OSD : Onera Scientific Day

ONERA
THE FRENCH AEROSPACE LAB

SYSTEMATIC
PARIS REGION SYSTEMS & ICT CLUSTER

CFD Workflow : Meshing, Solving, Visualizing, . . .

October 3, 2012

organization committee :
– ***C. Benoit (Onera)***
– ***Ph. d'Anfray (CEA, Aristote)***
– ***T.H. Lê (Onera)***

Amphithéâtre Becquerel, École Polytechnique, Palaiseau

http://www.association-aristote.fr
info@association-aristote.fr Edition du 19 vendémiaire an CCXXI (*vulg.* 10 octobre 2012)

ARISTOTE Association Loi de 1901. Siège social : CEA-DSI CEN Saclay
Bât. 474 91191 Gif-sur-Yvette Cedex.
Secrétariat : Aristote, École Polytechnique, 91128 Palaiseau Cedex.
Tél. : +33(0)1 69 33 99 66 Fax : +33(0)1 69 33 99 67
Courriel : Marie.Tetard@polytechnique.edu
Site internet http://www.association-aristote.fr

Table of contents

Chapitre 1

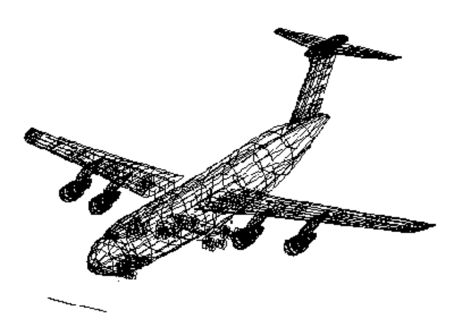
OSD Program

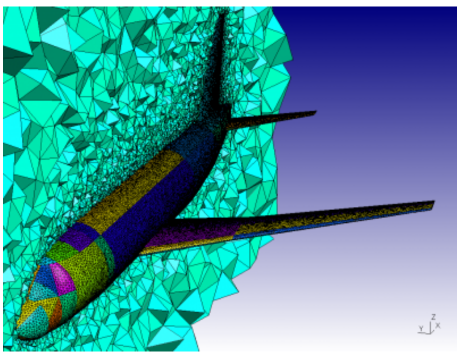
1.1 Introduction

Due to the increasing complexity of numerical simulations, the number of numerical components intervening in a simulation is growing. Today, it is common to run mesher, CFD solver, CSM solver, optimizer, complex post-processing and even runtime visualization together.

Managing a flexible and high-performing workflow has become a crucial topic. This seminar will cover industrial needs, Software editor answers, Research centers up-to-date techniques and future trends concerning the numerical simulation workflow.

OSD École Polytechnique october 3, 2012



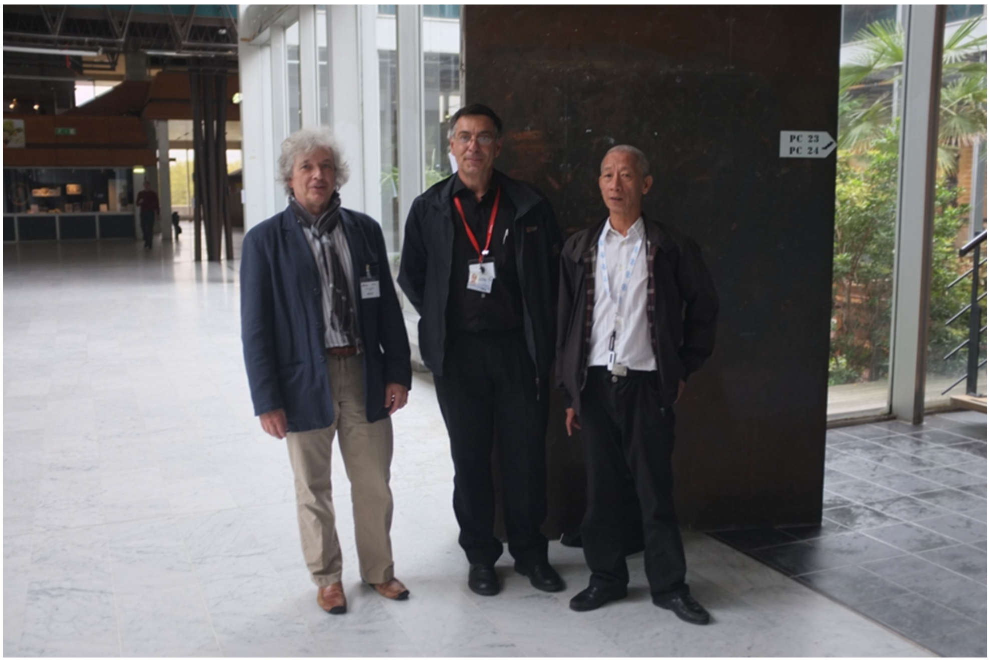


1.2 Agenda, october 3, 2012, morning

8h30-9h00	*Welcoming (and coffee)*	
9h00-9h05	**Ph. d'Anfray** (Aristote) : Opening	
9h05-9h30	**J.M. Le Gouez** (Onera) : Introductory	
9h30-10h45	**Session 1 (Chair : C. Benoit (ONERA)**	
	M. Ravachol (Dassault-Aviation)	Visualization to support decision making : needs and challenges
	D. Caromel (INRIA, ActiveEon, Univ. Nice Sophia Antipolis)	CFD Workflows and Open Source CLOUD with OW2 ProActive : Renault OMD2 Use Cases
	L. Reimer (DLR)	Multidisciplinary Analysis Workflow with the FlowSimulator
10h45-11h05	*Coffee break*	
11h05-12h45	**Session 1** (*Cont.*)	
	D. Snyder (CD-adapco)	STAR-CCM+ : A New Approach to Numerical Simulation
	V. Morgenthaler (ANSYS France)	Simulation Driven Product Development with ANSYS Workbench platform
	C. Hirsch (Numeca)	Components for an Integrated CFD Workflow for large scale Multidisciplinary Simulations
	C. Geuzaine (Univ. Liège)	Recent Advances in Quad Meshing
12h45-13h45	*Lunch*	

1.3 Agenda, october 3, 2012, afternoon

12h45-13h45	*Lunch*	
13h45-15h50	**Session 2 (Chair : J.C. Weill (CEA)**	
	P. Brenner (ASTRIUM ST)	Recent developments about overlapping grids for unstructured meshes
	S. Péron, C. Benoit, P. Raud (ONERA)	Cassiopée : pre- and post-processing for CFD Python CGNS workflow
	M. Poinot (ONERA)	Numerical Simulation Components in an Open Python Environment
	S. Deck, P.E. Weiss, R. Pain (ONERA)	Some reflections on massive post-processing of large unsteady flow simulation data sets
	K. Hillewaert (Cenaero)	New challenges and opportunities created by high order discretization schemes for industrial flows
15h50-16h10	*Coffee break*	
16h10-18h15	**Session 2** (*Cont.*)	
	Y. Fournier (EDF)	Evolving the Code_Saturne and NEPTUNE_CFD solver toolchains for billion-cell calculations
	V. Moureau (Coria)	Strategies for the massively parallel solving of reacting and two-phase flows with billion-cell meshes. A few casestudies with the YALES2 solver
	Y.M. Lefebvre (Intelligent Light)	CFD Workflow Improvements for Today and Tomorrow
	P. Sadlo (Univ. Stuttgart)	Advanced Techniques in Computational Flow Visualization
	P.F. Berte (ONERA)	Deploying and managing a visualization farm at Onera
18h15-18h30	**M. Ravachol (Systematic) : Closing**	



OSD October 3, 2012, Ph. d'Anfray (CEA, Aristote), M. Ravachol (Dassault-Aviation), T.H. Lê (ONERA)


OSD October 3, 2012, Thiên-Hiêp Lê (ONERA), Marie Tétard (Aristote)

Chapitre 2

Presentations

2.1 Opening

aristote — Nouvelles technologies de l'information et de la communication

OSD : Onera Scientific Day

CFD Workflow : Meshing, Solving, Visualizing, . . .

ONERA — THE FRENCH AEROSPACE LAB

SYSTEMATIC — PARIS REGION SYSTEMS & ICT CLUSTER

October 3, 2012

Aristote

What ? **Aristote** is a learned society in computer science and networking which brings together organizations and businesses interested in the latest developments and new uses of information technology (created in 1984 by INRIA, CEA, EDF & CNES).

How ? **ONERA** is also a long time member of **Aristote**, and scientific computing has always been a topic of interest within **Aristote** (working groups and seminars on *HPC, Grids, Cloud computing, GPU, etc.*)

Why ? Open **Onera Scientific Day(s)** to a larger community and new ideas.

aristote

2 octobre 2012 2 / 4

Agenda, october 3, 2012, morning

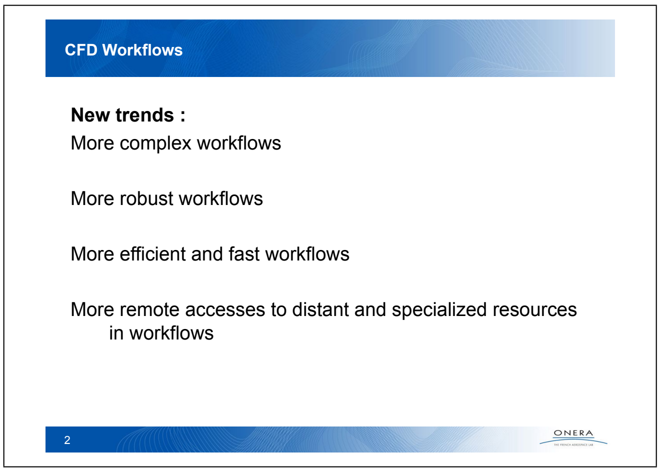
9h00-9h05	**Ph. d'Anfray** (Aristote) : Opening	
9h05-9h30	**J.M. Le Gouez** (Onera) : Introductory	
9h30-10h45	**Session 1 (Chair : C. Benoit (ONERA)**	
	M. Ravachol (Dassault-Aviation)	Visualization to support decision making : needs and challenges
	D. Caromel (INRIA, ActiveEon, Univ. Nice Sophia Antipolis)	CFD Workflows and Open Source CLOUD with OW2 ProActive : Renault OMD2 Use Cases
	L. Reimer (DLR)	Multidisciplinary Analysis Workflow with the FlowSimulator
10h45-11h05	*Coffee break*	
11h05-12h45	**Session 1** *(Cont.)*	
	D. Snyder (CD-adapco)	STAR-CCM+ : A New Approach to Numerical Simulation
	V. Morgenthaler (ANSYS France)	Simulation Driven Product Development with ANSYS Workbench platform
	C. Hirsch (Numeca)	Components for an Integrated CFD Workflow for large scale Multidisciplinary Simulations
	C. Geuzaine (Univ. Liège)	Recent Advances in Quad Meshing
12h45-13h45	*Lunch*	

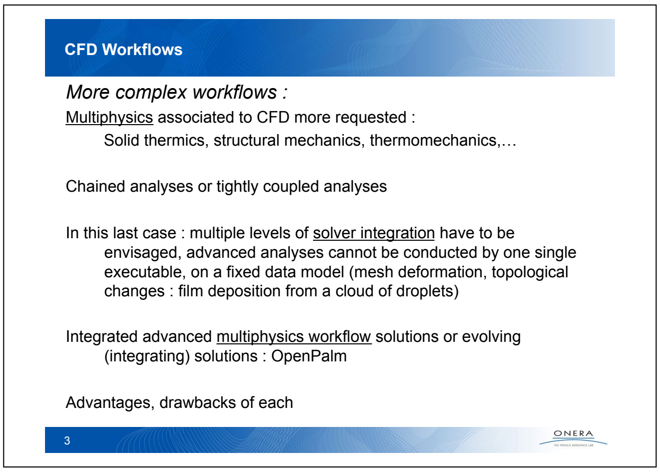
Agenda, october 3, 2012, afternoon

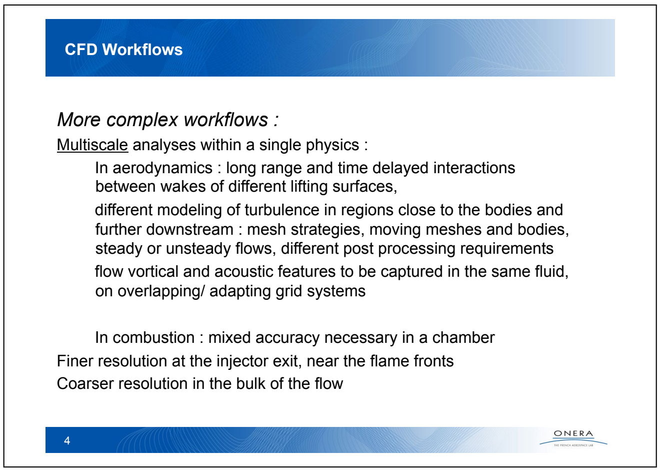
13h45-15h50	**Session 2 (Chair : J.C. Weill (CEA)**	
	P. Brenner (ASTRIUM ST)	Recent developments about overlapping grids for unstructured meshes
	S. Péron, C. Benoit, P. Raud (ONERA)	Cassiopée : pre- and post-processing for CFD Python CGNS workflow
	M. Poinot (ONERA)	Numerical Simulation Components in an Open Python Environment
	S. Deck, P.E. Weiss, R. Pain (ONERA)	Some reflections on massive post-processing of large unsteady flow simulation data sets
	K. Hillewaert (Cenaero)	New challenges and opportunities created by high order discretization schemes for industrial flows
15h50-16h10	*Coffee break*	
16h10-18h15	**Session 2** *(Cont.)*	
	Y. Fournier (EDF)	Evolving the Code_Saturne and NEPTUNE_CFD solver toolchains for billion-cell calculations
	V. Moureau (Coria)	Strategies for the massively parallel solving of reacting and two-phase flows with billion-cell meshes. A few casestudies with the YALES2 solver
	Y.M. Lefebvre (Intelligent Light)	CFD Workflow Improvements for Today and Tomorrow
	P. Sadlo (Univ. Stuttgart)	Advanced Techniques in Computational Flow Visualization
	P.F. Berte (ONERA)	Deploying and managing a visualization farm at Onera
18h15-18h30	**M. Ravachol (Systematic) : Closing**	

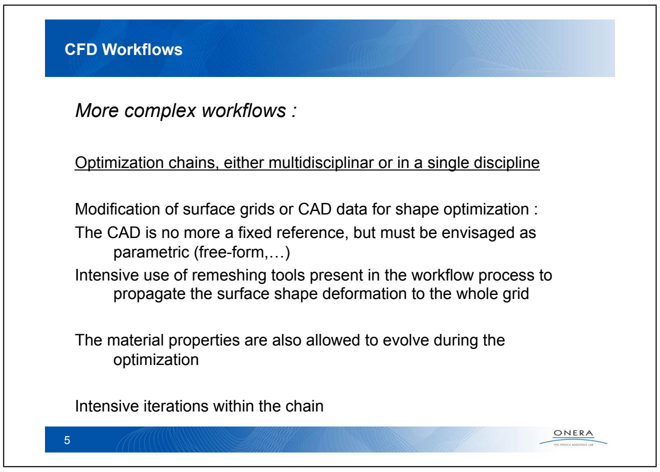
2.2 J.-M. Le Gouez (ONERA)

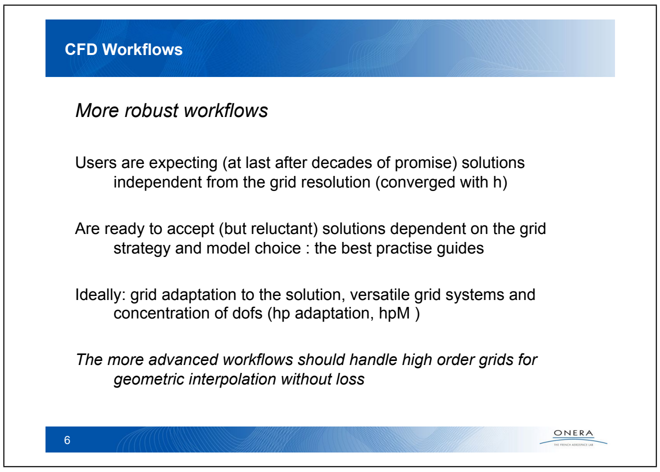
Introductory











CFD Workflows

More robust workflows

Relying on generic representation models
- data models : CAD, mesh, discrete flow and other fields,
- engineering representation : unit systems, reference frames, motions, integral objective functions
- workflow description : solver chain, dependencies between solver dependent data, coupling frequencies (macro, standard language)

Ex. GEM : Generic engineering model (Esprit IV) : STEP, Express modeling language, model parser

Capable of solver replacement for cross-validation, specialization

Usage for engineering studies by external consulting groups,…

7 ONERA

CFD Workflows

More efficient and fast workflows

Compatible with HPC :
- o very big data models with parallel I/O management,
- o discrete model subdivision,
- o optimization of data transfer and manipulation between 2 or more parallel solvers (example of the CWIPI interpolation library inside OpenPalm)
- o Providing post-processing on the fly for non storable fields, statistical processing in time and space, either by numbers or graphical

8 ONERA

CFD Workflows

More remote accesses to distant and specialized resources in workflows

Description of the processing resources available, optimization of the data model arrangement as function of the hardware architecture at hand (cloud computing from within the workflow manager ?)

Access to different levels of graphical resources : high-end workstations or clusters, office PCs

Need for a rich variety of software bricks, a common data model

9 ONERA

CFD Workflows Onera Scientific Day

Overview of the state of the art

Identification of new trends and on-going projects for novel developments,

Help for Onera in positioning its own strategy in between research and development, participating in enhancement / integration of productivity bricks and solvers in existing and promising solutions

10 ONERA

2.3 M. Ravachol (Dassault-Aviation)

Visualization to support decision making: needs and challenges

Michel Ravachol

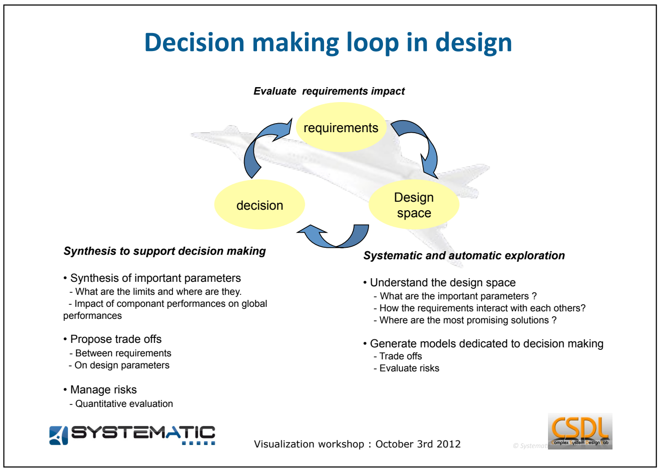
Dassault-Aviation

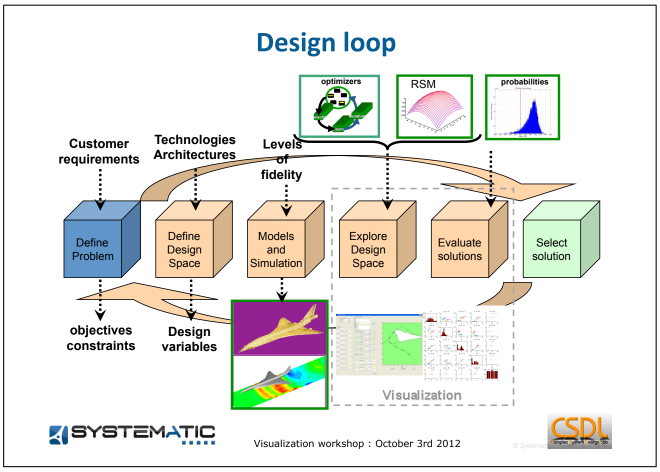
France

Abstract

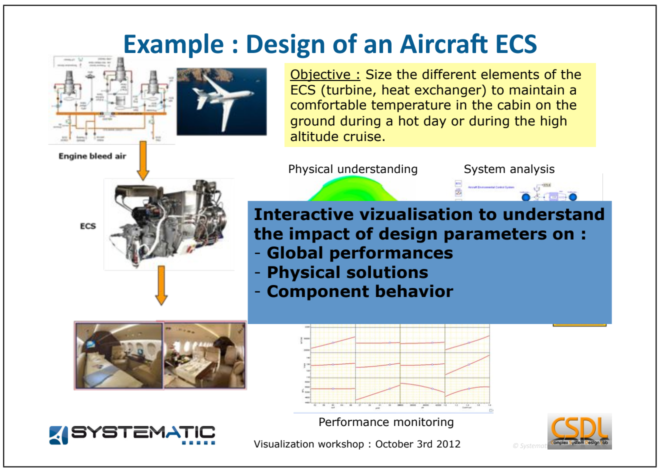
The design of complex systems generates a large amount of information that needs to be understood and synthesized in order to make decisions. The challenge is to immerse the decision makers in the decision space or more accurately within the space of compromise in order to enable them to better understand what they need by providing them with immediate answers to their questions. It is necessary to do this in a collaborative mode in order to ensure that all stakeholders can measure the impact of multiple interactions and be able to trace the analysis at the system level. To efficiently manage the trade-offs between breadth and depth this methodology must be used in an iterative process. Deciding each compromise at the system level allows one to focus future efforts on smaller areas but with an increase in the depth of details. We will present how visual analytics can be used and what are the challenges ahead.

