= (B_x, B_y) in two dimensions, or (B_x, B_y, B_z) in three dimensions	j	= scalar function defined at each point on S
B^T	= transpose operation on column vector or matrix B	k	= turbulent kinetic energy
b	= spatial vector $b \in \mathbb{R}^n$, where n is the dimension of the physical Cartesian space (in general, two or three)	$N(i)$	= set of all neighboring nodes of node i
C_D	= coefficient of drag	n	= unit normal vector
C_L	= coefficient of lift	P	= shear-stress transport turbulent kinetic energy production term
C_{M_y}	= pitching-moment coefficient	Pr_d	= dynamic Prandtl number
C_p	= coefficient of pressure	Pr_t	= turbulent Prandtl number
c	= airfoil chord length	p	= static pressure
c_p	= specific heat at constant pressure	Q	= vector of source terms
\bar{D}^{vk}	= Jacobian of the viscous fluxes with respect to ∇U	q_ρ	= generic density source term
d_s	= nearest wall distance	$q_{\rho E}$	= generic density source term
d	= force projection vector	$q_{\rho v}$	= generic momentum source term
E	= total energy per unit mass	R	= gas constant
\bar{F}_c^{ij}	= numerical convective flux between nodes i and j	$\mathcal{R}(U)$	= system of governing flow equations
\bar{F}_{vk}^{ij}	= numerical viscous fluxes between nodes i and j	Re	= Reynolds number
F_c^{ij}	= convective flux	\mathcal{R}_i	= system of governing equation residual at node i
F_{vk}	= viscous fluxes	S	= solid wall flow domain boundary
		\bar{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, $n_S \cdot \nabla(\cdot)$
		μ_{dyn}	= laminar dynamic viscosity
		μ_{tur}	= turbulent eddy viscosity
		μ^{v1}	= total viscosity as a sum of dynamic and turbulent components, $\mu_{dyn} + \mu_{tur}$
		μ^{v2}	= effective thermal conductivity; $(\mu_{dyn}/Pr_d) + (\mu_{tur}/Pr_t)$
		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. Copyright © 2015 by T. D. Economon, F. Palacios, S. R. Copeland, T. W. Lukaczyk, and J. J. Alonso. Published by the American Institute of Aeronautics and Astronautics, Inc., with permission. Copies of this paper may be made for personal or internal use, or condition that the copier pay the \$10.00 per-copy fee to the Copyright Clearance Center, Inc., 222 Rosewood Drive, Danvers, MA 01923; include the code 1533-385X/15 and \$10.00 in correspondence with the CCC.
*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.

828

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>

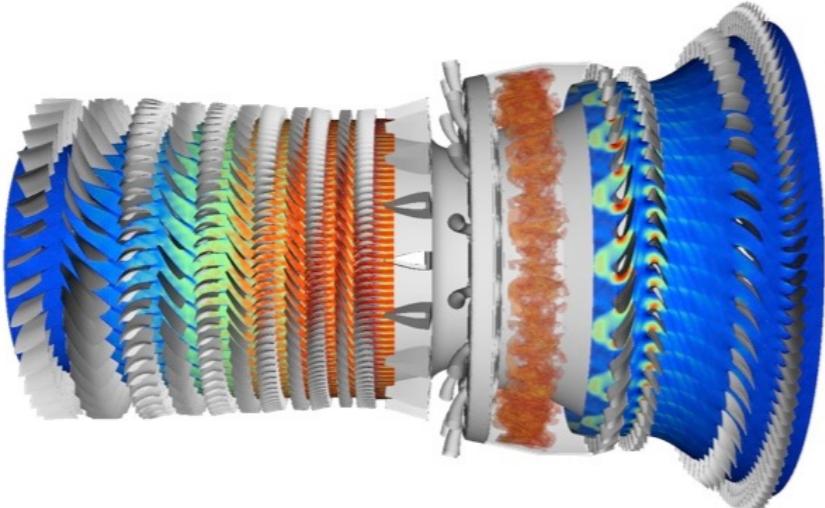
Our Guiding Principles

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



Jan 2011
Francisco joins
ADL @ Stanford

2003-2008

2009

2010

2011

Summer/Fall 2009
Francisco spends
3 months at Stanford

2010
Work on
CADES (predecessor
to SU2) begins

Summer/Fall
Preparations for
releasing SU2 as
open source

“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

SU² Pre-Release Workshop

Presented by Thomas D. Economon
Hosted by the SU² Development Team

Aerospace Design Lab, Stanford University, Stanford, CA 94305, U.S.A.

January 17, 2012

STANFORD UNIVERSITY
Unstructured Code

Jan 17
Pre-release
Workshop



STANFORD | SU²

Stanford University Unstructured

SU²
The New Open Source CFD Code

Computational analysis tools have revolutionized the way we design aerospace systems, but most established codes are proprietary, unavailable, or prohibitively expensive for many users. The SU team is changing this, making computational analysis and design freely available as open-source software and involving everyone in its creation and development.

Analyze. Optimize. Design!

Visualizations About the Code Cite Us

Information Latest News Follow Us

SU² website tops 40,000 visits
SU² version 2.0, release promotional flyer [PDF]
SU² mesh output from Pointwise®.
SU² ver 1.1, release promotional flyer [PDF]
SU² ver 1.0, pre-release workshop presentation [PDF]

SU² AIAA Paper 2013-0287 [PDF]
Presentations of the SU² workshop,
SU² ver 2.0, release promotional
flyer [PDF]
SU² mesh output from Pointwise®.
SU² ver 1.1, release promotional
flyer [PDF]
SU² ver 1.0, pre-release workshop
presentation [PDF]

Tweets Follow @su2code
SU² @su2code
SU², the main generator by Larcena
Engineering Dynamics now supports SU2
mesh format [larcena.com/reviews.html](#)
Expand
SU² @su2code
SU² 2.0.5 (newer version) is now
ADM at Stanford University
21 Jun

Oct 31

[su2.stanford.edu](#) update



<http://su2.stanford.edu>

About SU²

SU² is an open-source software suite specialized in high-fidelity Partial Differential Equation (PDE) analysis and design of PDE-constrained systems on unstructured grids. The suite includes C++ analysis modules, linked via python scripts, that:

- Solve the PDE system
- Decompose the domain for parallel computation
- Determine sensitivities of an objective function (e.g. lift, drag)
- Deform the model and grid to perform shape optimization
- Perform adaptive grid refinement

Mac OS X, Linux and Windows binary executables can be downloaded from the SU² website: <http://su2.stanford.edu>

Open-Source Analysis and Design

SU² is under active development by the Aerospace Design Laboratory at Stanford University. Visit the ADL at: <http://adl.stanford.edu>



<http://su2.stanford.edu>

STANFORD UNIVERSITY UNSTRUCTURED CODE (SU²) RELEASED
TODAY, THURSDAY JANUARY 19, 2012

The First Release of The SU² Open-Source Computational Fluid Dynamics (CFD)
Analysis and Optimization Suite is Out Today

Jan 19
SU2 v1.0
SU2 is born!

Winter

2012

Spring

Jun 25
SU2 v1.1

Summer



Check out [su2.stanford.edu](#) for an open source computational design and analysis tool!

6:59 PM - 25 Sep 2012

Reply to @su2code

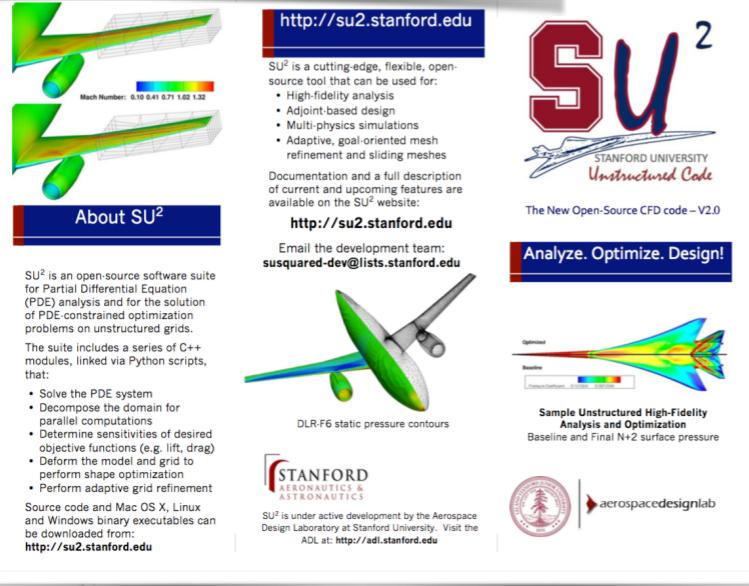
Sept 25
First tweet
@su2code

Fall

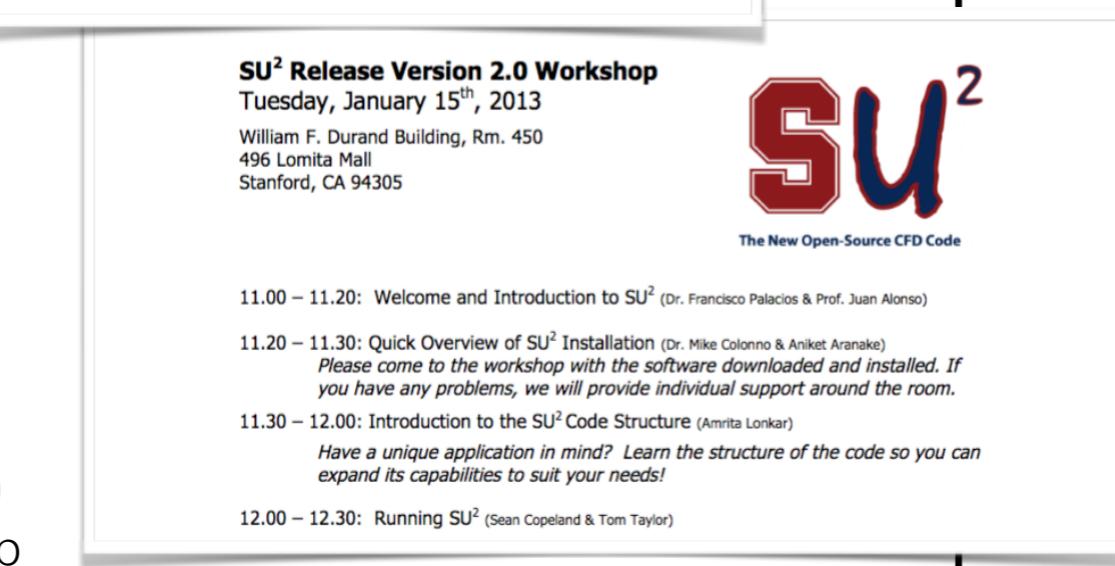
Stanford University Unstructured (SU²): An open-source integrated computational environment for multi-physics simulation and design.

Francisco Palacios*, Michael R. Colombo*,
 Aniket C. Aranake† Alejandro Campos† Sean R. Copeland† Thomas D. Economou†
 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



Jan 15
 SU2 v2.0
 Workshop



May 17 & 25
 SU2's first two PhDs

openMDAO
SU²
 The Open-Source CFD Code

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 start tutorial*.
- 11.30 – 11.45: Short break.
- 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 – 14.00: Brainstorming for ideas for possible projects.
- 14.00 – 16.45: Hack-a-thon. *Work side-by-side writing OpenMDAO/SU² 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 SU².
 - Unsteady RANS simulation. SU² has multitude of capabilities for performing high-fidelity analysis of complex geometries. Learn about them here.
 - Design and Optimization Using SU². Learn why SU² 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/SU² 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 SU² home-page (<http://su2.stanford.edu>).