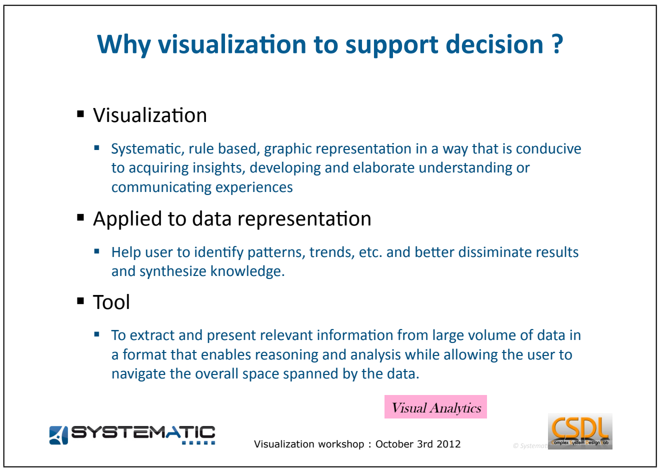

SYSTEMATIC
Visualization to support decision making:
Needs and Challenges
M. Ravachol, Dassault-Aviation
CSDL
Complex System Design Lab
© Systematic 2010

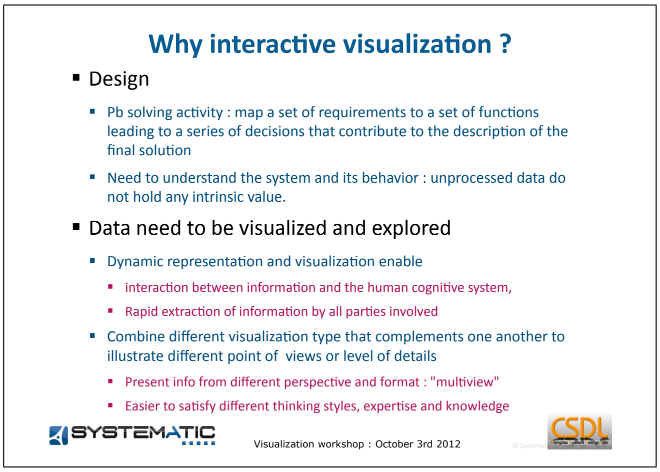

Decision making loop in design
Evaluate requirements impact
requirements
decision
Design space
Synthesis to support decision making
• Synthesis of important parameters
- What are the limits and where are they.
- Impact of componant performances on global performances
• Propose trade offs
- Between requirements
- On design parameters
• Manage risks
- Quantitative evaluation
Systematic and automatic exploration
• Understand the design space
- What are the important parameters ?
- How the requirements interact with each others?
- Where are the most promising solutions ?
• Generate models dedicated to decision making
- Trade offs
- Evaluate risks
SYSTEMATIC
Visualization workshop : October 3rd 2012

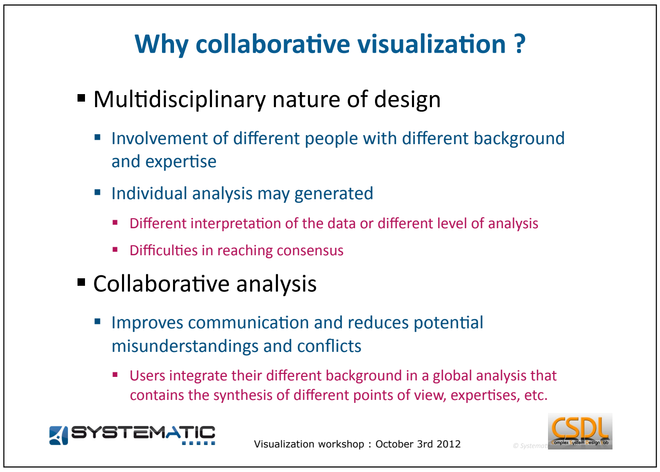

Design loop
optimizers
RSM
probabilities
Customer requirements
Technologies Architectures
Levels of fidelity
Define Problem
Define Design Space
Models and Simulation
Explore Design Space
Evaluate solutions
Select solution
objectives constraints
Design variables
Visualization
SYSTEMATIC
Visualization workshop : October 3rd 2012


CSDL project
Technical challenges :
Management of a hierarchy of interoperable surrogate models
Evaluate the robustess of results wrt risks and uncertainties
Exploration techniques adapted to the different level of fidelity of the models
Methogology to analyze the design process of complex systems
Develop interactive visualization tools to support decision making
Inspiration : Georgia Tech ASDL
Collaborative Visualization Environnement
SYSTEMATIC


Example : Design of an Aircraft ECS
Objective : Size the different elements of the ECS (turbine, heat exchanger) to maintain a comfortable temperature in the cabin on the ground during a hot day or during the high altitude cruise.
Engine bleed air
ECS
Physical understanding
System analysis
Interactive vizualisation to understand the impact of design parameters on :
- Global performances
- Physical solutions
- Component behavior
Performance monitoring
SYSTEMATIC
Visualization workshop : October 3rd 2012


Why visualization to support decision ?
Visualization
Systematic, rule based, graphic representation in a way that is conducive to acquiring insights, developing and elaborate understanding or communicating experiences
Applied to data representation
Help user to identify patterns, trends, etc. and better dissiminate results and synthesize knowledge.
Tool
To extract and present relevant information from large volume of data in a format that enables reasoning and analysis while allowing the user to navigate the overall space spanned by the data.
Visual Analytics
SYSTEMATIC
Visualization workshop : October 3rd 2012


Why interactive visualization ?
Design
Pb solving activity : map a set of requirements to a set of functions leading to a series of decisions that contribute to the description of the final solution
Need to understand the system and its behavior : unprocessed data do not hold any intrinsic value.
Data need to be visualized and explored
Dynamic representation and visualization enable
interaction between information and the human cognitive system,
Rapid extraction of information by all parties involved
Combine different visualization type that complements one another to illustrate different point of views or level of details
Present info from different perspective and format : "multiview"
Easier to satisfy different thinking styles, expertise and knowledge
SYSTEMATIC
Visualization workshop : October 3rd 2012

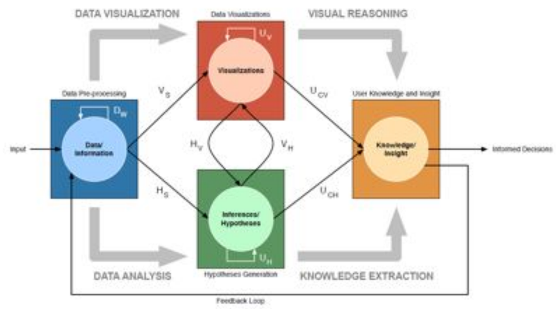

Why collaborative visualization ?
Multidisciplinary nature of design
Involvement of different people with different background and expertise
Individual analysis may generated
Different interpretation of the data or different level of analysis
Difficulties in reaching consensus
Collaborative analysis
Improves communication and reduces potential misunderstandings and conflicts
Users integrate their different background in a global analysis that contains the synthesis of different points of view, expertises, etc.
SYSTEMATIC
Visualization workshop : October 3rd 2012

Visual reasoning

- Visual data exploration as an hypothesis generation process
  - New hypothesis can be confirmed or not based on the information extracted from the data
    - Increase insights and better understanding of the problem
- Data exploration
  - Obtain new information of the data through visualization
    - Dynamically interact with visualization (connection of two or more views)
    - Decide to recompute new data and steer analysis in a different direction
    - …

SYSTEMATIC Visualization workshop : October 3rd 2012 CSDL

Visual reasoning process



From D. Mavris

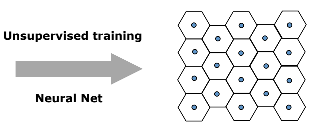
SYSTEMATIC Visualization workshop : October 3rd 2012 CSDL

Self Organizing Maps: an example of « advanced » data-processing

- Introduced by Teuvo Kohonen
  - Prototypes vectors are positionned on a regular low-dimensional grid in an ordered fashion by grouping similar data items together
  - ==> SOM are a powerful visualization tool

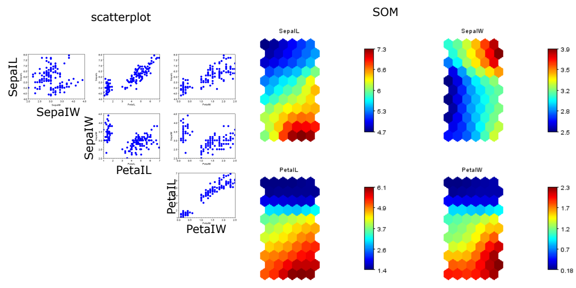
$$x_1^1, x_2^1, \cdots, x_d^1$$
$$\vdots$$
$$x_1^p, x_2^p, \cdots, x_d^p$$

Observations



SYSTEMATIC Visualization workshop : October 3rd 2012 CSDL

SOM Example



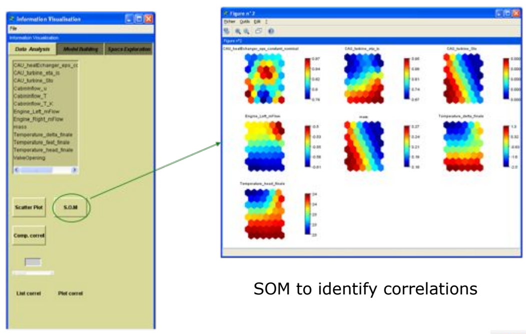
SYSTEMATIC Visualization workshop : October 3rd 2012 CSDL

Collaborative review of the Aircraft ECS

- Aircraft ECS design review
  - All the data necessary have been generated (CFD database, thermal database)
  - Collaborative review scenario
    - What are the influent parameters ?
    - What are the acceptable designs ?
    - What are the optimal design regions for the design performances ?
    - In an optimal region, what is the actual flow in the cabin ?

SYSTEMATIC Visualization workshop : October 3rd 2012 CSDL

Understanding the design space



SOM to identify correlations

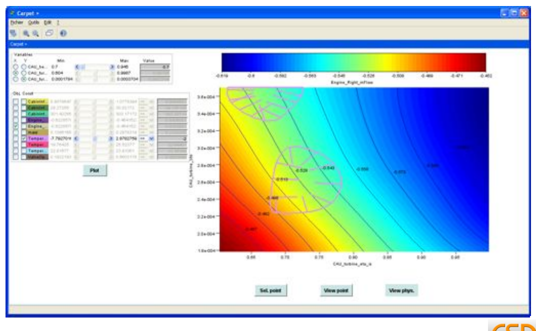
SYSTEMATIC Visualization workshop : October 3rd 2012 CSDL

Surrogate models for design space exploration

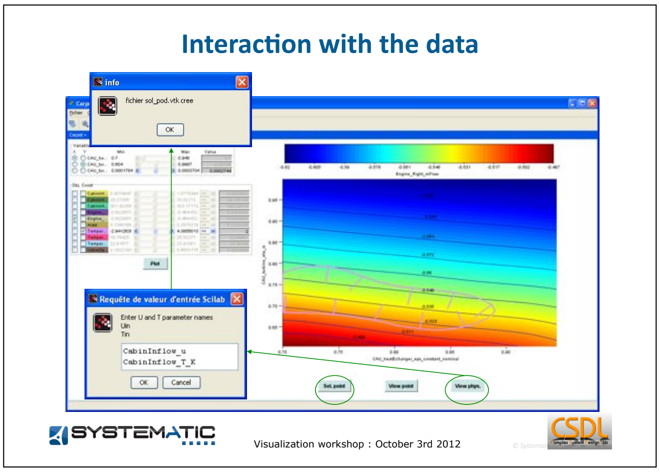
ANOVA to identify impacts of parameters and sensitivities of outputs

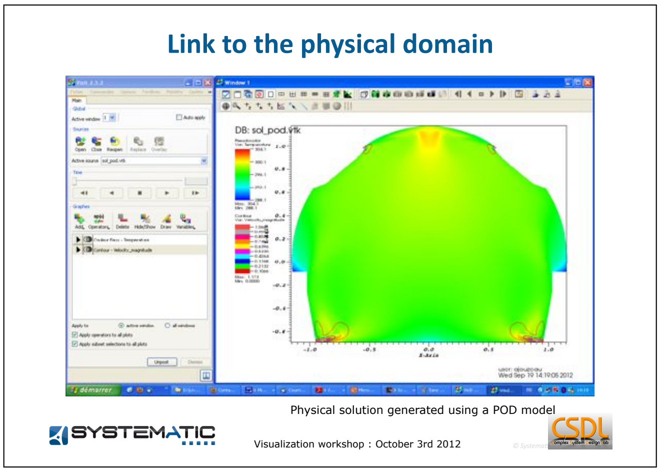
SYSTEMATIC Visualization workshop : October 3rd 2012 CSDL

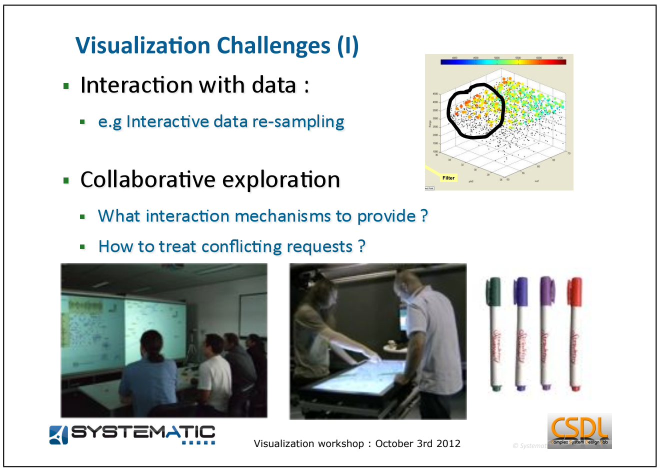
Feasible domain or objective function with constraints

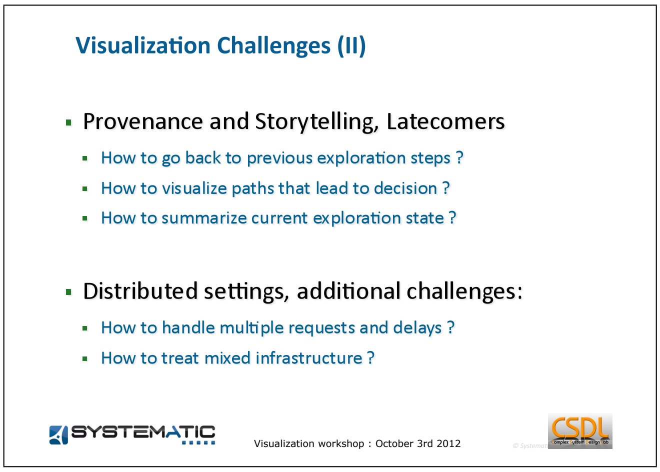


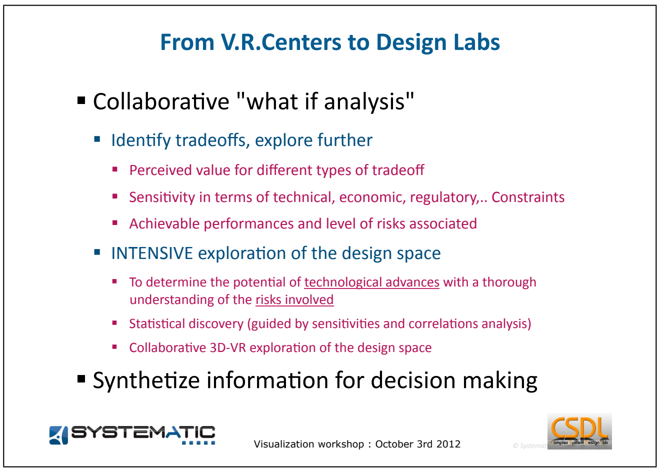
SYSTEMATIC Visualization workshop : October 3rd 2012 CSDL

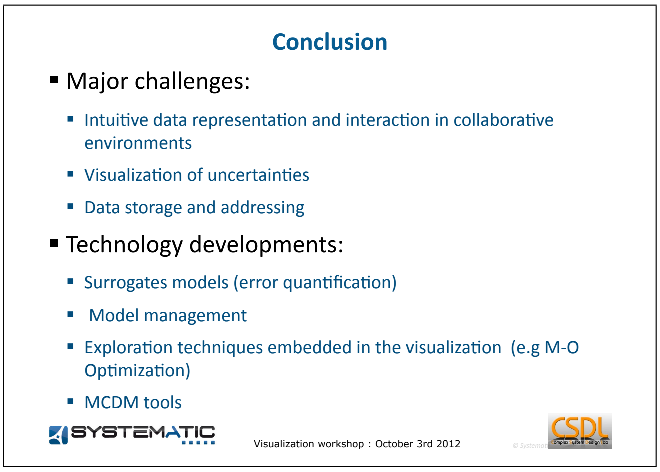

Interaction with the data
SYSTEMATIC
Visualization workshop : October 3rd 2012
CSDL

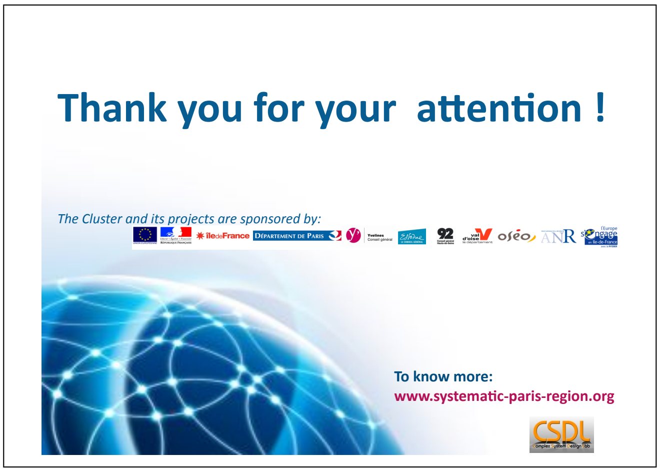

Link to the physical domain
DB: sol_pod.vtk
Physical solution generated using a POD model
SYSTEMATIC
Visualization workshop : October 3rd 2012
CSDL


Visualization Challenges (I)
Interaction with data :
e.g Interactive data re-sampling
Collaborative exploration
What interaction mechanisms to provide ?
How to treat conflicting requests ?
SYSTEMATIC
Visualization workshop : October 3rd 2012
CSDL


Visualization Challenges (II)
Provenance and Storytelling, Latecomers
How to go back to previous exploration steps ?
How to visualize paths that lead to decision ?
How to summarize current exploration state ?
Distributed settings, additional challenges:
How to handle multiple requests and delays ?
How to treat mixed infrastructure ?
SYSTEMATIC
Visualization workshop : October 3rd 2012
CSDL


From V.R.Centers to Design Labs
Collaborative "what if analysis"
Identify tradeoffs, explore further
Perceived value for different types of tradeoff
Sensitivity in terms of technical, economic, regulatory,.. Constraints
Achievable performances and level of risks associated
INTENSIVE exploration of the design space
To determine the potential of technological advances with a thorough understanding of the risks involved
Statistical discovery (guided by sensitivities and correlations analysis)
Collaborative 3D-VR exploration of the design space
Synthetize information for decision making
SYSTEMATIC
Visualization workshop : October 3rd 2012
CSDL


Conclusion
Major challenges:
Intuitive data representation and interaction in collaborative environments
Visualization of uncertainties
Data storage and addressing
Technology developments:
Surrogates models (error quantification)
Model management
Exploration techniques embedded in the visualization (e.g M-O Optimization)
MCDM tools
SYSTEMATIC
Visualization workshop : October 3rd 2012
CSDL


Thank you for your attention !
The Cluster and its projects are sponsored by:
To know more:
www.systematic-paris-region.org
CSDL

2.4 D. Caromel (INRIA, ActiveEon & Univ. Nice Sophia Antipolis)

CFD Workflows and Open Source CLOUD with OW2 ProActive: Renault OMD2 Use Cases

Denis Caromel

INRIA-Univ. Nice Sophia Antipolis-ActiveEon

Abstract

ProActive OW2 is an Open Source library for parallel, distributed, and concurrent computing, offering advanced HPC workflows integrated with Scheduling capacities. It also features management of heterogeneous private Clouds, with burst capacity on Data Center and Public clouds. Offering full accounting and security, ProActive handles multi-tenant Cloud, and a smooth path for application migration to the Cloud thanks to comprehensive interfaces (Graphical Studio, CLI, Java and REST, User and Admin Portals). Unique characteristics of ProActive are the capacity to manage both Virtual and Physical machines, to orchestrate native and virtualized applications.

The presentation will specifically give an overview of various methods for building and executing CFD Workflows, especially in the framework of the OMD2 project.

OMD2 is a collaborative R&D project dedicated to large scale multidisciplinary optimizations, especially in the context of the Car Manufacturing. Lead by RENAULT, the project includes the companies CD-adapco, SIREHNA, ACTIVEEON, and the academics INRIA, ENSM-SE, UTC, ECP, IRCCyN, ENS CACHAN, together with the DIGITEO consortium. The talk will also feature a few related Use Cases: CPU/GPU workflows, Map/Reduce applications, Load Injection and continuous integration.

We will show live demonstrations of use cases on a Scientific Grid and Cloud platform of 1200 cores, 30TB storage, and 480 CUDA Cores http://proactive.inria.fr/pacagrid/pacagrid-cluster

Short biography

Denis Caromel, professor at University of Nice-Sophia Antipolis, and member of the INRIA-Univ. Nice-CNRS OASIS team, is also founder and CEO of the INRIA startup ActiveEon. His research interests include parallel, concurrent, distributed, and Cloud computing.

Denis Caromel gave many invited talks on Object, Parallel and Distributed Computing around the world (Jet Propulsion Laboratory, Berkeley, Stanford, ISI, USC, Electrotechnical Laboratory Tsukuba, Sydney, Oracle-BEA EMEA, Digital System Research Center in Palo Alto, NASA Langley, IBM Tom Watson and Zurich). He acted as keynote speaker at several major conferences (MDM, DAPSYS 2008, CGW'08, Shanghai CCGrid 2009, IEEE ICCP'09, ICPADS

2009 in Hong Kong). Recently, he gave two important invited talks at Sun Microsystems HPC Consortium (Austin, Tx), and at Devoxx (gathering about 3500 persons), and an invited conference at Expo Universal 2010, Oct. 18, Shanghai, China. http://www-sop.inria.fr/oasis/caromel/


CFD Workflows + OS CLOUD with OW2 ProActive
Denis Caromel
Université Nice Sophia Antipolis
Inria
ActiveEon SCALE BEYOND LIMITS
Agenda
1. Background: INRIA, ActiveEon
2. ProActive Parallel Suite
3. Use Cases & Demos
Renault OMD2 Use Cases
Workflow
Acceleration -- Parallelism
Scheduling, Resource & Data Management, Deployment
ProActive Parallel Suite
OW2 Consortium
1


ProActive Parallel Suite
1. Background
2
ProActive Parallel Suite
2

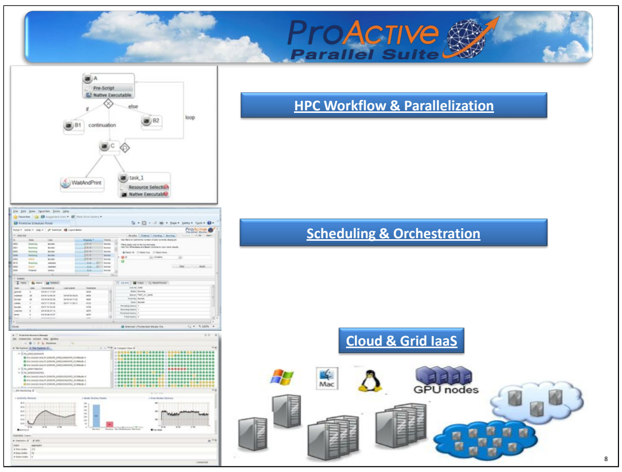

OASIS Team
Inria
Université Nice Sophia Antipolis
cnrs
Oasis
ProActive Parallel Suite was started in the team
ProActive Parallel Suite
3

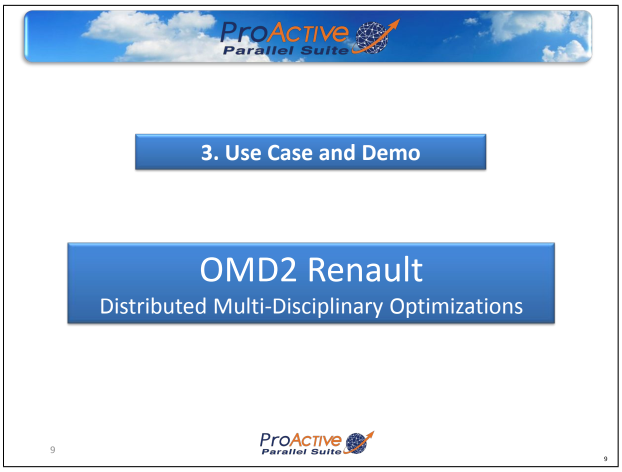

Technology Transfer in 2007
ActiveEon SCALE BEYOND LIMITS
Located in Sophia Antipolis, between
Nice and Cannes, France
Visitors Welcome!
4

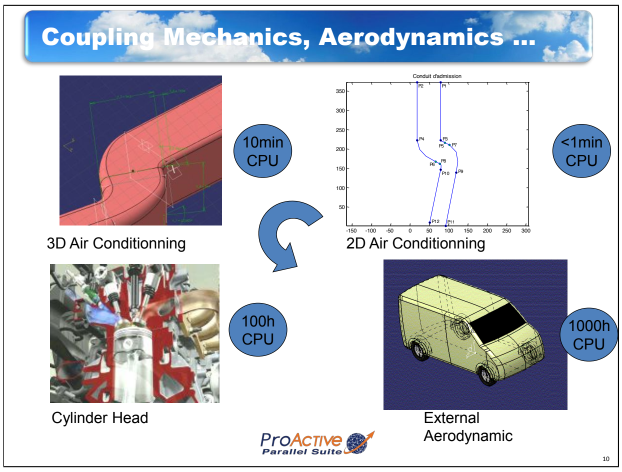

ActiveEon Overview
ActiveEon SCALE BEYOND LIMITS
ActiveEon, a software company born of INRIA, founded in 2007, HQ in the French scientific park Sophia Antipolis
Developping, with INRIA contributions, ProActive Parallel Suite®, a Professional Open Source middleware for parallel, distributed, multi-core computing
Core mission: Scale Beyond Limits
Providing a full range of services for ProActive Parallel Suite
Worldwide production customers and users:
DELL
Bull
hp
cea
orange
ifp Energies nouvelles
RENAULT
eDF
GMO
DEXIA
INEO
HUAWEI
INRA
THALES
sirehna
mediametrie
AMADEUS
EMBL-EBI
ProActive Parallel Suite
5

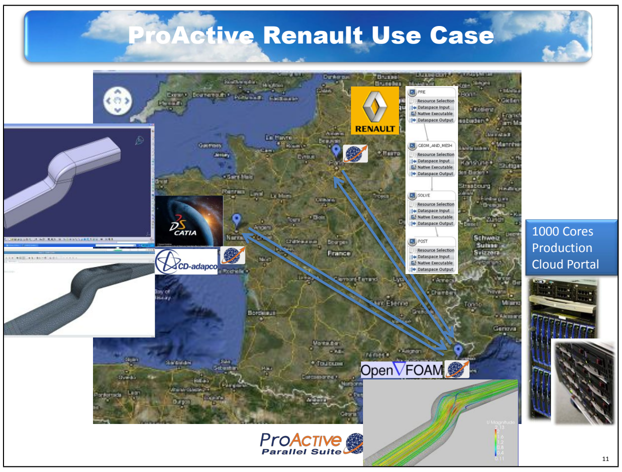

A Wide Range of Services
Training and Certification
Accelerate learning process
Consulting
Optimize your infrastructure and maximize ROI
Technical Support - Subscription
The guarantee of a quick and efficient assistance
Integration-Development
Get ActiveEon's products fine tuned to your specific needs
Partnerships
With OEMs and ISVs
ProActive Parallel Suite
6

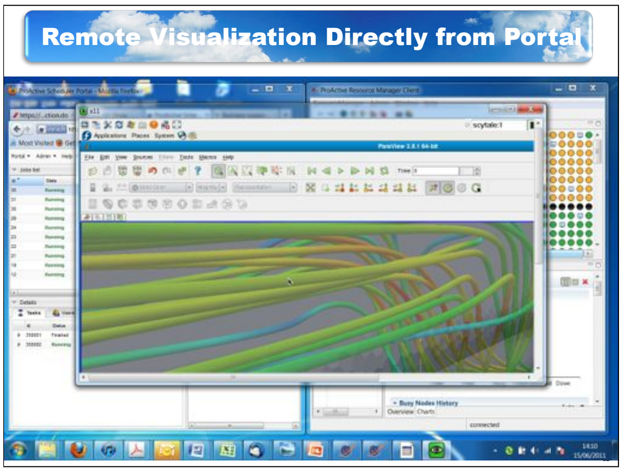

ProActive Parallel Suite
2. ProActive Parallel Suite
7
ProActive Parallel Suite
7

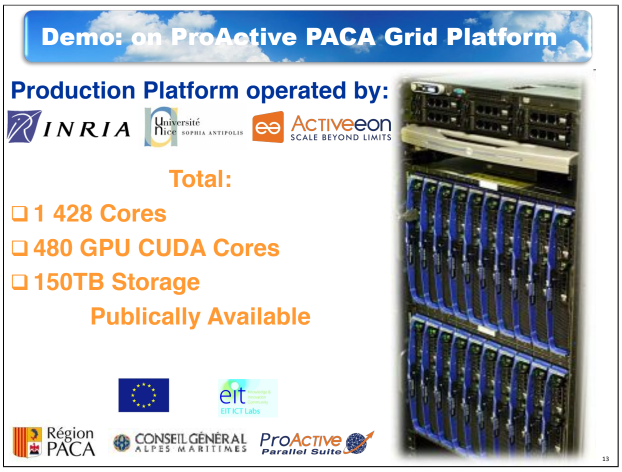

ProActive Parallel Suite
HPC Workflow & Parallelization
Scheduling & Orchestration
Cloud & Grid IaaS
Mac
GPU nodes
8

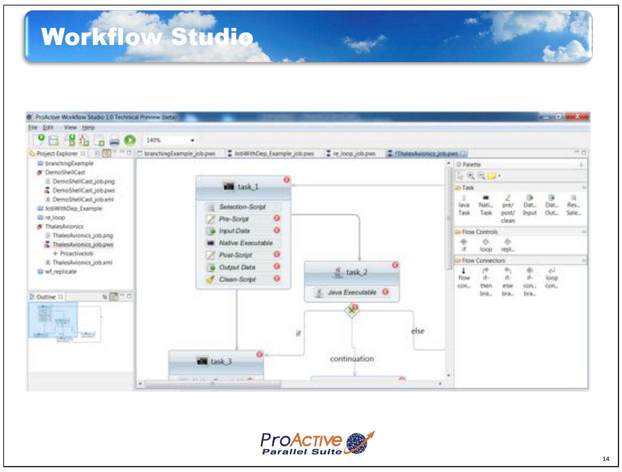

ProActive
Parallel Suite
3. Use Case and Demo
OMD2 Renault
Distributed Multi-Disciplinary Optimizations
9

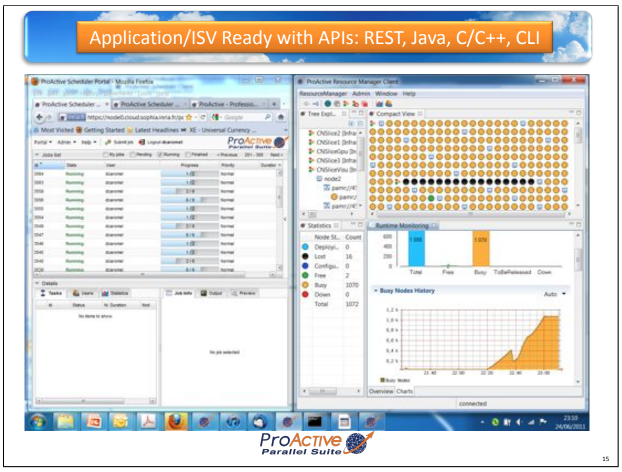

Coupling Mechanics, Aerodynamics ...
10min CPU
<1min CPU
3D Air Conditionning
2D Air Conditionning
100h CPU
1000h CPU
Cylinder Head
External Aerodynamic
10

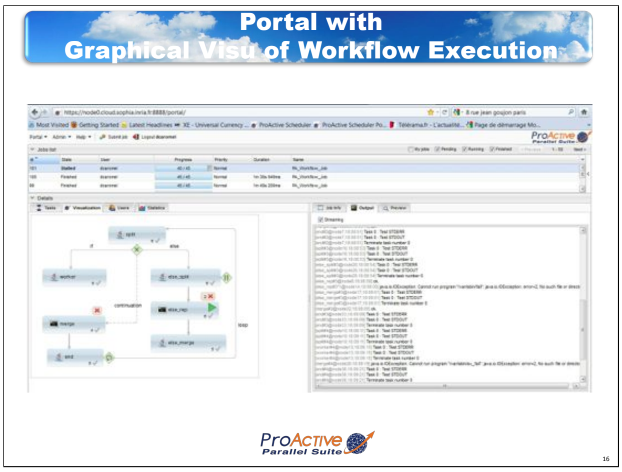

ProActive Renault Use Case
RENAULT
1000 Cores
Production
Cloud Portal
CD-adapco
OpenFOAM
11

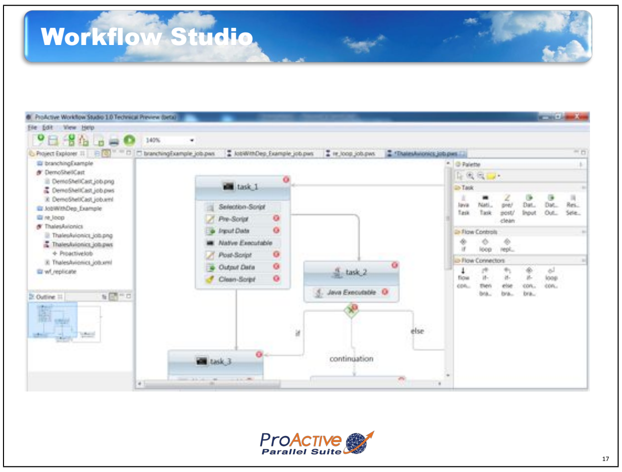

Remote Visualization Directly from Portal

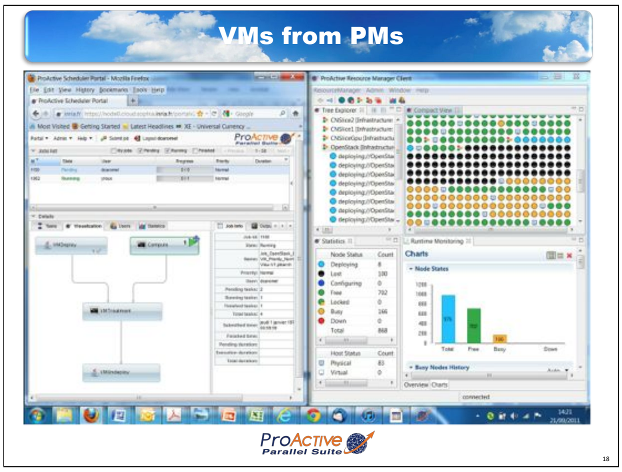

Demo: on ProActive PACA Grid Platform
Production Platform operated by:
INRIA
Université Nice Sophia Antipolis
ActiveEon
SCALE BEYOND LIMITS
Total:
1 428 Cores
480 GPU CUDA Cores
150TB Storage
Publically Available
eit
Région PACA
CONSEIL GÉNÉRAL ALPES MARITIMES
13

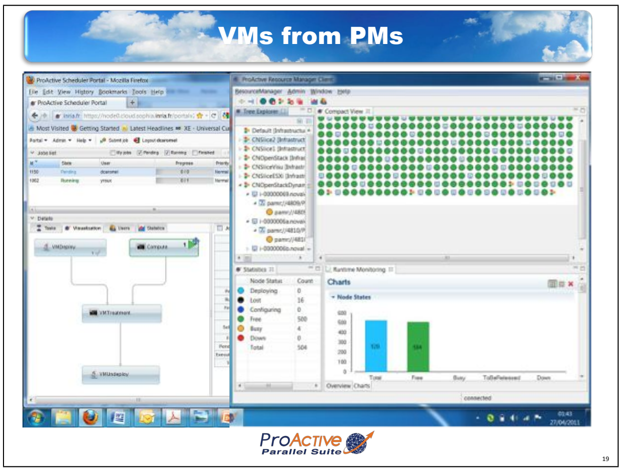

Workflow Studio
14

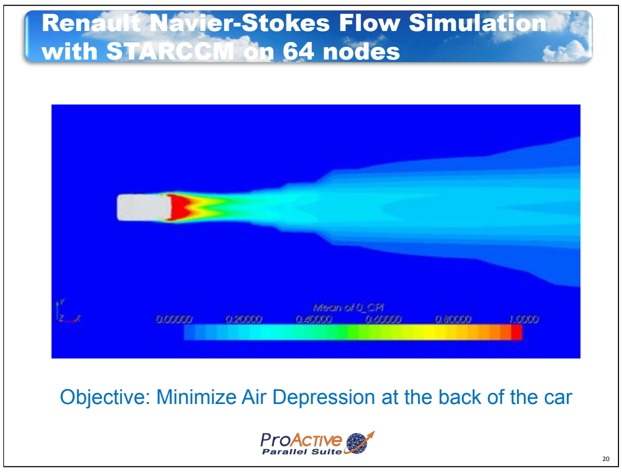

Application/ISV Ready with APIs: REST, Java, C/C++, CLI
15

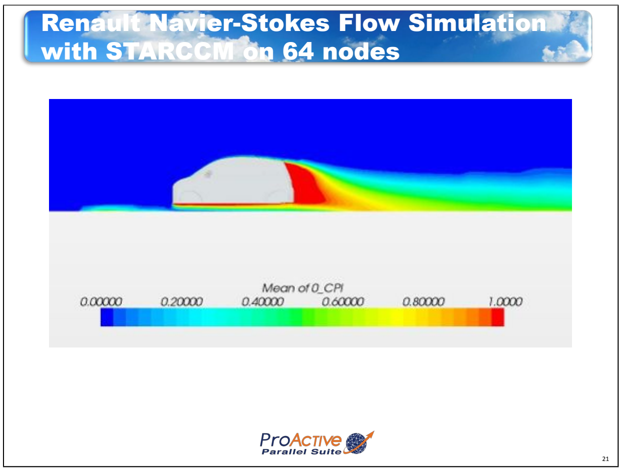

Portal with
Graphical Visu of Workflow Execution
16

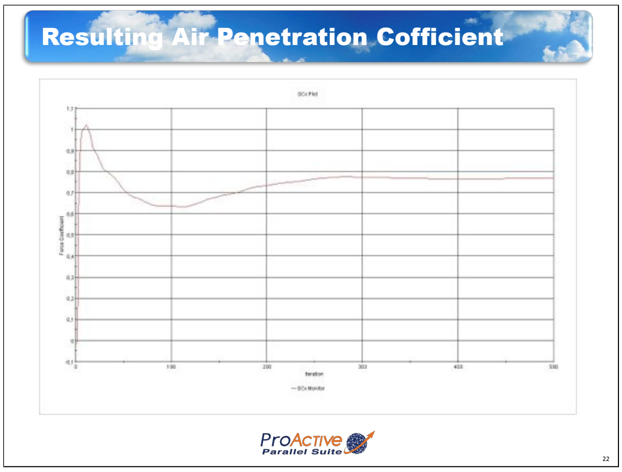

Workflow Studio
ProActive Parallel Suite
17

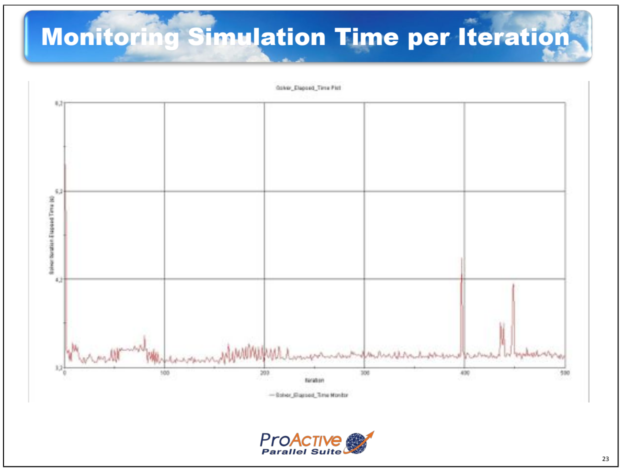

VMs from PMs
ProActive Parallel Suite
18


VMs from PMs
ProActive Parallel Suite
19

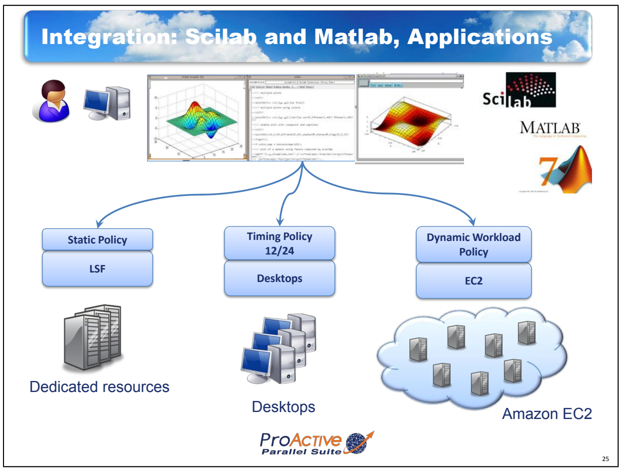

Renault Navier-Stokes Flow Simulation with STARCCM on 64 nodes
Objective: Minimize Air Depression at the back of the car
ProActive Parallel Suite
20

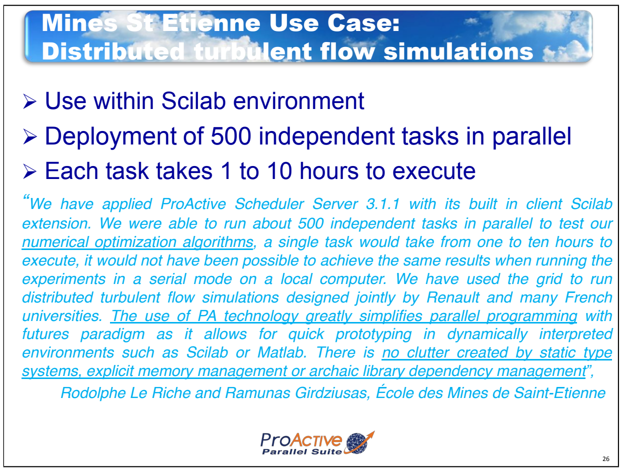

Renault Navier-Stokes Flow Simulation with STARCCM on 64 nodes
Mean of 0_CPI
0.00000
0.20000
0.40000
0.60000
0.80000
1.0000
ProActive Parallel Suite
21