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 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



Aug 10
 SU2 on GitHub

Winter

Spring

Summer

Fall

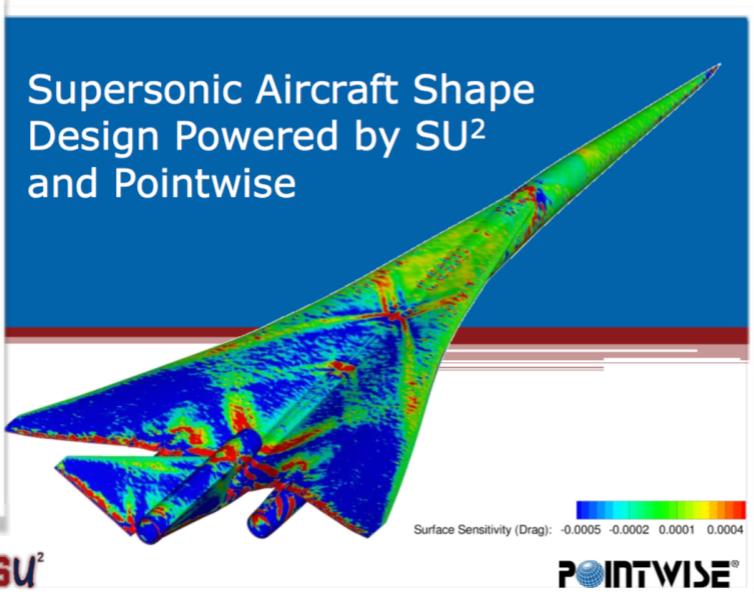
2013



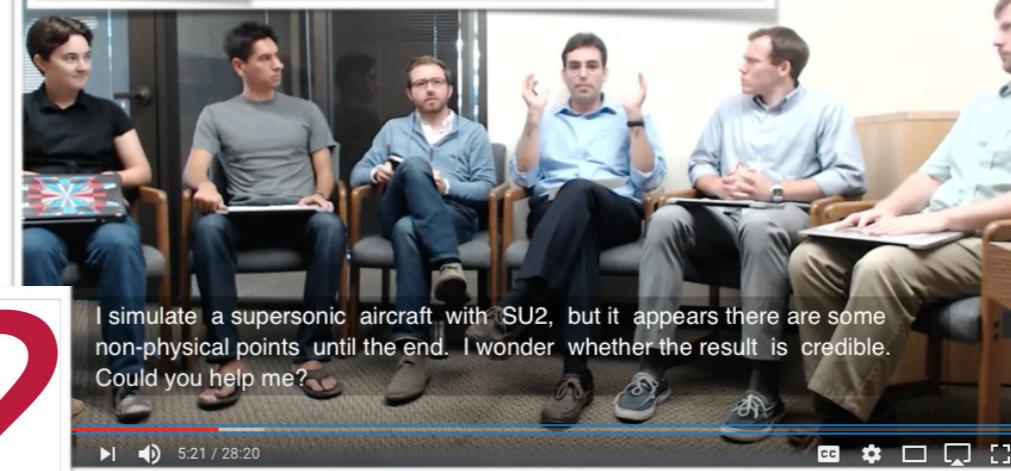
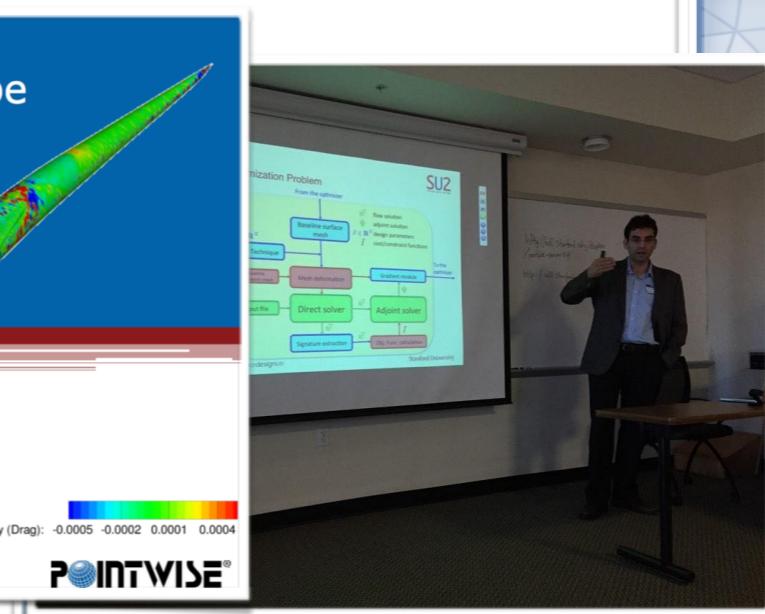
Jan 15
SU2 v3.0
SU2_EDU v1.0



Apr 14
SU2 v3.1



Apr 29
Pointwise-SU2
Webinar



July 29 - New Logo



May 7
Intel Parallel
Comp. Center



Sept 16
Dev
Email List

Winter

Spring

Summer

Fall

2014

Pointwise® and SU2 Joint Workshop

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 perform 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.