Resulting Air Penetration Cofficient
ProActive Parallel Suite
22

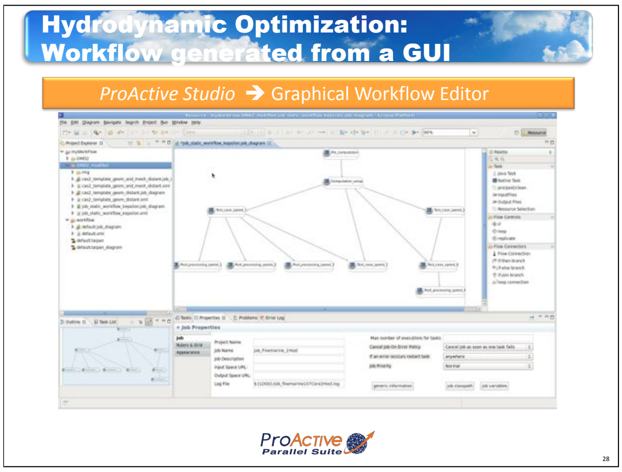

Monitoring Simulation Time per Iteration
ProActive Parallel Suite
23

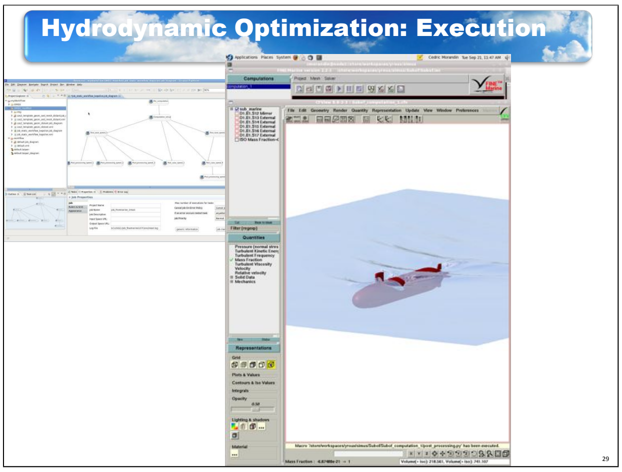

Integration with Applications
24
ProActive Parallel Suite
24

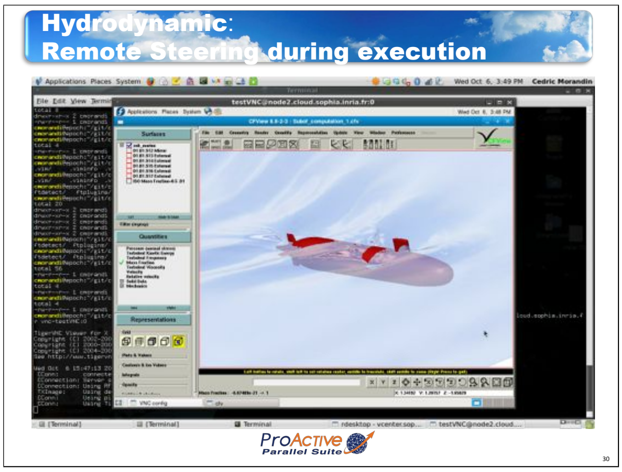

Integration: Scilab and Matlab, Applications
Scilab
MATLAB
Static Policy
LSF
Timing Policy
12/24
Desktops
Dynamic Workload
Policy
EC2
Dedicated resources
Desktops
Amazon EC2
ProActive
Parallel Suite
25


Mines St Etienne Use Case:
Distributed turbulent flow simulations
Use within Scilab environment
Deployment of 500 independent tasks in parallel
Each task takes 1 to 10 hours to execute
"We have applied ProActive Scheduler Server 3.1.1 with its built in client Scilab extension. We were able to run about 500 independent tasks in parallel to test our numerical optimization algorithms, a single task would take from one to ten hours to execute, it would not have been possible to achieve the same results when running the experiments in a serial mode on a local computer. We have used the grid to run distributed turbulent flow simulations designed jointly by Renault and many French universities. The use of PA technology greatly simplifies parallel programming with futures paradigm as it allows for quick prototyping in dynamically interpreted environments such as Scilab or Matlab. There is no clutter created by static type systems, explicit memory management or archaic library dependency management",
Rodolphe Le Riche and Ramunas Girdziuskas, École des Mines de Saint-Etienne
ProActive
Parallel Suite
26

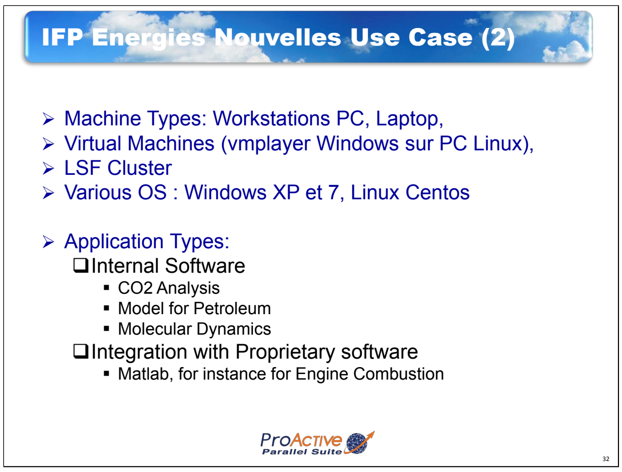

Use Case:
Hydrodynamic
with
K-Epsilon and FineMarine
K-EPSILON
HYDRODYNAMIC & AERODYNAMIC
ActiveEon
SCALE BEYOND LIMITS
ProActive
Parallel Suite
OW2
Consortium
27


Hydrodynamic Optimization:
Workflow generated from a GUI
ProActive Studio → Graphical Workflow Editor
ProActive
Parallel Suite
28


Hydrodynamic Optimization: Execution
29


Hydrodynamic:
Remote Steering during execution
ProActive
Parallel Suite
30


IFP Energies Nouvelles
Production User (Press Release)
31
ProActive
Parallel Suite
31


IFP Energies Nouvelles Use Case (2)
Machine Types: Workstations PC, Laptop,
Virtual Machines (vmplayer Windows sur PC Linux),
LSF Cluster
Various OS : Windows XP et 7, Linux Centos
Application Types:
Internal Software
CO2 Analysis
Model for Petroleum
Molecular Dynamics
Integration with Proprietary software
Matlab, for instance for Engine Combustion
ProActive
Parallel Suite
32

IFP Energies Nouvelles Use Case

- Deployment in production for all sites of IFP EN
- On over 600 computers
- Demanding applications: business workflows, numerical and financial simulations, Matlab ™ and Scilab, data analysis (Map / Reduce)
- Web-based portal as well as
  RCP and
  APIs : from within Application

ProActive Parallel Suite 33

IFP Energies Nouvelles Use Case

- Deployment in production for all sites of IFP EN
- On over 600 computers
- Demanding applications: business workflows, numerical and financial simulations, Matlab ™ and Scilab, data analysis (Map / Reduce)
- Web-based portal as well as RCP and APIs

"With the adoption of ProActive […] IFP EN enters in the era of Cloud Computing. We are going to cut on our hardware and software costs, to strengthen our business workflows, to use these globalized resources directly in our business software in order to accelerate them", Frédéric Gauluet, IFP EN

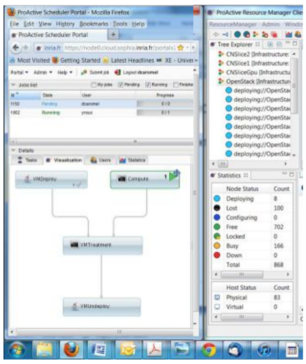
ProActive Parallel Suite 34

ProActive Parallel Suite

4. Conclusion

35 ProActive Parallel Suite 35

ProActive Parallel Suite Open the Missing Links



- Application Acceleration, Workflows: Script, GUI
- Advanced Scheduling and Mapping
  - ➔Resource Control
  - ➔Core, CPU, Host
- Heterogeneity Support
  - ➔Physical Machines: Linux, Windows, Mac
  - ➔Desktop, Clusters, Grids, Clouds

ProActive Parallel Suite 36

Request from CFD experts

- From Jean-Marie Le Gouez:
  - Use of various resources (Cluster, … Desktop)
  - Description of resources
  - Use of GPUs
  - On the fly post-processing (No intermediate storage), for Model Coupling and Visu.
- From Michel Ravachol:
  - Collaborative Vizualisation

ProActive Parallel Suite 37

ProActive Parallel Suite

Thank You!

Extra Material and Use Cases Below.

38 ProActive Parallel Suite 38

2.5 L. Reimer (DLR)

Multidisciplinary Analysis Workflow with the FlowSimulator

Lars Reimer

Deutsches Zentrum für Luft- und Raumfahrt e.V. (DLR)
German Aerospace Center
Institute of Aerodynamics and Flow Technology | C²A²S²E

Lilienthalplatz 7 | 38108 Braunschweig | Germany

Abstract

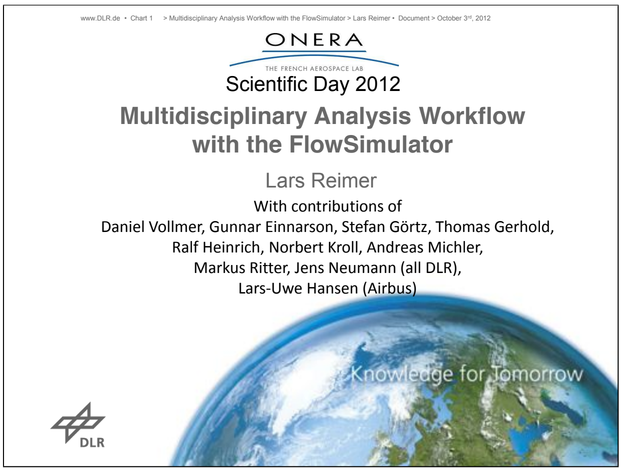
To improve the prediction accuracy of the aerodynamic performance and safety margins of aircraft one strives for enhancing existing CFD analysis workflows towards multi-disciplinary analysis workflows, i.e. one aims at analyzing directly flow-structure coupled or even flow-structure-flight mechanics coupled problems. In such analysis, the CFD computation forms only one constituent of the whole process chain. Mostly, the codes that are used in the process chain and that are highly-specialized in solving a particular discipline go back to independent developments. In many cases their development took not into account the option for coupling with other code. As a result, the necessary communication between involved codes often is only possible via file IO. But file IO is actually to be avoided or at least minimized in targeted massively-parallel high-performance computations. Moreover, most of conventionally designed process chains characterized by direct data exchange between process chain constituents are inflexible and lack expandability.

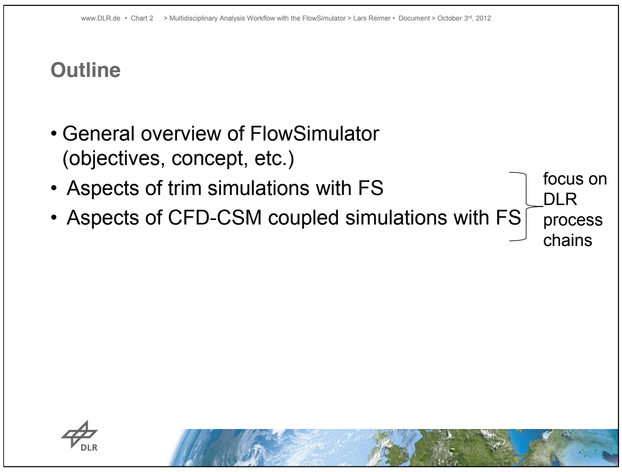
In order to remedy the situation described before, the FlowSimulator framework was developed in a joint project of Airbus, Cassidian, ONERA, DLR, universities and others. FlowSimulator aims to form a unique interface and environment for the assembly and the massively-parallel execution of multi-disciplinary process chains. According to the concept of FlowSimulator the mono-disciplinary codes involved in the process chain do not exchange data with each other, but efficiently in-memory with the FlowSimulator data manager. By doing so, individual simulation codes of the process chain can easily be replaced by others, for instance a structured flow solver by an unstructured one or vice versa. The FlowSimulator data manager as C++ core component of the FlowSimulator framework provides data containers for the parallel storage of mesh objects and the ability to execute operations on these mesh objects in parallel, such as mesh import/export from/into various data format, mesh partitioning, mesh transformation, mesh deformation, mesh extractions, mesh-to-mesh interpolation, etc. In the same way, data containers exist with corresponding functionality for geometry objects which are relevant for mesh generation and shape optimization.

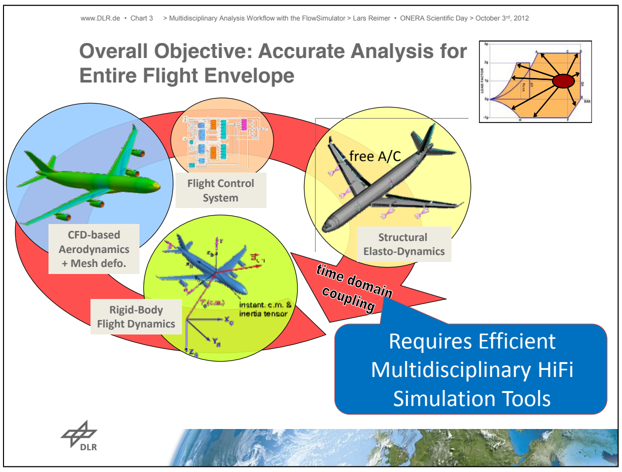
Users can easily access the functionality provided by the FlowSimulator data manager as well as FlowSimulator-interfaced applications on a Python script level. This way the user can script complex parallely executable process chains that cover a

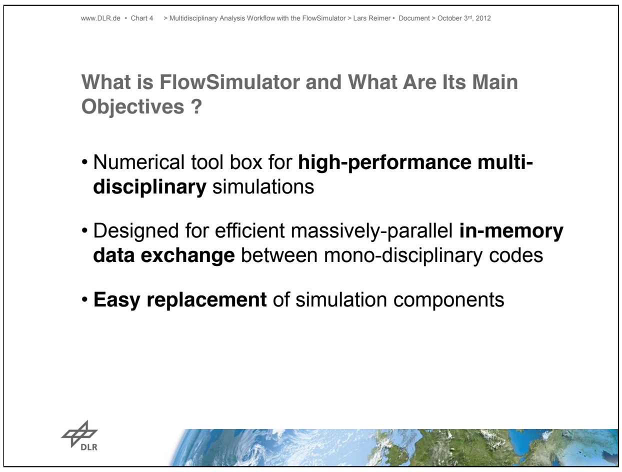
complete multi-disciplinary analyis workflow comprising CFD and CSM mesh generation, the actual simulation in combination with in-situ visualization, and postprocessing actions.

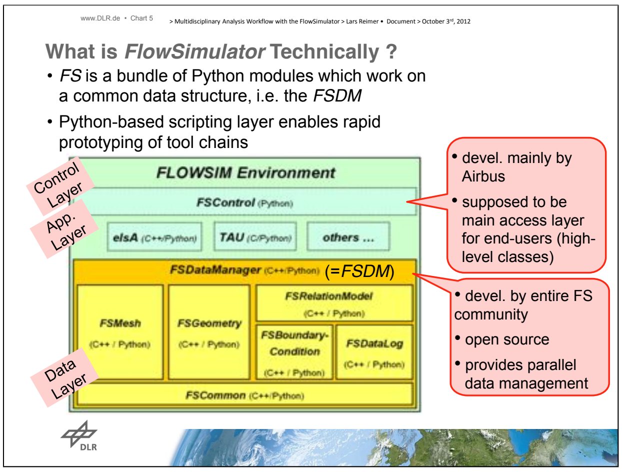
The talk shows the general concept of the FlowSimulator framework, points out essential elements of the FlowSimulator from DLR's point of view which have particular relevance for multidisciplinary applications, e.g. for flow-structure coupling, and highlights features of the FlowSimulator in a number of sample cases and industrially relevant use cases. FlowSimulator developments the DLR is currently working on and plans to work on in the future are outlined.

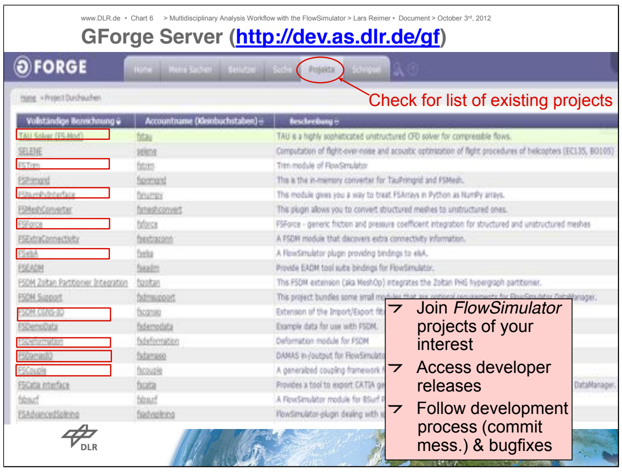

www.DLR.de • Chart 1 > Multidisciplinary Analysis Workflow with the FlowSimulator > Lars Reimer • Document > October 3rd, 2012
ONERA
THE FRENCH AEROSPACE LAB
Scientific Day 2012
Multidisciplinary Analysis Workflow with the FlowSimulator
Lars Reimer
With contributions of
Daniel Vollmer, Gunnar Einnarson, Stefan Görtz, Thomas Gerhold, Ralf Heinrich, Norbert Kroll, Andreas Michler, Markus Ritter, Jens Neumann (all DLR), Lars-Uwe Hansen (Airbus)
Knowledge for Tomorrow
DLR

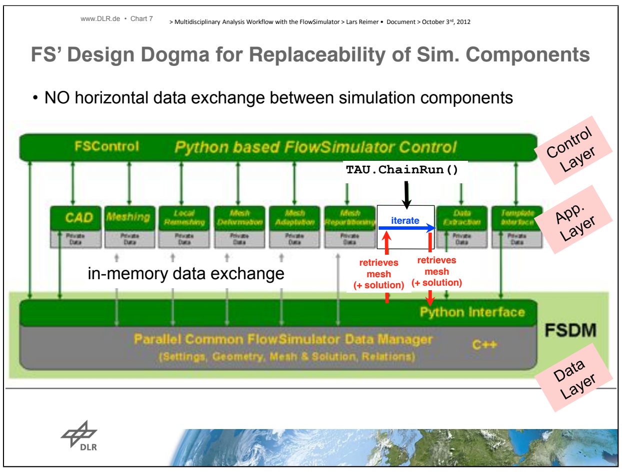

www.DLR.de • Chart 2 > Multidisciplinary Analysis Workflow with the FlowSimulator > Lars Reimer • Document > October 3rd, 2012
Outline
• General overview of FlowSimulator (objectives, concept, etc.)
• Aspects of trim simulations with FS
• Aspects of CFD-CSM coupled simulations with FS
focus on DLR process chains
DLR

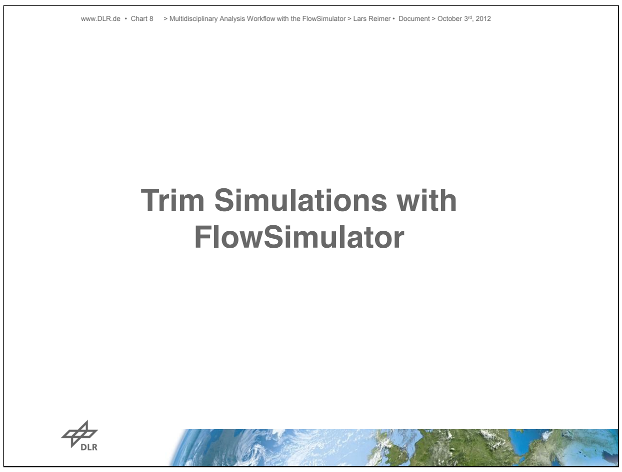

www.DLR.de • Chart 3 > Multidisciplinary Analysis Workflow with the FlowSimulator > Lars Reimer • ONERA Scientific Day > October 3rd, 2012
Overall Objective: Accurate Analysis for Entire Flight Envelope
Flight Control System
free A/C
CFD-based Aerodynamics + Mesh defo.
Structural Elasto-Dynamics
time domain coupling
Rigid-Body Flight Dynamics
instant. c.m. & inertia tensor
Requires Efficient Multidisciplinary HiFi Simulation Tools
DLR

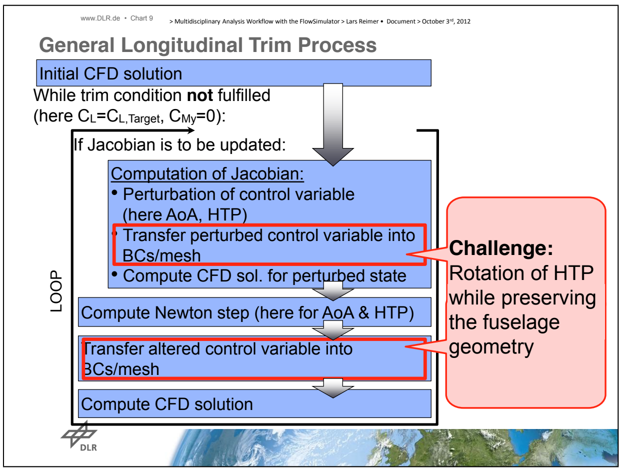

www.DLR.de • Chart 4 > Multidisciplinary Analysis Workflow with the FlowSimulator > Lars Reimer • Document > October 3rd, 2012
What is FlowSimulator and What Are Its Main Objectives ?
• Numerical tool box for high-performance multi-disciplinary simulations
• Designed for efficient massively-parallel in-memory data exchange between mono-disciplinary codes
• Easy replacement of simulation components
DLR

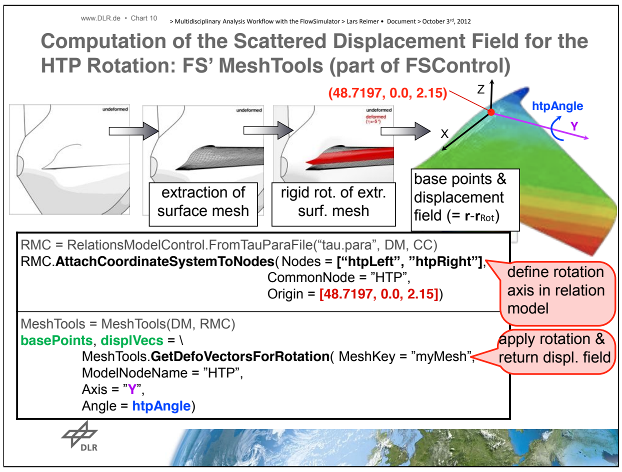

www.DLR.de • Chart 5 > Multidisciplinary Analysis Workflow with the FlowSimulator > Lars Reimer • Document > October 3rd, 2012
What is FlowSimulator Technically ?
• FS is a bundle of Python modules which work on a common data structure, i.e. the FSDM
• Python-based scripting layer enables rapid prototyping of tool chains
Control Layer
App. Layer
Data Layer
FLOWSIM Environment
FSControl (Python)
elsA (C++/Python)
TAU (C/Python)
others ...
FSDataManager (C++/Python) (=FSDM)
FSRelationModel (C++ / Python)
FSMesh (C++ / Python)
FSGeometry (C++ / Python)
FSBoundary-Condition (C++ / Python)
FSDataLog (C++ / Python)
FSCommon (C++/Python)
• devel. mainly by Airbus
• supposed to be main access layer for end-users (high-level classes)
• devel. by entire FS community
• open source
• provides parallel data management
DLR

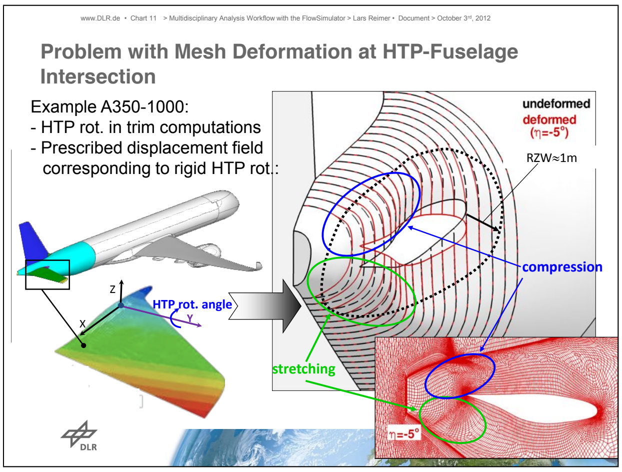

www.DLR.de • Chart 6 > Multidisciplinary Analysis Workflow with the FlowSimulator > Lars Reimer • Document > October 3rd, 2012
GForge Server (http://dev.as.dlr.de/gf)
FORGE
Check for list of existing projects
Join FlowSimulator projects of your interest
Access developer releases
Follow development process (commit mess.) & bugfixes
DLR

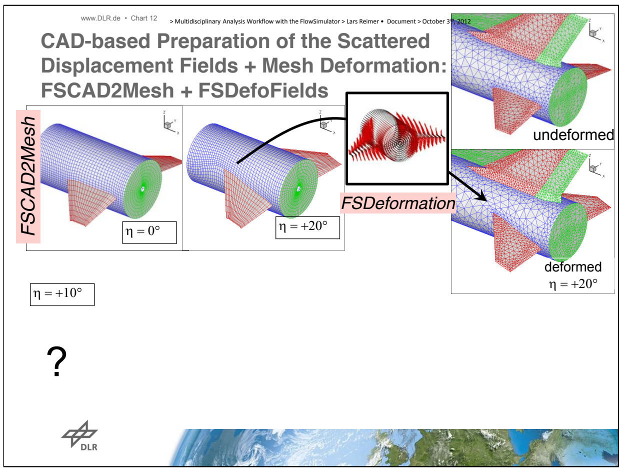

www.DLR.de • Chart 7 > Multidisciplinary Analysis Workflow with the FlowSimulator > Lars Reimer • Document > October 3rd, 2012
FS' Design Dogma for Replaceability of Sim. Components
• NO horizontal data exchange between simulation components
FSControl
Python based FlowSimulator Control
TAU.ChainRun()
Control Layer
App. Layer
CAD
Meshing
iterate
in-memory data exchange
retrieves mesh (+ solution)
retrieves mesh (+ solution)
Python Interface
FSDM
Parallel Common FlowSimulator Data Manager
C++
(Settings, Geometry, Mesh & Solution, Relations)
Data Layer
DLR

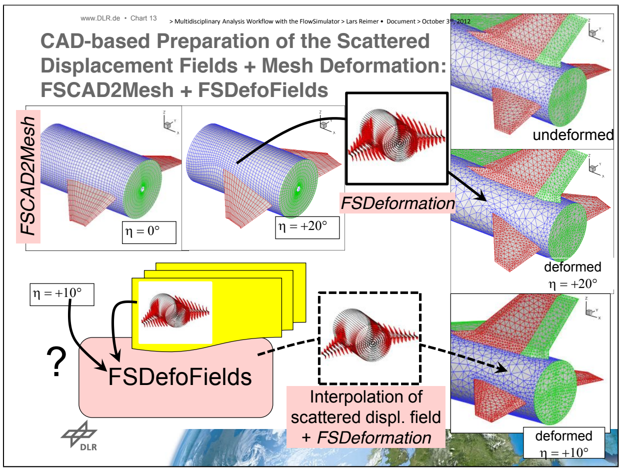

www.DLR.de • Chart 8 > Multidisciplinary Analysis Workflow with the FlowSimulator > Lars Reimer • Document > October 3rd, 2012
Trim Simulations with FlowSimulator
DLR

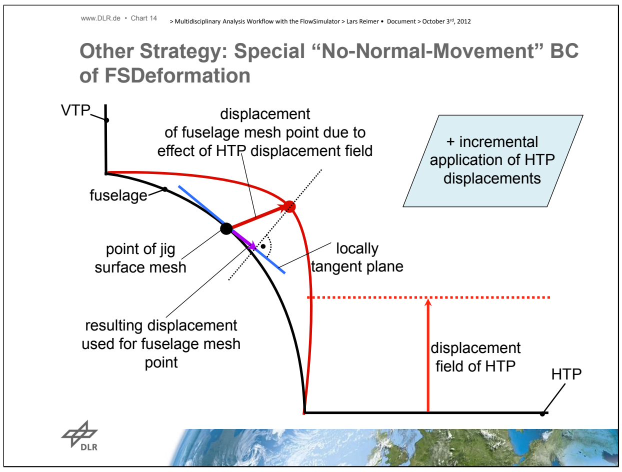

General Longitudinal Trim Process
Initial CFD solution
While trim condition not fulfilled
(here CL=CL,Target, CMy=0):
If Jacobian is to be updated:
Computation of Jacobian:
• Perturbation of control variable (here AoA, HTP)
• Transfer perturbed control variable into BCs/mesh
• Compute CFD sol. for perturbed state
Compute Newton step (here for AoA & HTP)
Transfer altered control variable into BCs/mesh
Compute CFD solution
LOOP
Challenge: Rotation of HTP while preserving the fuselage geometry
DLR

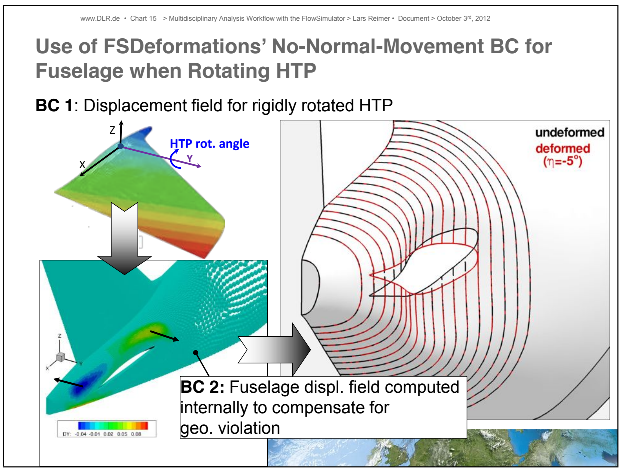

Computation of the Scattered Displacement Field for the HTP Rotation: FS' MeshTools (part of FSControl)
(48.7197, 0.0, 2.15)
z
x
htpAngle
Y
extraction of surface mesh
rigid rot. of extr. surf. mesh
base points & displacement field (= r-rRot)
RMC = RelationsModelControl.FromTauParaFile("tau.para", DM, CC)
RMC.AttachCoordinateSystemToNodes(Nodes = ["htpLeft", "htpRight"],
CommonNode = "HTP",
Origin = [48.7197, 0.0, 2.15])
define rotation axis in relation model
MeshTools = MeshTools(DM, RMC)
basePoints, displVecs = \
MeshTools.GetDefoVectorsForRotation(MeshKey = "myMesh",
ModelNodeName = "HTP",
Axis = "Y",
Angle = htpAngle)
apply rotation & return displ. field
DLR

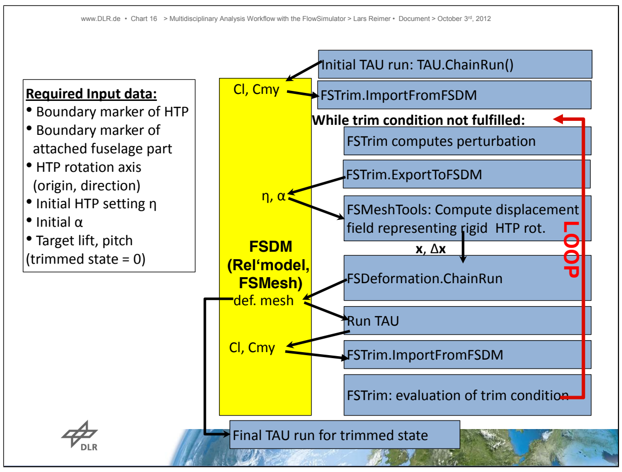

Problem with Mesh Deformation at HTP-Fuselage Intersection
Example A350-1000:
- HTP rot. in trim computations
- Prescribed displacement field corresponding to rigid HTP rot.:
undeformed
deformed
(η=-5°)
RZW≈1m
compression
stretching
HTP rot. angle
z
x
Y
η=-5°
DLR

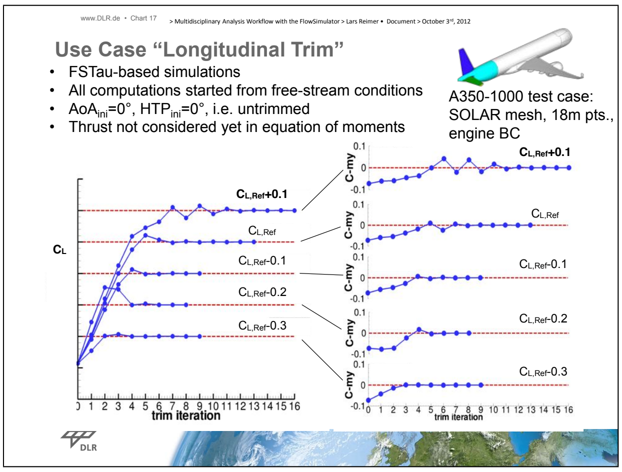

CAD-based Preparation of the Scattered Displacement Fields + Mesh Deformation: FSCAD2Mesh + FSDefoFields
FSCAD2Mesh
η = 0°
η = +20°
FSDeformation
undeformed
deformed
η = +20°
η = +10°
?
DLR


CAD-based Preparation of the Scattered Displacement Fields + Mesh Deformation: FSCAD2Mesh + FSDefoFields
FSCAD2Mesh
η = 0°
η = +20°
FSDeformation
undeformed
deformed
η = +20°
η = +10°
?
FSDefoFields
Interpolation of scattered displ. field + FSDeformation
deformed
η = +10°
DLR

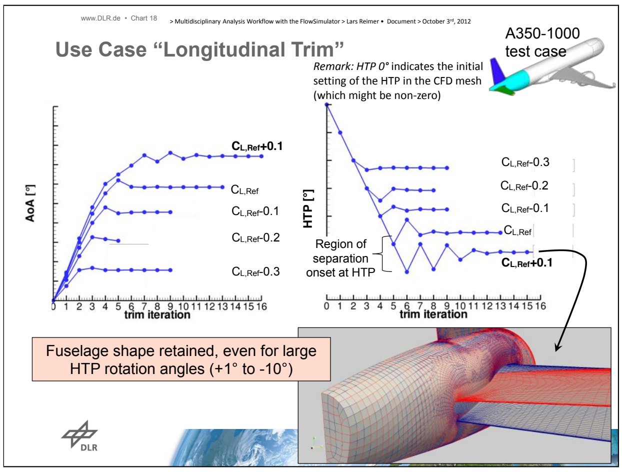

Other Strategy: Special "No-Normal-Movement" BC of FSDeformation
VTP
displacement of fuselage mesh point due to effect of HTP displacement field
+ incremental application of HTP displacements
fuselage
point of jig surface mesh
locally tangent plane
resulting displacement used for fuselage mesh point
displacement field of HTP
HTP
DLR


Use of FSDeformations' No-Normal-Movement BC for Fuselage when Rotating HTP
BC 1: Displacement field for rigidly rotated HTP
HTP rot. angle
z
x
Y
undeformed
deformed
(η=-5°)
BC 2: Fuselage displ. field computed internally to compensate for geo. violation
DLR

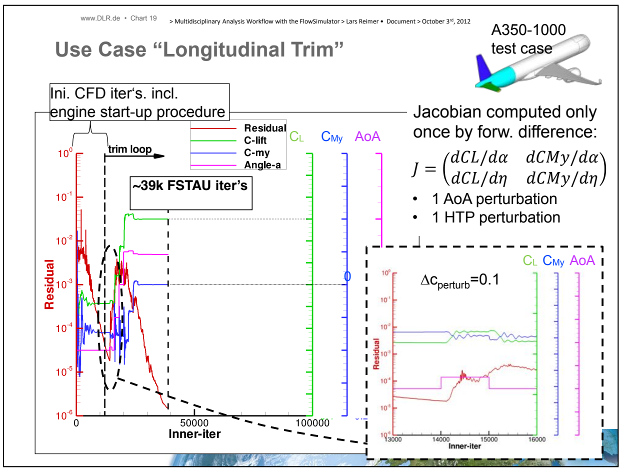

Required Input data:
• Boundary marker of HTP
• Boundary marker of attached fuselage part
• HTP rotation axis (origin, direction)
• Initial HTP setting η
• Initial α
• Target lift, pitch (trimmed state = 0)
Initial TAU run: TAU.ChainRun()
Cl, Cmy
FSTrim.ImportFromFSDM
While trim condition not fulfilled:
FSTrim computes perturbation
FSTrim.ExportToFSDM
η, α
FSMeshTools: Compute displacement field representing rigid HTP rot.
x, Δx
FSDM (Rel'model, FSMesh)
FSDeformation.ChainRun
def. mesh
Run TAU
Cl, Cmy
FSTrim.ImportFromFSDM
FSTrim: evaluation of trim condition
LOOP
Final TAU run for trimmed state
DLR



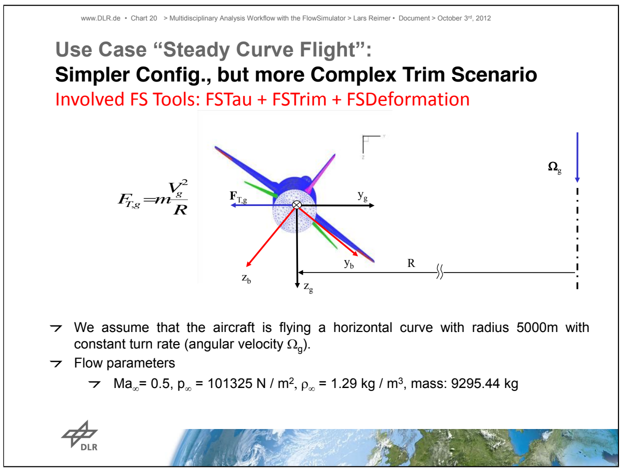
www.DLR.de • Chart 17 > Multidisciplinary Analysis Workflow with the FlowSimulator > Lars Reimer • Document > October 3rd, 2012
Use Case "Longitudinal Trim"
FSTau-based simulations
All computations started from free-stream conditions
AoA_ini=0°, HTP_ini=0°, i.e. untrimmed
Thrust not considered yet in equation of moments
A350-1000 test case: SOLAR mesh, 18m pts., engine BC
C_L
C_L,Ref+0.1
C_L,Ref
C_L,Ref-0.1
C_L,Ref-0.2
C_L,Ref-0.3
C-my
trim iteration
DLR



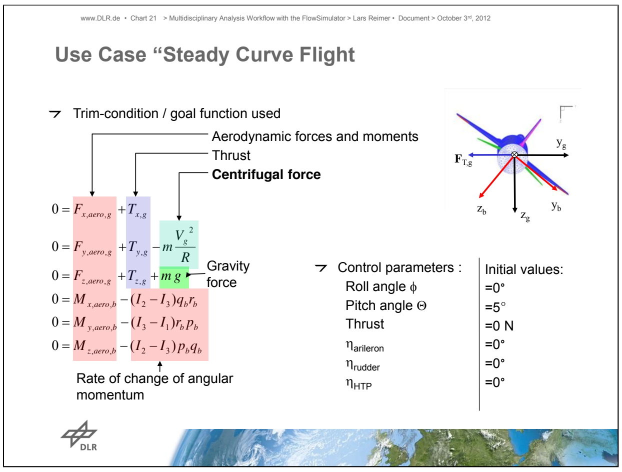
www.DLR.de • Chart 18 > Multidisciplinary Analysis Workflow with the FlowSimulator > Lars Reimer • Document > October 3rd, 2012
Use Case "Longitudinal Trim"
A350-1000 test case
Remark: HTP 0° indicates the initial setting of the HTP in the CFD mesh (which might be non-zero)
AoA [°]
HTP [°]
C_L,Ref+0.1
C_L,Ref
C_L,Ref-0.1
C_L,Ref-0.2
C_L,Ref-0.3
Region of separation onset at HTP
trim iteration
Fuselage shape retained, even for large HTP rotation angles (+1° to -10°)
DLR



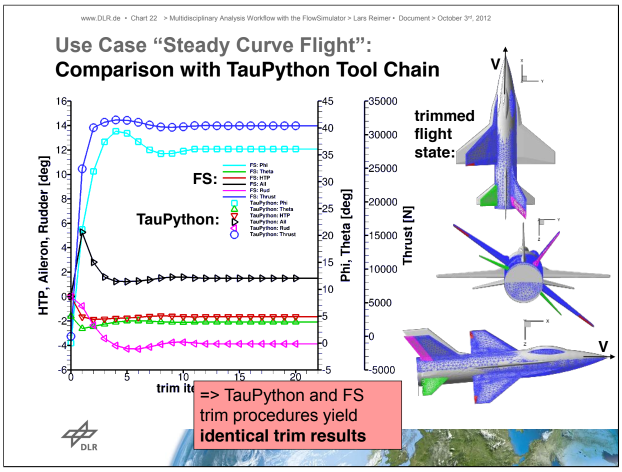
www.DLR.de • Chart 19 > Multidisciplinary Analysis Workflow with the FlowSimulator > Lars Reimer • Document > October 3rd, 2012
Use Case "Longitudinal Trim"
A350-1000 test case
Ini. CFD iter's. incl. engine start-up procedure
trim loop
~39k FSTAU iter's
Residual
C-lift
C-my
Angle-a
C_L C_My AoA
Jacobian computed only once by forw. difference:
$J = \begin{pmatrix} dCL/d\alpha & dCMy/d\alpha \\ dCL/d\eta & dCMy/d\eta \end{pmatrix}$
1 AoA perturbation
1 HTP perturbation
Δc_perturb=0.1
Residual
Inner-iter
DLR



www.DLR.de • Chart 20 > Multidisciplinary Analysis Workflow with the FlowSimulator > Lars Reimer • Document > October 3rd, 2012
Use Case "Steady Curve Flight":
Simpler Config., but more Complex Trim Scenario
Involved FS Tools: FSTau + FSTrim + FSDeformation
$F_{T,g} = m \frac{V_g^2}{R}$
We assume that the aircraft is flying a horizontal curve with radius 5000m with constant turn rate (angular velocity Ω_g).
Flow parameters
Ma_∞= 0.5, p_∞ = 101325 N / m², ρ_∞ = 1.29 kg / m³, mass: 9295.44 kg
DLR