Learn how to modify your favorite CFD solver.
Learn how to add new options in the configuration file and codebase through GitHub® pull requests.
Your solver. Do you want to implement your own solver in SU2? Get the solver structure from the original developers of SU2. Use the flexible SU2 capabilities for unsteady problems, including meshes.
led.
rise*:
pic tetrahedral meshing). Clean dirty CAD and learn to generate meshes.
arn to inspect grid quality and locate problems prior to export.
your meshing tasks using our Glyph scripting language.
rtant note: We will host a hands-on session during the more about Pointwise® and SU2.
imes are Pacific Time (PDT).
age (<http://su2.stanford.edu>).
use visit the following:
wise.com

installed (Pointwise, Inc. will provide a 2-day license during the event). If you have any

Sept 29
Pointwise-SU2
Joint Workshop

Sept 24
“This Week in SU2”
Youtube Season 1
Premiere



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

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

Thomas D. Economon^B
Stanford University, Stanford, California 94305
Francisco Palacios^B
The Boeing Company, Long Beach, California 90808
and
Sean R. Copeland^B, Trent W. Lukaczyk^B, and Juan J. Alonso^B
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 standards. SU2 is a multiphysics 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 that can be tightly coupled. 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	
A^c	Jacobian of the convective flux with respect to U
A^{vk}	Jacobian of the viscous fluxes with respect to U
B	column vector or matrix B , unless capitalized symbol clearly defined otherwise
B^T	transpose operation on column vector or matrix B
b	scalar vector $b \in \mathbb{R}^n$, 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	pitch-moment coefficient
C_P	coefficient of pressure
c	airfoil chord length
c_p	specific heat at constant pressure
\bar{D}^{vk}	Jacobian of the viscous fluxes with respect to VU
d	nearest wall distance
d_i	force projection vector
E	total energy per unit mass
F_{ij}^{cv}	numerical convective flux between nodes i and j
F_{ij}^{vi}	numerical viscous fluxes between nodes i and j
F^v	convective flux
F^v_k	viscous flux
f	force vector on the surface
\hat{I}	identity matrix
J	cost function defined as an integral over S
j	scalar function defined at each point on S
J'	turbulent kinetic energy
$N(i)$	set of neighbors of a node i
n	unit normal vector
P	shear-stress transport turbulent kinetic energy production term
P_{r_d}	dynamic Prandtl number
P_{r_t}	turbulent Prandtl number
p	state variable
Q	vector of source terms
q_g	generic density source term
$q_{\dot{g}}$	generic energy source term
q_m	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
T	Spalart-Allmaras turbulence production term
t	temperature
U	time variable
W	vector of conservative variables
W_c	vector of characteristic variables
W_{fa}	far-field characteristic variables
Γ	flow domain boundary
Γ_{fd}	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 derivative of a quantity
$\partial_x(\cdot)$	normal gradient operator at a surface point, $\mathbf{n}_s \cdot \nabla(\cdot)$
μ_{d3}	turbulent eddy viscosity
μ_{t3}	total viscosity as a sum of dynamic and turbulent components, $\mu_{dyn} + \mu_{turb}$
μ^{12}	effective thermal conductivity: $(\mu_{dyn}/Pr_d) + (\mu_{turb}/Pr_t)$
v	flow velocity vector
ρ	flow density
τ	pseudotime

828

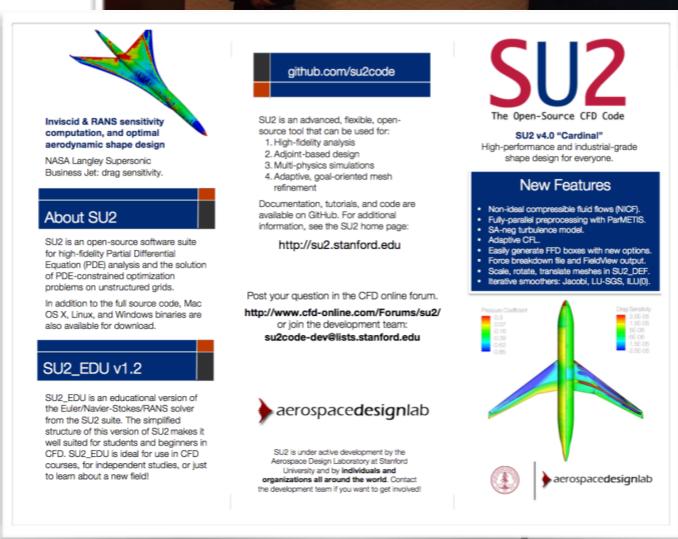
Mar 14 Francisco's farewell from Stanford

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



March
TU Kaiserslautern
Visits Stanford

Winter

2015

Jun 23
SU2 v4.0@ AIAA
AVIATION
Travis CI

Argonne
NATIONAL LABORATORY
August 6
ALCF Theta
ESP Selection

Imperial College
London

Aug - Sept
Imperial College
Visits Stanford

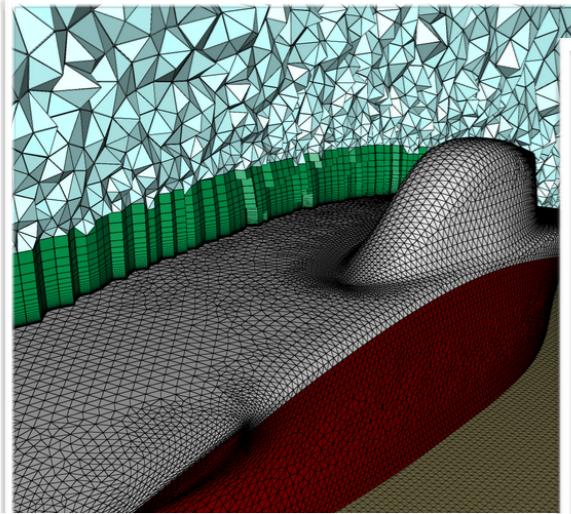
Spring

Summer

Fall

December 29
SU2 in the
AIAA Journal

UNIVERSITY OF TWENTE.



SU2

The Open-Source CFD Code

Continuous or and Discrete.