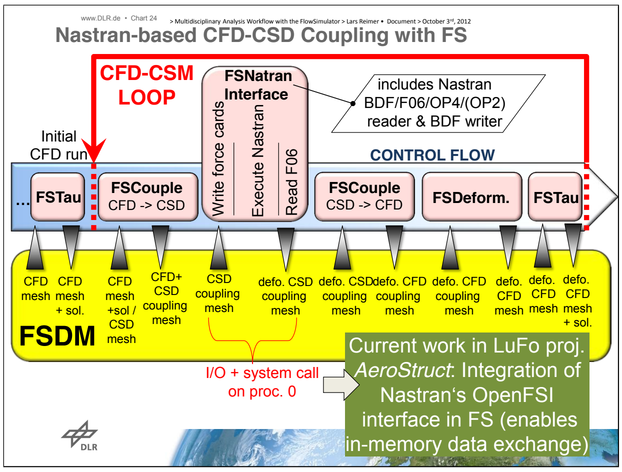


www.DLR.de • Chart 21 > Multidisciplinary Analysis Workflow with the FlowSimulator > Lars Reimer • Document > October 3rd, 2012
Use Case "Steady Curve Flight
Trim-condition / goal function used
Aerodynamic forces and moments
Thrust
Centrifugal force
$0 = F_{x,aero,g} + T_{x,g}$
$0 = F_{y,aero,g} + T_{y,g} - m\frac{V_g^2}{R}$
$0 = F_{z,aero,g} + T_{z,g} + mg$
$0 = M_{x,aero,b} - (I_2 - I_3) q_b r_b$
$0 = M_{y,aero,b} - (I_3 - I_1) r_b p_b$
$0 = M_{z,aero,b} - (I_2 - I_3) p_b q_b$
Gravity force
Rate of change of angular momentum
Control parameters :
Roll angle ϕ
Pitch angle Θ
Thrust
η_arileron
η_rudder
η_HTP
Initial values:
=0°
=5°
=0 N
=0°
=0°
=0°
DLR

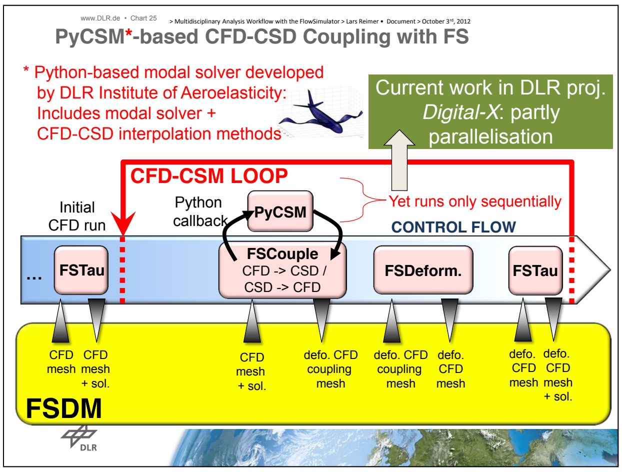


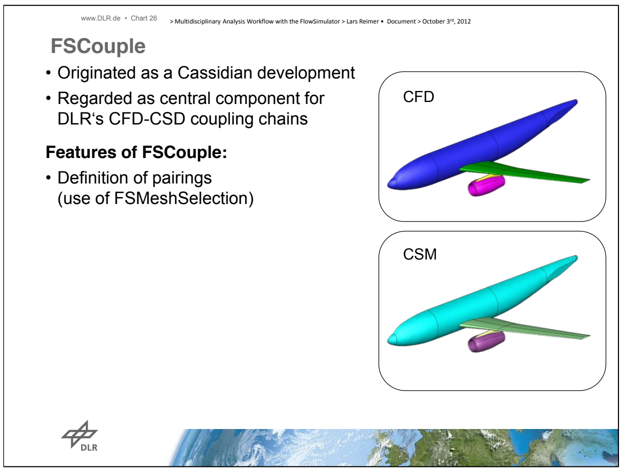
www.DLR.de • Chart 22 > Multidisciplinary Analysis Workflow with the FlowSimulator > Lars Reimer • Document > October 3rd, 2012
Use Case "Steady Curve Flight":
Comparison with TauPython Tool Chain
HTP, Aileron, Rudder [deg]
Phi, Theta [deg]
Thrust [N]
FS:
TauPython:
trimmed flight state:
V
=> TauPython and FS trim procedures yield identical trim results
DLR

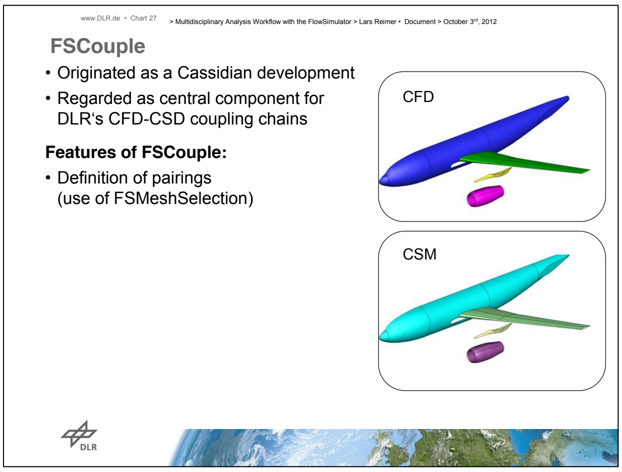
www.DLR.de • Chart 23 > Multidisciplinary Analysis Workflow with the FlowSimulator > Lars Reimer • Document > October 3rd, 2012
CFD-CSD Coupled Simulations with FlowSimulator
DLR

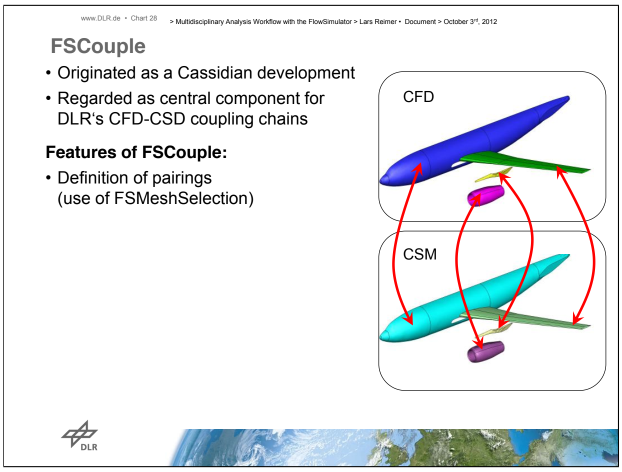


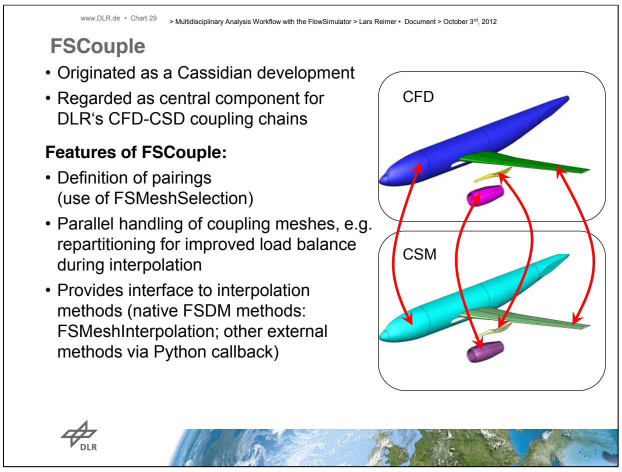
www.DLR.de • Chart 24 > Multidisciplinary Analysis Workflow with the FlowSimulator > Lars Reimer • Document > October 3rd, 2012
Nastran-based CFD-CSD Coupling with FS
CFD-CSM LOOP
FSNatran Interface
includes Nastran BDF/F06/OP4/(OP2) reader & BDF writer
Initial CFD run
CONTROL FLOW
FSTau
FSCouple CFD -> CSD
Write force cards
Execute Nastran
Read F06
FSCouple CSD -> CFD
FSDeform.
FSTau
CFD mesh
CFD mesh + sol.
CFD mesh +sol / CSD mesh
CFD+ CSD coupling mesh
CSD coupling mesh
defo. CSD coupling mesh
defo. CSD coupling mesh
defo. CFD coupling mesh
defo. CFD coupling mesh
defo. CFD mesh
defo. CFD mesh
defo. CFD mesh + sol.
FSDM
I/O + system call on proc. 0
Current work in LuFo proj. AeroStruct: Integration of Nastran's OpenFSI interface in FS (enables in-memory data exchange)
DLR

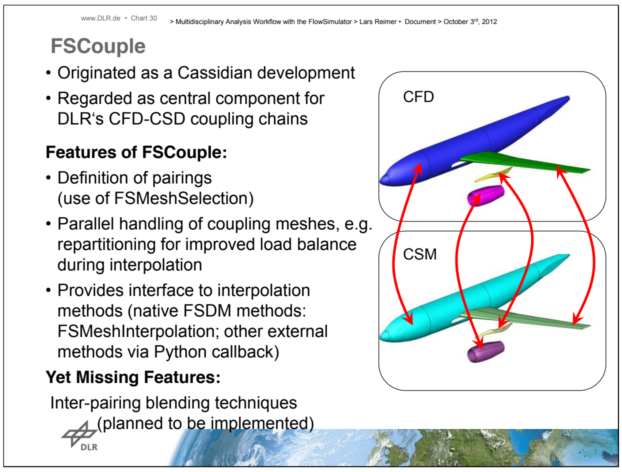

www.DLR.de • Chart 25 > Multidisciplinary Analysis Workflow with the FlowSimulator > Lars Reimer • Document > October 3rd, 2012
PyCSM*-based CFD-CSD Coupling with FS
* Python-based modal solver developed by DLR Institute of Aeroelasticity: Includes modal solver + CFD-CSD interpolation methods
Current work in DLR proj. Digital-X: partly parallelisation
CFD-CSM LOOP
Yet runs only sequentially
Initial CFD run
Python callback
PyCSM
CONTROL FLOW
...
FSTau
FSCouple CFD -> CSD / CSD -> CFD
FSDeform.
FSTau
CFD mesh
CFD mesh + sol.
CFD mesh + sol.
defo. CFD coupling mesh
defo. CFD coupling mesh
defo. CFD mesh
defo. CFD mesh
defo. CFD mesh + sol.
FSDM
DLR

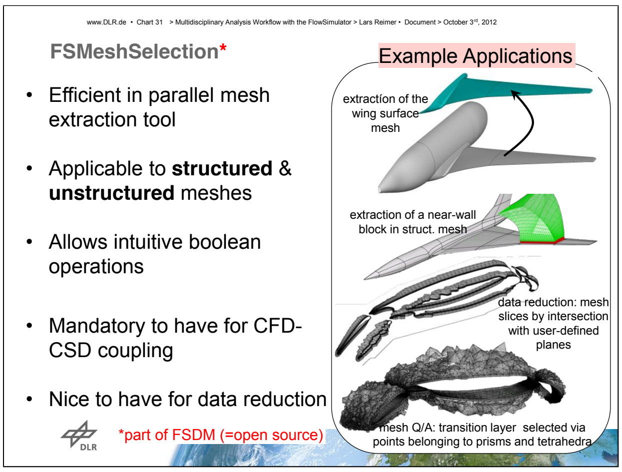

www.DLR.de • Chart 26 > Multidisciplinary Analysis Workflow with the FlowSimulator > Lars Reimer • Document > October 3rd, 2012
FSCouple
• Originated as a Cassidian development
• Regarded as central component for DLR‘s CFD-CSD coupling chains
Features of FSCouple:
• Definition of pairings (use of FSMeshSelection)
CFD
CSM
DLR

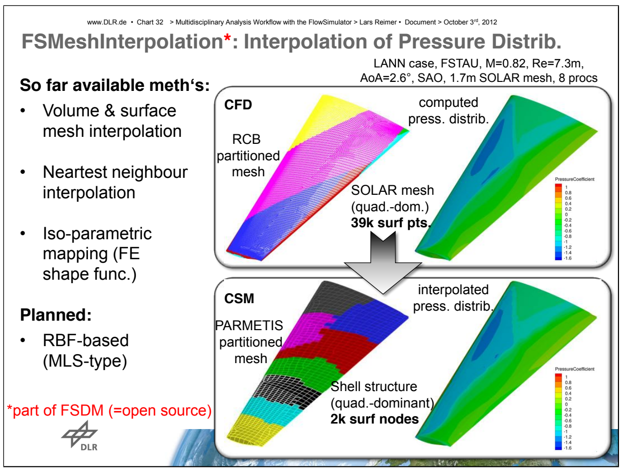

www.DLR.de • Chart 27 > Multidisciplinary Analysis Workflow with the FlowSimulator > Lars Reimer • Document > October 3rd, 2012
FSCouple
• Originated as a Cassidian development
• Regarded as central component for DLR‘s CFD-CSD coupling chains
Features of FSCouple:
• Definition of pairings (use of FSMeshSelection)
CFD
CSM
DLR

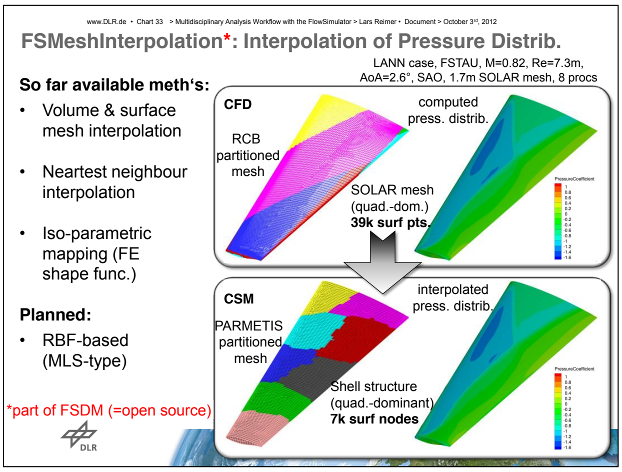

www.DLR.de • Chart 28 > Multidisciplinary Analysis Workflow with the FlowSimulator > Lars Reimer • Document > October 3rd, 2012
FSCouple
• Originated as a Cassidian development
• Regarded as central component for DLR‘s CFD-CSD coupling chains
Features of FSCouple:
• Definition of pairings (use of FSMeshSelection)
CFD
CSM
DLR

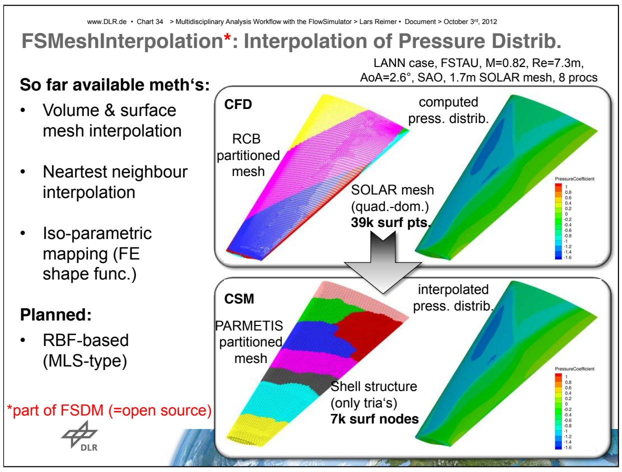

www.DLR.de • Chart 29 > Multidisciplinary Analysis Workflow with the FlowSimulator > Lars Reimer • Document > October 3rd, 2012
FSCouple
• Originated as a Cassidian development
• Regarded as central component for DLR‘s CFD-CSD coupling chains
Features of FSCouple:
• Definition of pairings (use of FSMeshSelection)
• Parallel handling of coupling meshes, e.g. repartitioning for improved load balance during interpolation
• Provides interface to interpolation methods (native FSDM methods: FSMeshInterpolation; other external methods via Python callback)
CFD
CSM
DLR

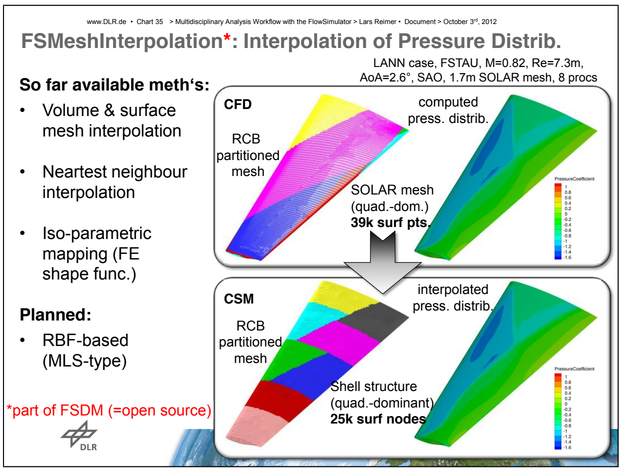

www.DLR.de • Chart 30 > Multidisciplinary Analysis Workflow with the FlowSimulator > Lars Reimer • Document > October 3rd, 2012
FSCouple
• Originated as a Cassidian development
• Regarded as central component for DLR‘s CFD-CSD coupling chains
Features of FSCouple:
• Definition of pairings (use of FSMeshSelection)
• Parallel handling of coupling meshes, e.g. repartitioning for improved load balance during interpolation
• Provides interface to interpolation methods (native FSDM methods: FSMeshInterpolation; other external methods via Python callback)
Yet Missing Features:
Inter-pairing blending techniques (planned to be implemented)
CFD
CSM
DLR

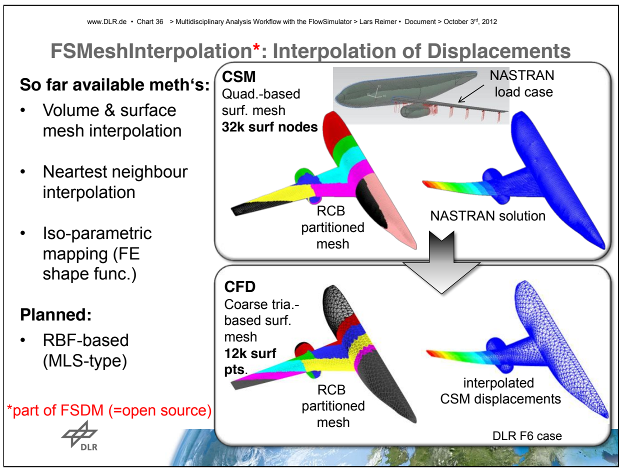

www.DLR.de • Chart 31 > Multidisciplinary Analysis Workflow with the FlowSimulator > Lars Reimer • Document > October 3rd, 2012
FSMeshSelection*
• Efficient in parallel mesh extraction tool
• Applicable to structured & unstructured meshes
• Allows intuitive boolean operations
• Mandatory to have for CFD-CSD coupling
• Nice to have for data reduction
*part of FSDM (=open source)
Example Applications
extraction of the wing surface mesh
extraction of a near-wall block in struct. mesh
data reduction: mesh slices by intersection with user-defined planes
mesh Q/A: transition layer selected via points belonging to prisms and tetrahedra
DLR

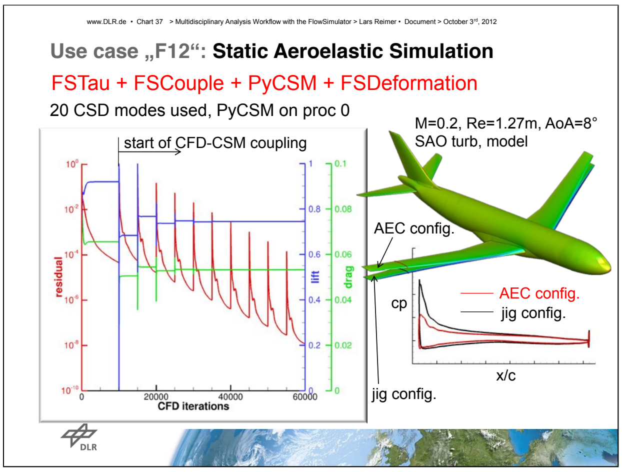

www.DLR.de • Chart 32 > Multidisciplinary Analysis Workflow with the FlowSimulator > Lars Reimer • Document > October 3rd, 2012
FSMeshInterpolation*: Interpolation of Pressure Distrib.
LANN case, FSTAU, M=0.82, Re=7.3m, AoA=2.6°, SAO, 1.7m SOLAR mesh, 8 procs
So far available meth‘s:
• Volume & surface mesh interpolation
• Neartest neighbour interpolation
• Iso-parametric mapping (FE shape func.)
Planned:
• RBF-based (MLS-type)
*part of FSDM (=open source)
CFD
RCB partitioned mesh
computed press. distrib.
SOLAR mesh (quad.-dom.) 39k surf pts.
CSM
PARMETIS partitioned mesh
interpolated press. distrib.
Shell structure (quad.-dominant) 2k surf nodes
DLR