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](#)

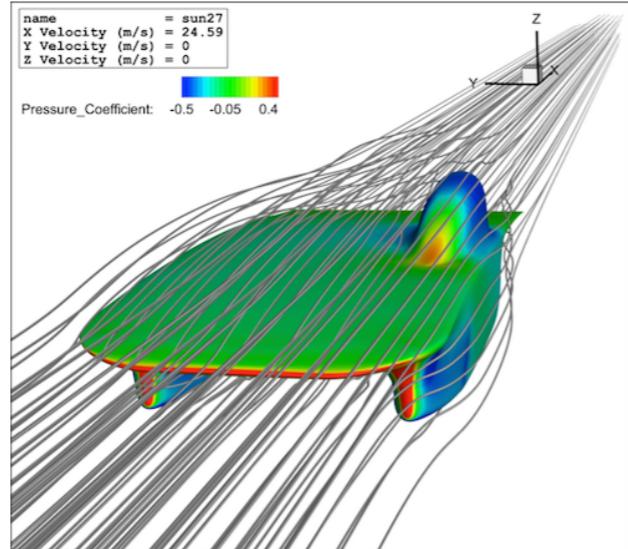
Through collaboration with the SciComp Team at TU Kaiserslautern, we are proud to introduce the support of Algorithmic Differentiation (AD). Based on C++ interfaces, this enables exact derivative computations throughout implementation uses the recently released open-source library, Q. Derivatives of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

Jan 7
SU2 v4.1



Feb 29
NASA LBFD
Announced

The Stanford Solar Car Project's Race for Aerodynamic Efficiency



You are invited to a free webinar on Tue, Apr 5, 2016 8:00 AM - 9:00 AM PDT.

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

Spring

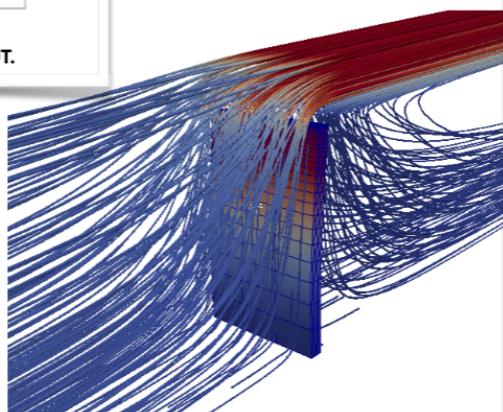
2016



SU2

The Open-Source CFD Code

Fluid, meet Structure.



The open-source SU2 package for CFD analysis and design was conceived as a tool for multi-physics research. We've been hard at work improving our C++ class architecture to 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](#)

Jun 15
SU2 v4.2

Summer

July / August
U of Liege
Visits Stanford



Université
de Liège
Aug
SU2 v4.3

Sept 5
1st Annual SU2
Developers Meeting

SU2

The Open-Source CFD Code

1st Annual SU2 Developers Meeting

Sept 5th, 2016
TU Delft, AULA Conference Center, Commissie Kamer 3
Mekelweg 5, 2628 CC Delft, Netherlands

Meeting Agenda

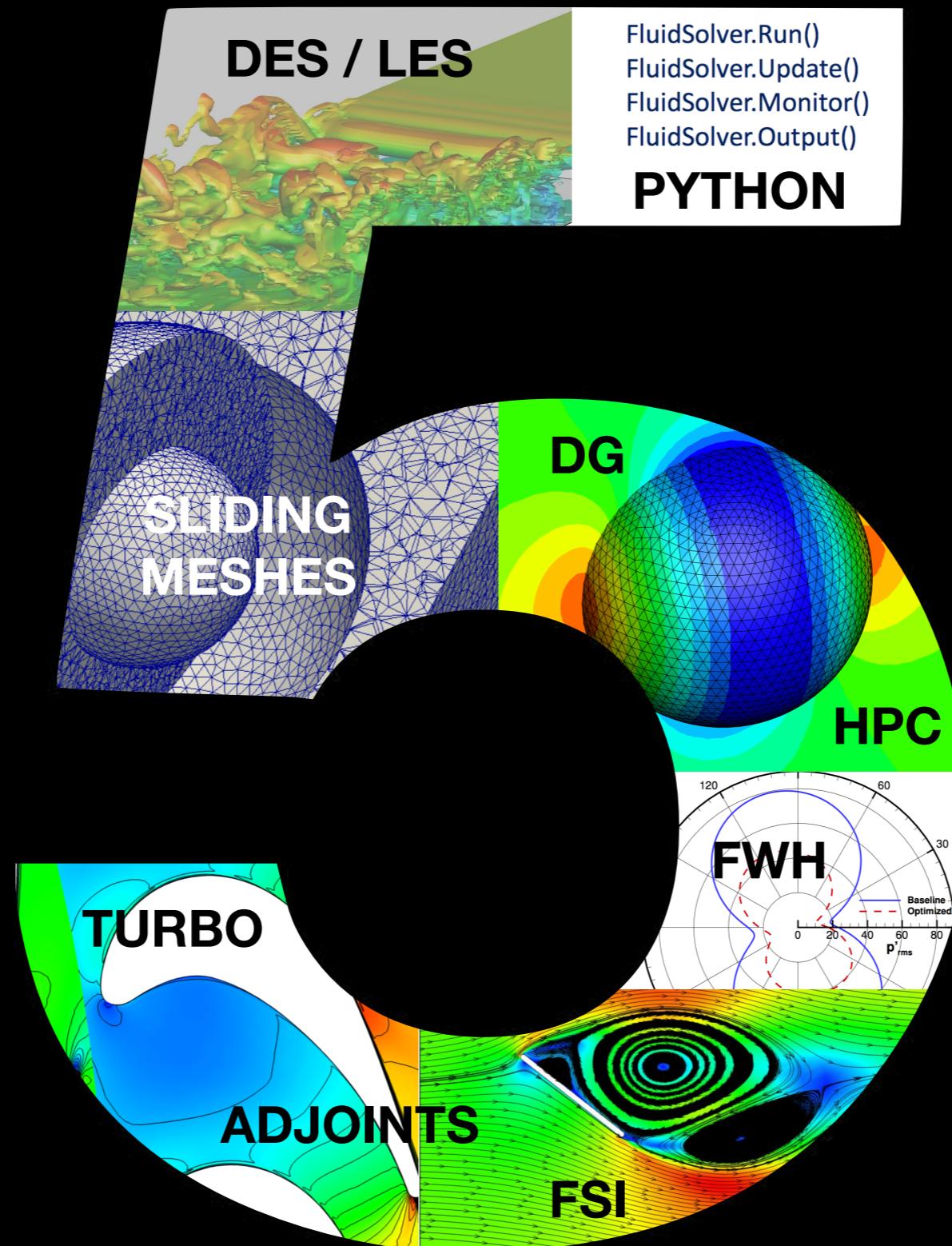
- 09.00 – 09.15: Welcome & Introduction
- 09.15 – 09.35: SU2: Overview of History, Status, and Future Developments
Prof. Juan J. Alonso & Dr. Thomas D. Economon, Stanford University
- 09.35 – 10.00: NICFD (Non Ideal Compressible Fluid Dynamics) in the SU2 Framework
Prof. Alberto Guardone, Politecnico 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 Sanchez, Imperial College, Prof. Vincent Trappou & 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 | TU Delft | Imperial College London | UNIVERSITY OF TWENTE | Argonne | TECHNION | Israel Institute of Technology