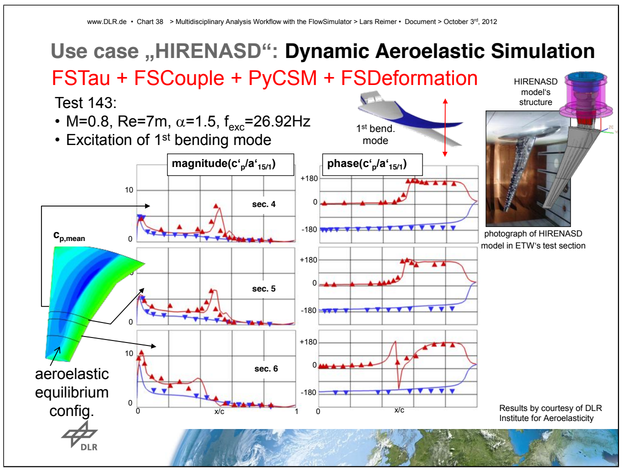

www.DLR.de • Chart 33 > Multidisciplinary Analysis Workflow with the FlowSimulator > Lars Reimer • Document > October 3rd, 2012
FSMeshInterpolation*: Interpolation of Pressure Distrib.
LANN case, FSTAU, M=0.82, Re=7.3m, AoA=2.6°, SAO, 1.7m SOLAR mesh, 8 procs
So far available meth's:
• Volume & surface mesh interpolation
• Neartest neighbour interpolation
• Iso-parametric mapping (FE shape func.)
Planned:
• RBF-based (MLS-type)
*part of FSDM (=open source)
CFD
RCB partitioned mesh
computed press. distrib.
SOLAR mesh (quad.-dom.) 39k surf pts.
CSM
PARMETIS partitioned mesh
interpolated press. distrib.
Shell structure (quad.-dominant) 7k surf nodes
DLR

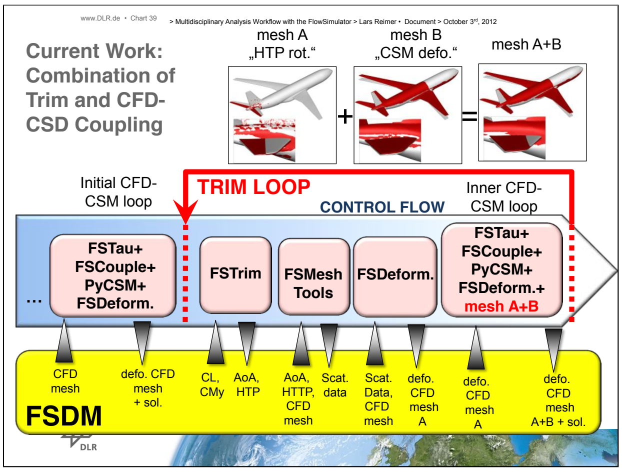

www.DLR.de • Chart 34 > Multidisciplinary Analysis Workflow with the FlowSimulator > Lars Reimer • Document > October 3rd, 2012
FSMeshInterpolation*: Interpolation of Pressure Distrib.
LANN case, FSTAU, M=0.82, Re=7.3m, AoA=2.6°, SAO, 1.7m SOLAR mesh, 8 procs
So far available meth's:
• Volume & surface mesh interpolation
• Neartest neighbour interpolation
• Iso-parametric mapping (FE shape func.)
Planned:
• RBF-based (MLS-type)
*part of FSDM (=open source)
CFD
RCB partitioned mesh
computed press. distrib.
SOLAR mesh (quad.-dom.) 39k surf pts.
CSM
PARMETIS partitioned mesh
interpolated press. distrib.
Shell structure (only tria's) 7k surf nodes
DLR

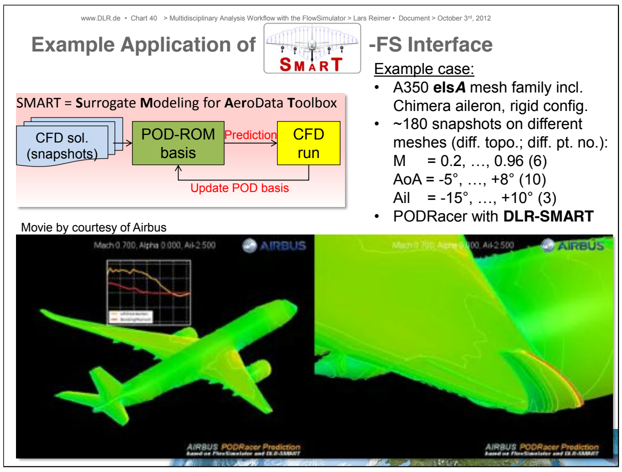

www.DLR.de • Chart 35 > Multidisciplinary Analysis Workflow with the FlowSimulator > Lars Reimer • Document > October 3rd, 2012
FSMeshInterpolation*: Interpolation of Pressure Distrib.
LANN case, FSTAU, M=0.82, Re=7.3m, AoA=2.6°, SAO, 1.7m SOLAR mesh, 8 procs
So far available meth's:
• Volume & surface mesh interpolation
• Neartest neighbour interpolation
• Iso-parametric mapping (FE shape func.)
Planned:
• RBF-based (MLS-type)
*part of FSDM (=open source)
CFD
RCB partitioned mesh
computed press. distrib.
SOLAR mesh (quad.-dom.) 39k surf pts.
CSM
RCB partitioned mesh
interpolated press. distrib.
Shell structure (quad.-dominant) 25k surf nodes
DLR


www.DLR.de • Chart 36 > Multidisciplinary Analysis Workflow with the FlowSimulator > Lars Reimer • Document > October 3rd, 2012
FSMeshInterpolation*: Interpolation of Displacements
So far available meth's:
• Volume & surface mesh interpolation
• Neartest neighbour interpolation
• Iso-parametric mapping (FE shape func.)
Planned:
• RBF-based (MLS-type)
*part of FSDM (=open source)
CSM
Quad.-based surf. mesh
32k surf nodes
NASTRAN load case
RCB partitioned mesh
NASTRAN solution
CFD
Coarse tria.-based surf. mesh
12k surf pts.
RCB partitioned mesh
interpolated CSM displacements
DLR F6 case
DLR


www.DLR.de • Chart 37 > Multidisciplinary Analysis Workflow with the FlowSimulator > Lars Reimer • Document > October 3rd, 2012
Use case „F12“: Static Aeroelastic Simulation
FSTau + FSCouple + PyCSM + FSDeformation
20 CSD modes used, PyCSM on proc 0
start of CFD-CSM coupling
M=0.2, Re=1.27m, AoA=8° SAO turb, model
AEC config.
residual
lift
drag
CFD iterations
cp
AEC config.
jig config.
x/c
jig config.
DLR


www.DLR.de • Chart 38 > Multidisciplinary Analysis Workflow with the FlowSimulator > Lars Reimer • Document > October 3rd, 2012
Use case „HIRENASD“: Dynamic Aeroelastic Simulation
FSTau + FSCouple + PyCSM + FSDeformation
Test 143:
• M=0.8, Re=7m, α=1.5, fexc=26.92Hz
• Excitation of 1st bending mode
HIRENASD model's structure
1st bend. mode
magnitude(c'p/a'15/1)
phase(c'p/a'15/1)
cp,mean
sec. 4
sec. 5
sec. 6
x/c
aeroelastic equilibrium config.
photograph of HIRENASD model in ETW's test section
Results by courtesy of DLR Institute for Aeroelasticity
DLR

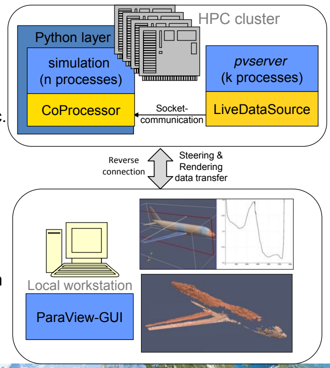

www.DLR.de • Chart 39 > Multidisciplinary Analysis Workflow with the FlowSimulator > Lars Reimer • Document > October 3rd, 2012
Current Work: Combination of Trim and CFD-CSD Coupling
mesh A „HTP rot.“
mesh B „CSM defo.“
mesh A+B
Initial CFD-CSM loop
TRIM LOOP
CONTROL FLOW
Inner CFD-CSM loop
FSTau+ FSCouple+ PyCSM+ FSDeform.
FSTrim
FSMesh Tools
FSDeform.
FSTau+ FSCouple+ PyCSM+ FSDeform.+ mesh A+B
CFD mesh
defo. CFD mesh + sol.
CL, CMy
AoA, HTP
AoA, HTTP, CFD mesh
Scat. data
Scat. Data, CFD mesh
defo. CFD mesh A
defo. CFD mesh A
defo. CFD mesh A+B + sol.
FSDM
DLR


www.DLR.de • Chart 40 > Multidisciplinary Analysis Workflow with the FlowSimulator > Lars Reimer • Document > October 3rd, 2012
Example Application of SMART-FS Interface
SMART = Surrogate Modeling for AeroData Toolbox
CFD sol. (snapshots)
POD-ROM basis
Prediction
CFD run
Update POD basis
Example case:
• A350 elsA mesh family incl. Chimera aileron, rigid config.
• ~180 snapshots on different meshes (diff. topo.; diff. pt. no.):
M = 0.2, …, 0.96 (6)
AoA = -5°, …, +8° (10)
Ail = -15°, …, +10° (3)
• PODRacer with DLR-SMART
Movie by courtesy of Airbus
AIRBUS

Co-processing-based Visualisation

Benefits:
- Online monitoring
- Data reduction
- In parallel with sim. w/o file I/O
- Parallel graphical data proc.

Implementation:
- Based on ParaView (version 3.9 + small own modifications)
- Uses ParaView-CoProcessing lib for Python-based integration in simulation
- FS interface available and applied at Airbus



DLR

Co-processing-based Visualisation

DLR

Summary

- Potential of FS demonstrated for multidisciplinary parallel analysis
- DLR strives to use FS for all multidisciplinary simulations in the near future
- DLR takes over maintenance of FSDeformation
- DLR will further develop FSTrim, FSDeformation, FSCouple, FSDM and will contribute FS6DoF
- With ongoing FS development more synergy effects of ONERA and DLR contributations can be expected

DLR

Examples of Currently Running Projects Involving DLR with Extense Usage and Development of FlowSImulator

- DLR project DIGITAL X
- German Aerospace Research Programme (LuFo) project AeroStruct
- CFD4Loads project (initiated by Airbus)

AIRBUS ONERA ARA DLR

…

DLR

End

DLR

2.6 D. Snyder (CD-adapco)

Abstract: Onera Scientific Day 2012

STAR-CCM+: A New Approach to Numerical Simulation
Deryl Snyder, Ph.D.
CD-adapco

If asked for the basic definition of CFD, the most common response will likely be something similar to: numerical methods and algorithms to solve problems involving fluid flow. From an industrial standpoint, however, CFD is more than that. It is a tool to design and/or analyze components or systems that have some aspect related to fluid flow. There is a key difference between the two definitions. The first is essentially referring only to the CFD solver. The second incorporates the entire CFD process, from the input geometry definition through the final desired data extracted from the numerical solution. This is the definition that is important to keep in mind when discussing CFD workflow. To that end, I will present the integrated CAD-to-solution workflow developed within STAR-CCM+, the flagship general-purpose, high-end physics, CFD-focused CAE tool from CD-adapco.

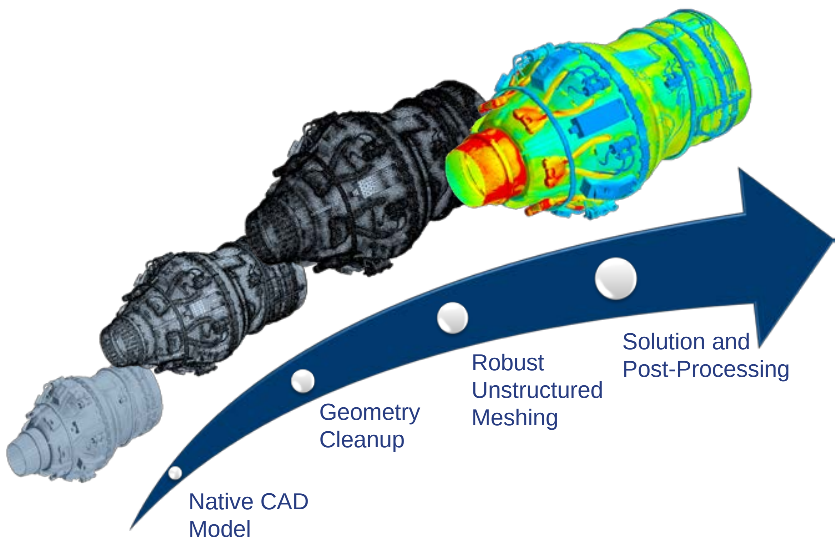


Figure 1. STAR-CCM+ integrated workflow; from native CAD geometry through post-processed solution.

Today, one of the greatest challenges facing the aerospace industry is remaining at the forefront of innovation while coping with reduced budgets, increased performance requirements, and intense competition. CFD, along with other computer-aided simulation tools in various disciplines, is critical in successfully facing this challenge. Not surprisingly, CFD analysis trends seen throughout the industry include increased physics and geometric complexity, coupled with shortened schedules. In addition, there is a desire to utilize CFD earlier in the design process in order to achieve a greater impact on the final design. To do so, the process needs to be fast, repeatable, and automated. Currently, on the order of 65-80% of the engineer’s time spent on CFD analysis is in the pre-processing stage, so obviously this is an area where time-reducing capabilities in the workflow can have large returns. These factors are at the heart of the design of the STAR-CCM+ workflow.

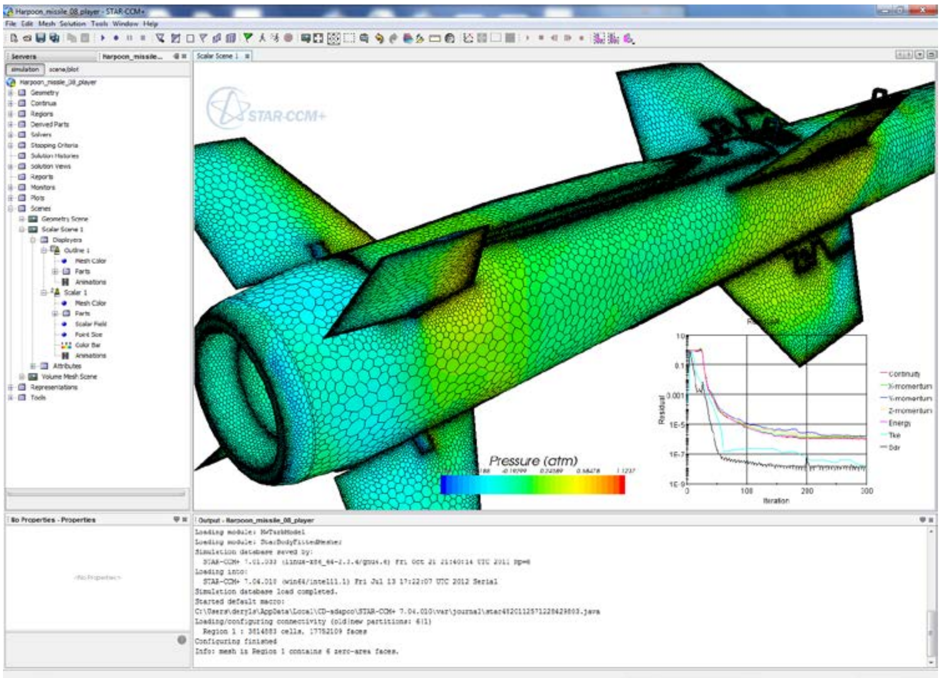


Figure 2. STAR-CCM+ integrated workspace, including tree-structure access to settings and information for the geometry, meshing, solution, and post-processing components of the analysis process.

Key components to the workflow are highlighted below. These will be discussed in detail in the final paper and presentation.

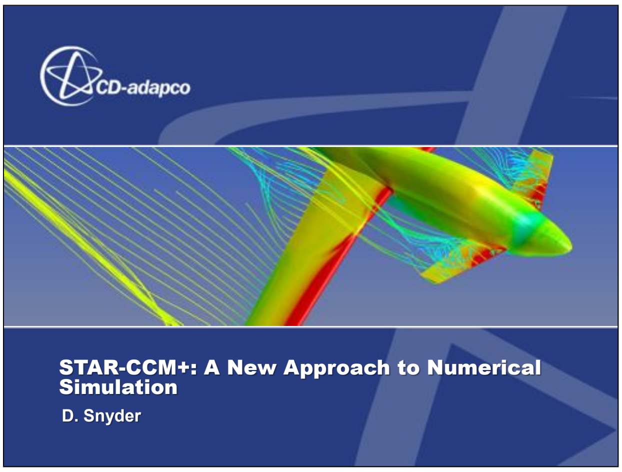
1. *A single software program for the entire CFD analysis process.* This is the primary differentiator of the STAR-CCM+ workflow, and goes beyond building a single user interface for multiple, but individual, software components. By integrating all components into a single program, an efficient pipeline is created, where geometry, mesh, solution, and post-processing are all "two-way" coupled, allowing greater flexibility and giving the engineer greater insight into the results.
2. *Handling of CAD geometry.* Dealing directly with CAD tools and CAD files is necessary to efficiently cope with design modifications expected at the early stages of the development process. Several approaches are available, including integration directly into CAD tools, directly importing native CAD file formats, and construction of CAD geometry directly within the CFD program. In all cases, the ability to parameterize the geometry is important.
3. *Geometry preparation.* In this part of the workflow, the CAD geometry, which is often of poor quality, is converted into a closed volume ready for meshing. Approaches for doing so include surface wrapping – which automatically de-features, closes gaps, and determines leakage paths – or surface repair tools.
4. *Mesh generation.* For industrial applications, the use of unstructured meshes has proven to be the method of choice, and continues to gain popularity. Many studies (published and unpublished) have shown that properly-built unstructured meshes can produce results of the same quality as structured meshes, but at a greatly reduced mesh generation cost. Advanced

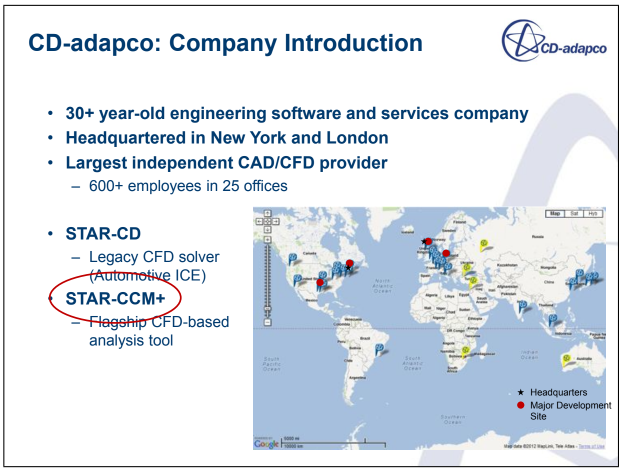
meshing algorithms that employ general polyhedral cells are used to produce quality meshes on complex geometries.

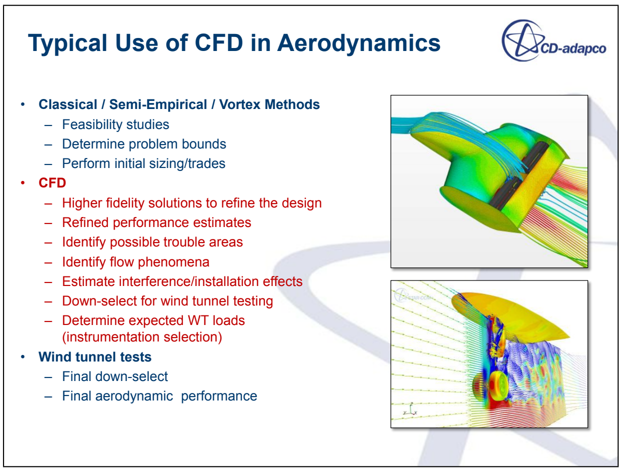
5. *Physics solution.* In addition to robust and accurate numerics, advanced algorithms for solution initialization and convergence steering allow the engineer to be more hands-off during the solution phase.
6. *Post-processing.* Real-time post processing, even on large computing clusters.

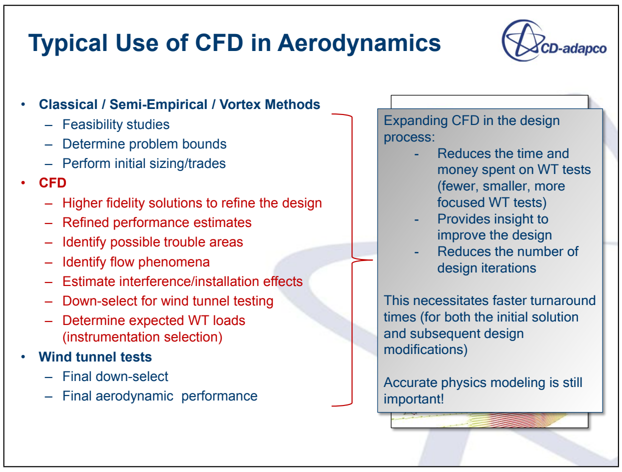
Other aspects of the workflow that will be discussed:

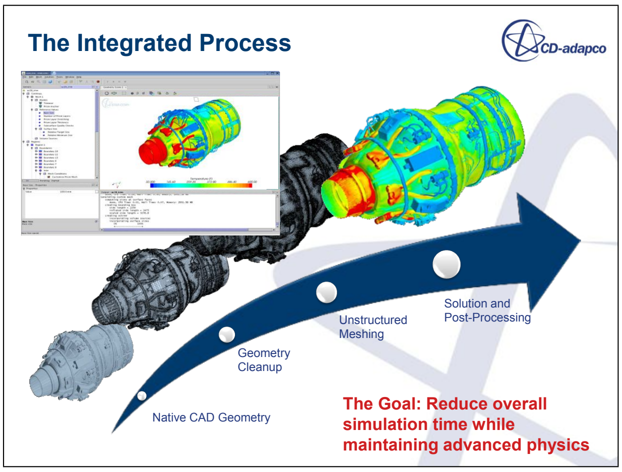
1. *Client-server architecture.* This software approach allows interaction with the program during any stage of the analysis on remote computers, including large computing clusters. In addition, this allows multiple users to connect to the same simulation simultaneously, resulting in a truly collaborative procedure.
2. *Automation.* The complete CAD-to-solution process is automated using a unified macro in order to improve efficiency and hardware utilization, as well as incorporate best practices into the solution process.