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 SciTech
5-9 January 2015, Kissimmee, Florida
53rd AIAA Aerospace
AIAA Aviation
13-17 June 2016, Washington, D.C.
17th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference

51st AIAA
07 - 10 Jan
AIAA
22-25 June 2016
16th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference, 13 – 17 June 2015, Washington, D.C.

CrossMark

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 Jameson¹ discussed the success and challenges in CFD and listed optimization and design as one of 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 continuous⁵ and discrete⁶ adjoint approaches were 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 preaccumulation¹⁴ 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, transition 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.

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

Copyright © American Institute of Aeronautics and Astronautics. All rights reserved.

1 of 15

American Institute of Aeronautics and Astronautics

Copyright © 2016 by the American Institute of Aeronautics and Astronautics, Inc. All rights reserved.

American Institute of Aeronautics and Astronautics

Copyright © 2016 by the American Institute of Aeronautics and Astronautics, Inc. All rights reserved.

AIAA 2016-3318

CrossMark

Optimization Conference

7th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference, 13 – 17 June 2016, Washington, D.C.

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*

Carlos R. Ilario da Silva[§], Thomas D. Economon,[¶] and Juan J. Alonso ||
*Department of Aeronautics and Astronautics, Stanford University
Stanford, CA 94305, U.S.A.*

aper, we present an unsteady aerodynamic and aeroacoustic optimization framework in which differentiation (AD) is applied to the open-source multi-physics solver SU2 to obtain design sensitivities. An AD-based consistent discrete adjoint solver is developed which directly inherits the convergence behavior of the primal flow solver due to the differentiation of the entire nonlinear fixed-point iterator. In addition, an AD-based far-field noise prediction framework using a permeable surface Ffowcs Williams-Hughes approach in frequency domain is also developed. The resultant AD-based discrete adjoint solver is applied to both aerodynamic and aeroacoustic optimization problems. The results suggest that the unsteady aerodynamic and aeroacoustic optimization provided by this AD-based discrete adjoint framework is accurate and robust, due to the differentiation of the entire design chain including the dynamic mesh movement routine and variance model, as well as the hybrid CFD-CAA model. This study also shows the aerodynamic and aeroacoustic objectives to be mutually competing.

I. Introduction

al decades have seen significant progresses in the numerical methods for the design and optimization of aircraft. Since the advent of adjoint-based methods,^{1,2} for which the computational cost is independent of the number of design variables, researchers have been able to tackle many large-scale and practical problems, such as aerodynamic and aerostructural optimizations of complete aircraft configurations.^{3,4} In most applications, the flow field is considered to be in a steady state, as is evident from the rich body of literatures on the numerical shape optimization with steady Euler and Reynolds-averaged Navier-Stokes equations (RANS). However, many aerospace problems are unsteady in nature, such as active flow control, turbomachinery, aeroelasticity, and wind tunnel testing. In comparison to the large strides made in its steady counterpart-based optimization has not received as much attention and methods available to address such problems are currently less mature. This has been primarily due to the need to store prohibitively large amount of data required to solve the unsteady adjoint equation. Furthermore, many unsteady problems involve moving meshes. The need to accurately account for the requisite mesh movement in the governing equations and the resulting adjoint equations introduces additional difficulties.

With the growth of computing power and the improvement of time-accurate numerical methods, has led to significant progress in this area over the past decade. The renewed interest in reduction of aircraft noise – a key phenomenon, due to the ever-stringent aviation noise regulations, also served as a necessary motivation. Zingg and Zingg developed a discrete adjoint formulation for unsteady aerodynamic shape optimization

Student Member AIAA, yuxiang.zhou@scicomp.uni-kl.de
im.albring@scicomp.uni-kl.de
Member AIAA, nicolas.gauger@scicomp.uni-kl.de
Research Associate, Associate Member AIAA, carlos.ilario@stanford.edu
Student Member AIAA, economon@stanford.edu
Fellow AIAA, jjalonso@stanford.edu

Copyright © 2016 by the American Institute of Aeronautics and Astronautics, Inc. All rights reserved.

1 of 18

American Institute of Aeronautics and Astronautics

Copyright © 2016 by the American Institute of Aeronautics and Astronautics, Inc. All rights reserved.

American Institute of Aeronautics and Astronautics

Copyright © 2016 by the American Institute of Aeronautics and Astronautics, Inc. All rights reserved.

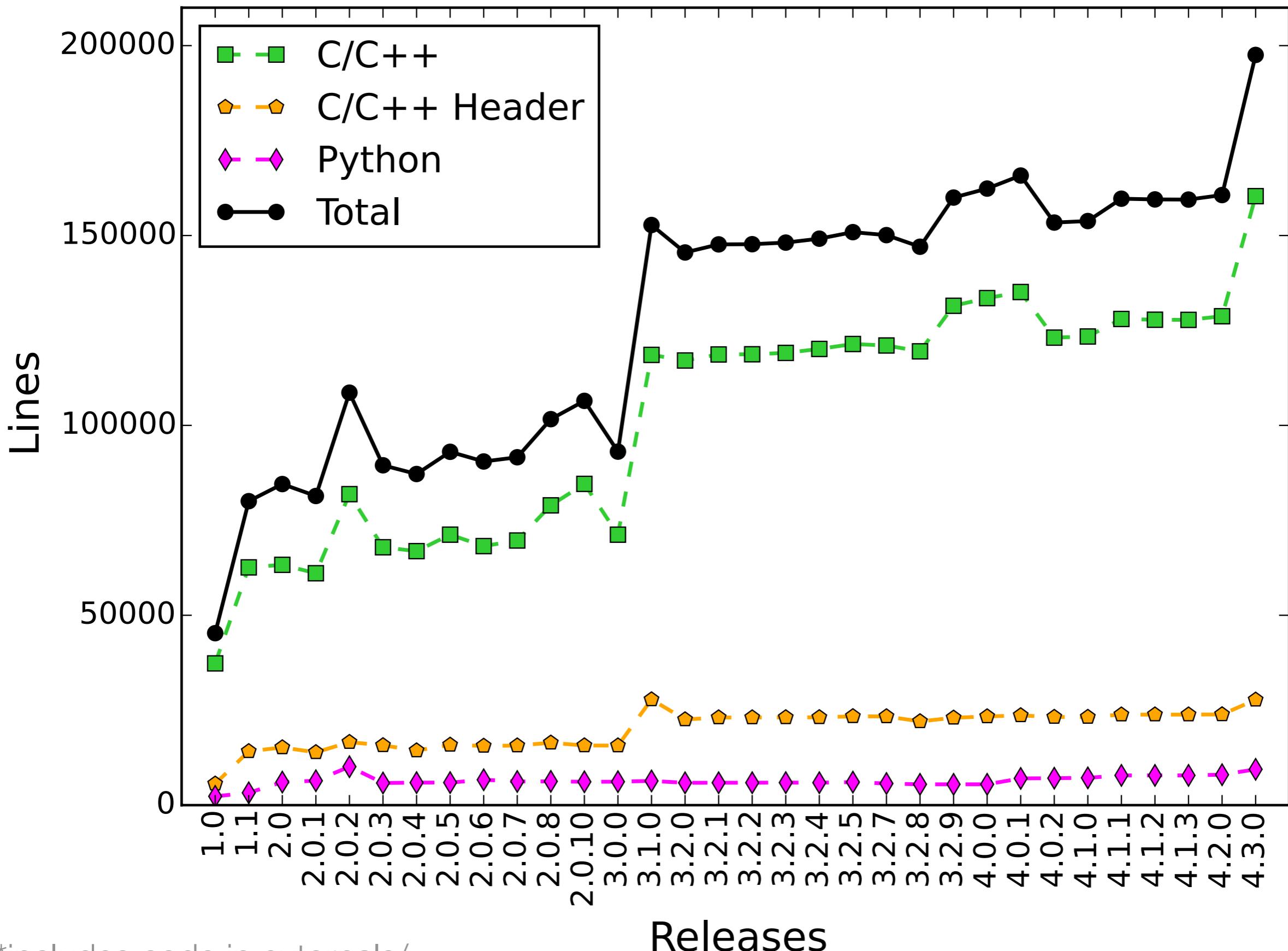
AIAA 2016-3369

CrossMark

Optimization Conference

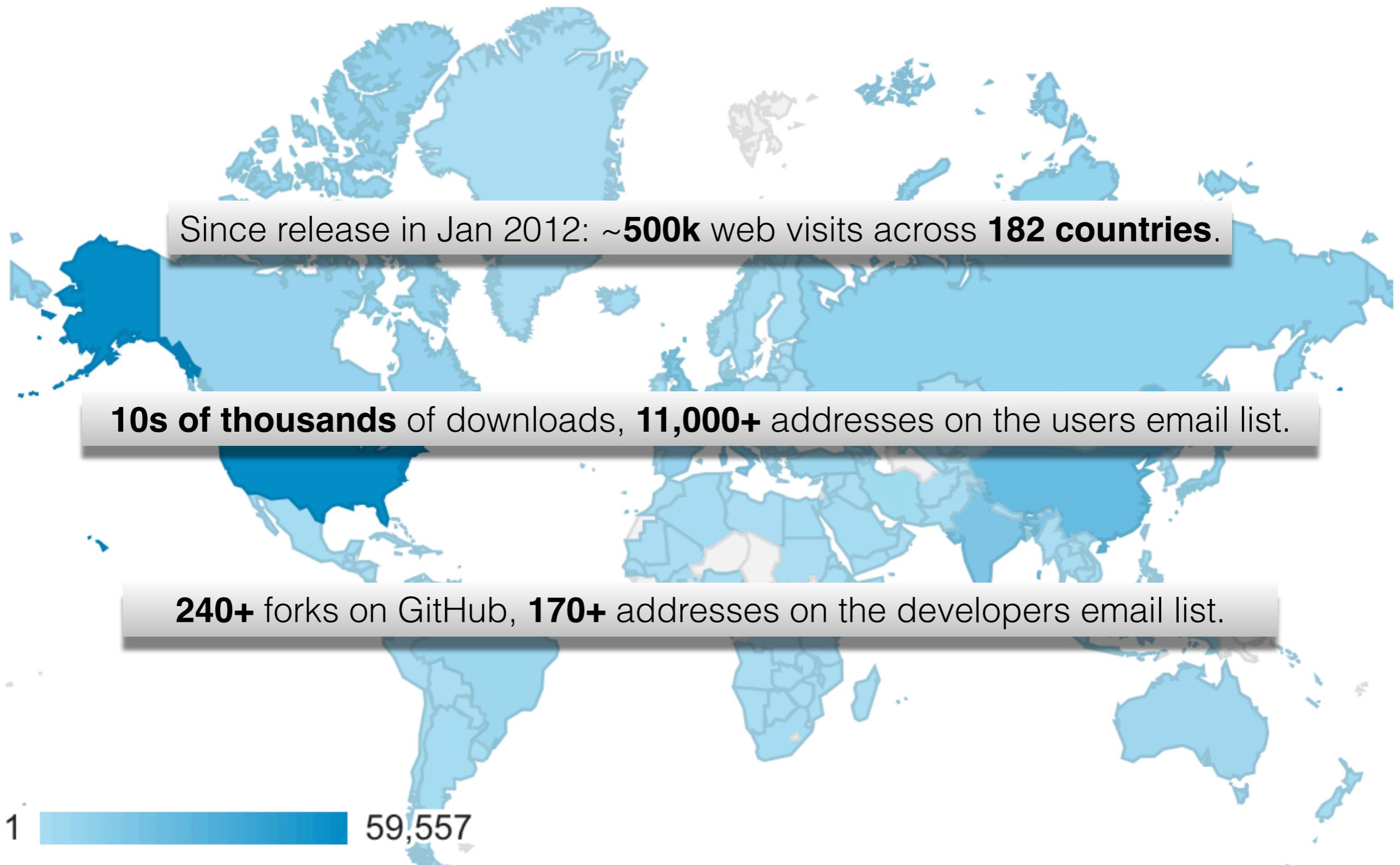
7th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference, 13 – 17 June 2016, Washington, D.C.

Lines of Code in SU2 by Release (w/out comments or blanks)

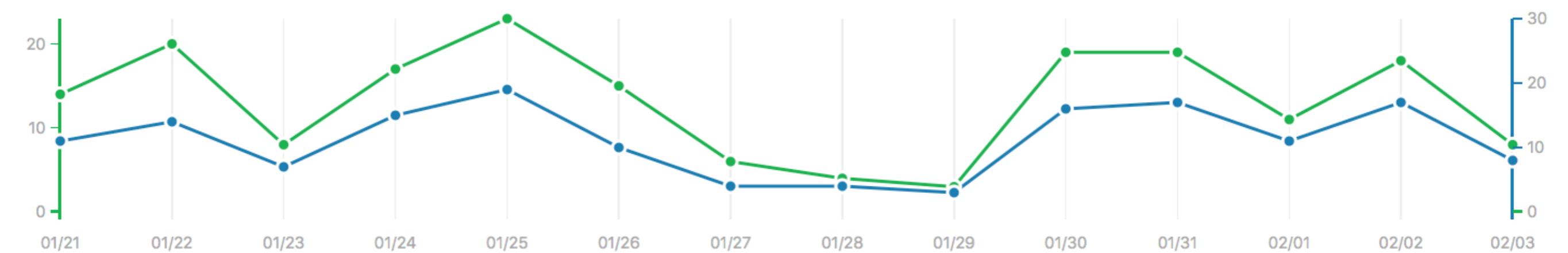


*includes code in externals/

Where are we today? Everywhere.



Git clones



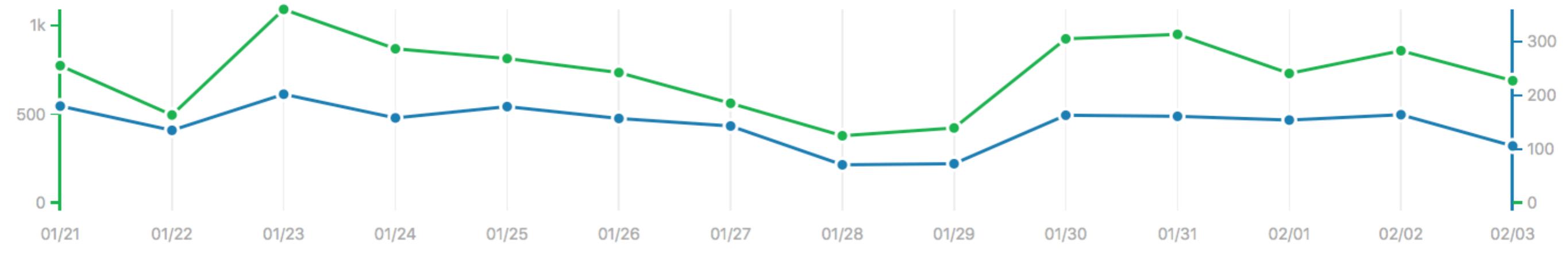
185

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...

The image shows a screenshot of the SU2 GitHub wiki's navigation menu. The 'User Docs' section is expanded, displaying sub-sections for Software Components, Download, Installation (with sub-sections for Build from Source, Windows Installation, and Cygwin Build for Windows), Input Files (with sub-sections for Configuration File, Mesh File, and Restart File), Execution, Post-processing, and Test Cases. To the right of the main menu, a sidebar displays the 'User Tutorials' section, which is also expanded, listing various examples such as 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, Getting Started, Developing SU2 on GitHub (Internal Developers), Running Regression Tests, Code Review, Code Structure, Style Guide, Advanced AD Techniques, FAQ, License, and Contact.

- 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
 - Getting 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



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>



Institutions that have downloaded SU2. Sized by frequency.