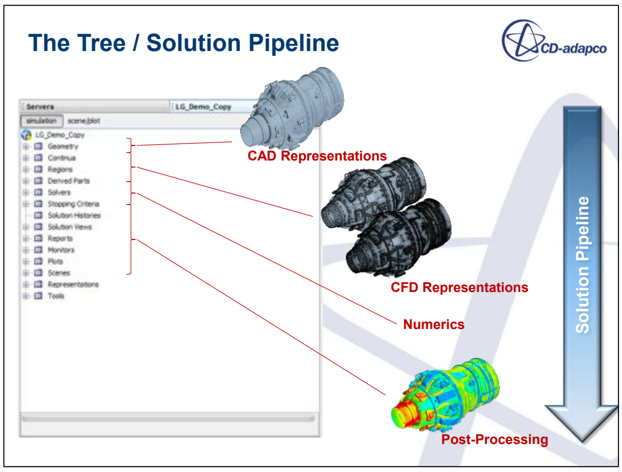

CD-adapco
STAR-CCM+: A New Approach to Numerical Simulation
D. Snyder

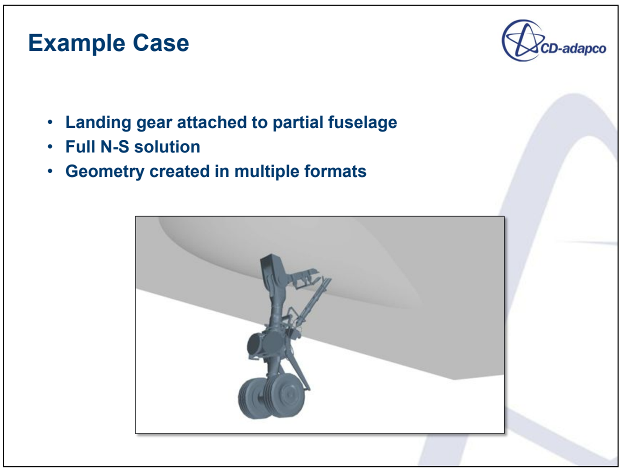

CD-adapco: Company Introduction
CD-adapco
• 30+ year-old engineering software and services company
• Headquartered in New York and London
• Largest independent CAD/CFD provider
– 600+ employees in 25 offices
• STAR-CD
– Legacy CFD solver (Automotive ICE)
• STAR-CCM+
– Flagship CFD-based analysis tool
★ Headquarters
● Major Development Site

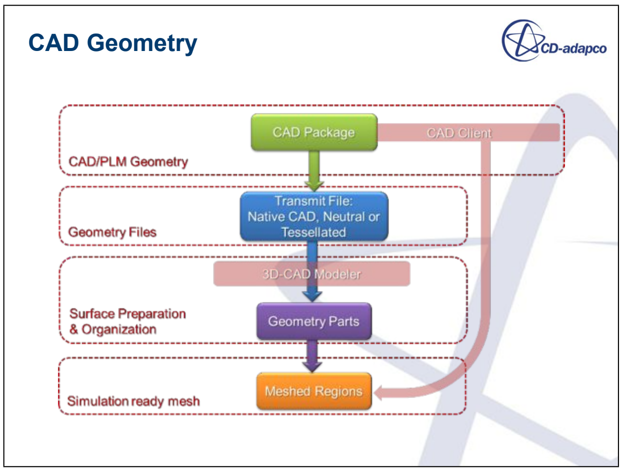

Typical Use of CFD in Aerodynamics
CD-adapco
• Classical / Semi-Empirical / Vortex Methods
– Feasibility studies
– Determine problem bounds
– Perform initial sizing/trades
• CFD
– Higher fidelity solutions to refine the design
– Refined performance estimates
– Identify possible trouble areas
– Identify flow phenomena
– Estimate interference/installation effects
– Down-select for wind tunnel testing
– Determine expected WT loads (instrumentation selection)
• Wind tunnel tests
– Final down-select
– Final aerodynamic performance


Typical Use of CFD in Aerodynamics
CD-adapco
• Classical / Semi-Empirical / Vortex Methods
– Feasibility studies
– Determine problem bounds
– Perform initial sizing/trades
• CFD
– Higher fidelity solutions to refine the design
– Refined performance estimates
– Identify possible trouble areas
– Identify flow phenomena
– Estimate interference/installation effects
– Down-select for wind tunnel testing
– Determine expected WT loads (instrumentation selection)
• Wind tunnel tests
– Final down-select
– Final aerodynamic performance
Expanding CFD in the design process:
- Reduces the time and money spent on WT tests (fewer, smaller, more focused WT tests)
- Provides insight to improve the design
- Reduces the number of design iterations
This necessitates faster turnaround times (for both the initial solution and subsequent design modifications)
Accurate physics modeling is still important!

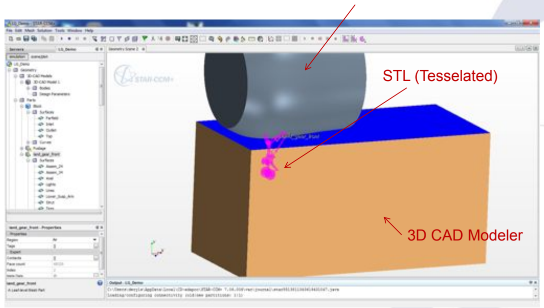

The Integrated Process
CD-adapco
Native CAD Geometry
Geometry Cleanup
Unstructured Meshing
Solution and Post-Processing
The Goal: Reduce overall simulation time while maintaining advanced physics


The Tree / Solution Pipeline
CD-adapco
CAD Representations
CFD Representations
Numerics
Post-Processing
Solution Pipeline


Example Case
CD-adapco
• Landing gear attached to partial fuselage
• Full N-S solution
• Geometry created in multiple formats

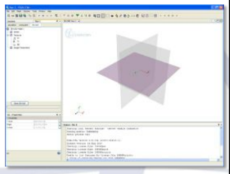

CAD Geometry
CD-adapco
CAD/PLM Geometry
CAD Package
CAD Client
Geometry Files
Transmit File: Native CAD, Neutral or Tessellated
3D-CAD Modeler
Surface Preparation & Organization
Geometry Parts
Simulation ready mesh
Meshed Regions

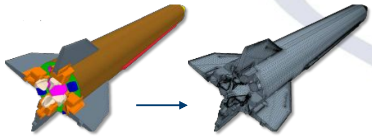
CAD Geometry

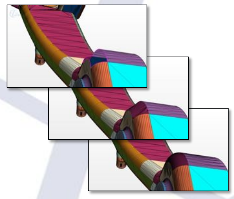


Surface Preparation

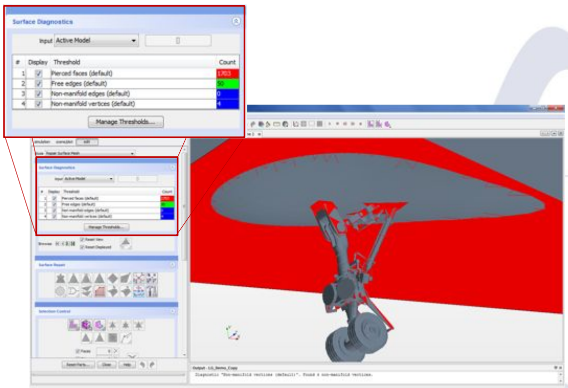
- **Automatic Surface Wrapper to prepare complex/dirty geometry**
  - Automatic boundary extraction
  - Feature detection and automatic refinement
  - Automatic de-featuring
  - Global and local control of all settings
- **Surface repair tools**
  - Manual and/or automatic tools
  - De-feature, fill holes, repair, imprint, split, etc.
- **3D CAD Modeler to generate / modify geometry**



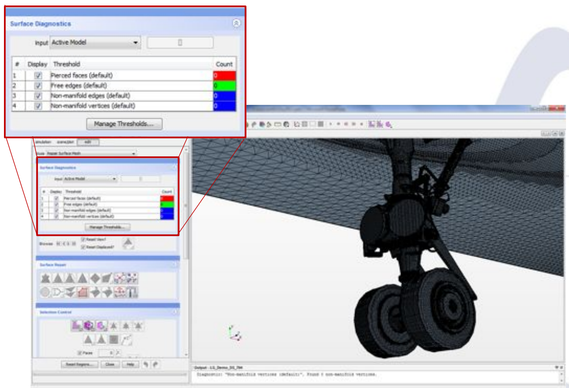




Surface Repair Tools

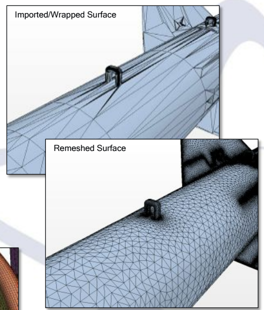


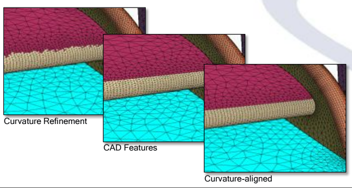
Surface Wrapper



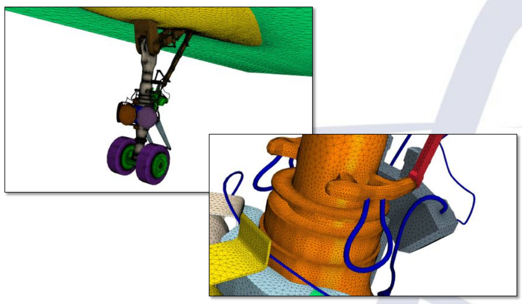
Automatic Surface Remesher

- **Re-triangulates a closed surface**
  - Generates a quality triangle surface mesh used as basis for volume mesh
  - Sizing controls
    - Global / Region / Surface / Edge
  - Maintain CAD features
  - Curvature-aligned meshes

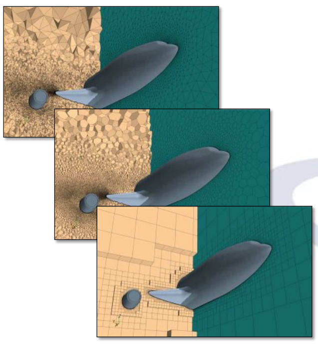




Remeshed Surface



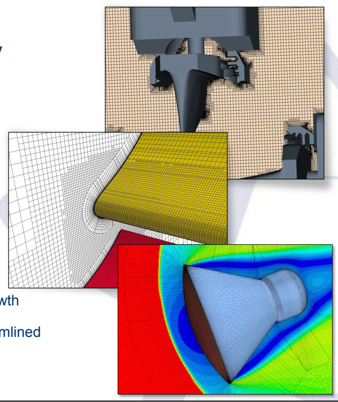
Automatic Volume Meshing



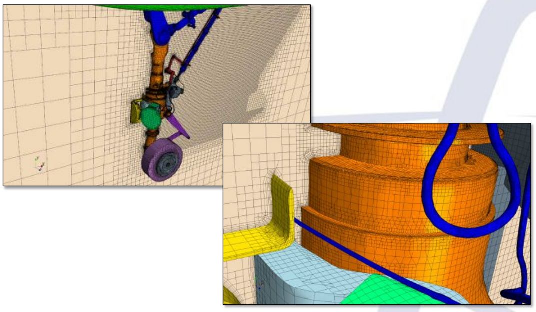
- **Tetrahedra**
  - Legacy purposes
- **Polyhedra**
  - Varying flow direction
  - Recirculating Flow
  - Vortical Flow
- **Unstructured Hex (Cartesian)**
  - Well defined primary flow direction

Automatic Prism Layer Generation

- **Robust for complex geometry**
- **High quality**
  - Grid lines normal to the wall
  - Consistent near-wall thickness
- **Full refinement control**
  - Number of layers
  - Stretching function
  - First layer thickness
  - Total extrusion layer thickness
- **Automatic geometry-based thickness adaptation**
  - Compression
  - Retraction
- **Advancing Layer Mesher**
  - Smooth, pseudo-structured growth from wall boundaries
  - Better align grid lines with streamlined flow, shocks, etc.

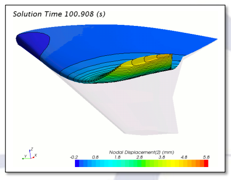


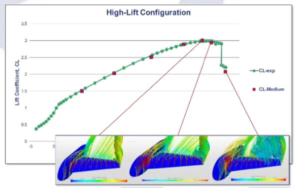
Volume Mesh



Physics Modeling

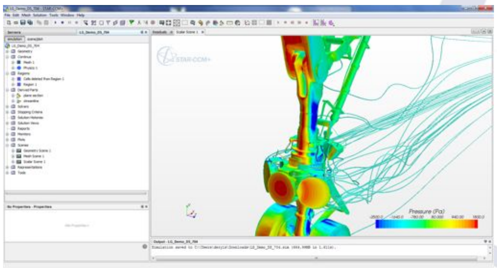
- **Density- and Pressure-Based Flow Solvers**
  - Robust, stable, accurate
  - Incompressible through hypersonic
  - Advanced initialization and convergence acceleration routines
  - Full set of turbulence models
  - Boundary-layer transition models
  - Multi-Species Flow
  - Combustion
  - …
- **Multi-Physics**
  - CHT / Thermal Stress
  - Aeroacoustics
  - Fluid-Structure Interaction
  - Electromagnetics
  - …



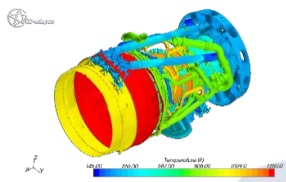


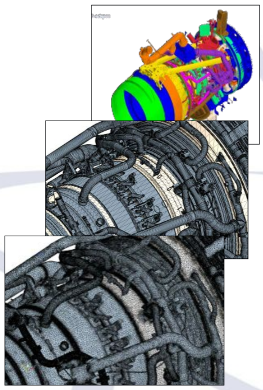
Solution/Post Processing

- **Post processing can be performed concurrent with solution**
  - Scenes, reports, monitors, plots, integrals, etc.
- **Serial/Parallel**
- **Workstation/Computing Cluster**



Engine Thermal

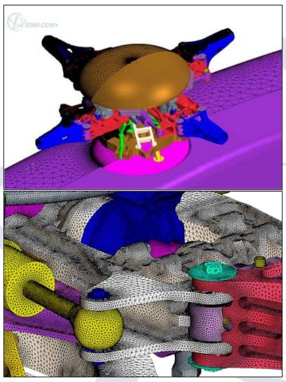




- Original Process: 2 months to generate mesh
  - No prism layer
  - Simplified geometry
- STAR-CCM+ full CAD model: 1 week (including solution)

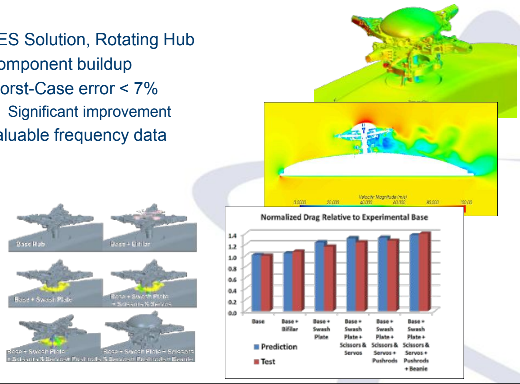
Rotorcraft Hub Drag: Mesh

- Hub drag contributes up to 30% of total vehicle drag
- Prediction of total hub drag and contribution of each component is desired
- Simplified "classical" approach does not provide consistently accurate results
- Meshing complex geometry has been too expensive (weeks to months) so CFD has not been used.
- STAR-CCM+ CAD-to-mesh in 2 days, keeping full complexity



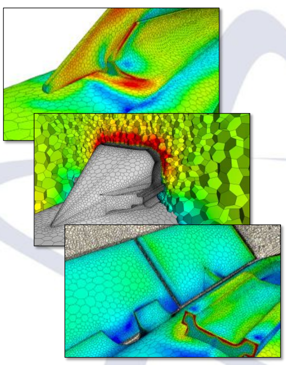
Rotorcraft Hub Drag: Results

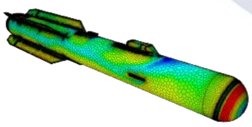
- DES Solution, Rotating Hub
- Component buildup
- Worst-Case error < 7%
  - Significant improvement
- Valuable frequency data



Missile Aerodynamics: Mesh

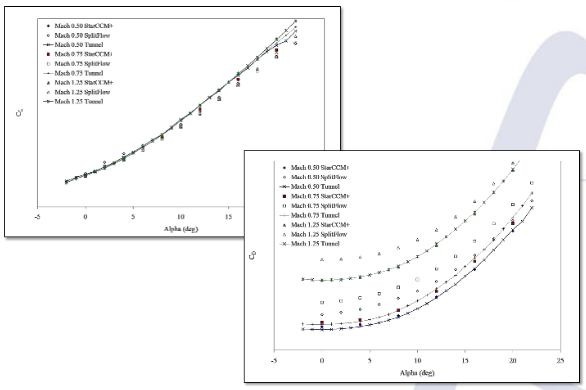
- Retaining complex geometric features (lugs, straps, steps, etc.) necessary to achieve desired accuracy
- Robust prism layer generation
- Refinement regions to capture bluff body wakes, base pressure, etc.
- Fast turn-around time. Pre-processing time reduced by a factor of 5 over previous tools.



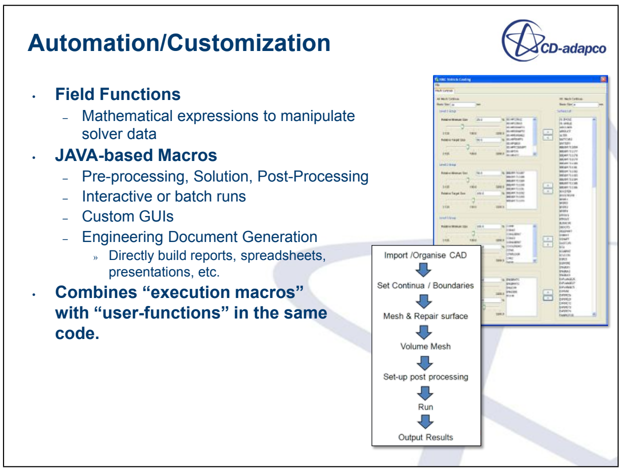


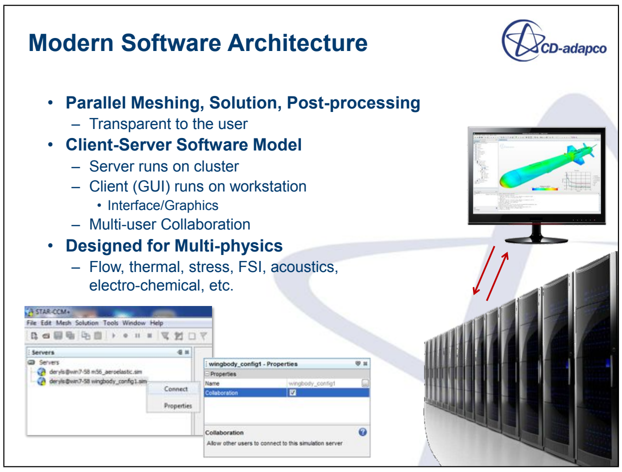
Lockheed Martin Public Release: ORL201102002

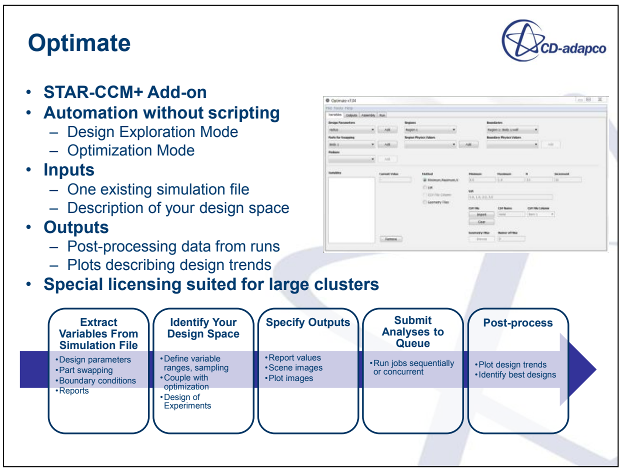
Missile Aerodynamics: Results

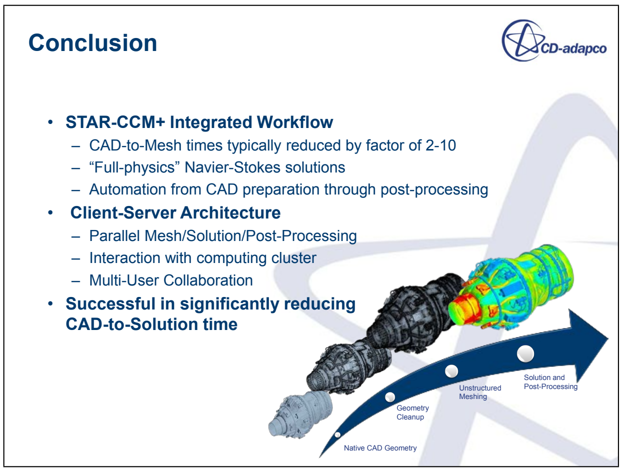


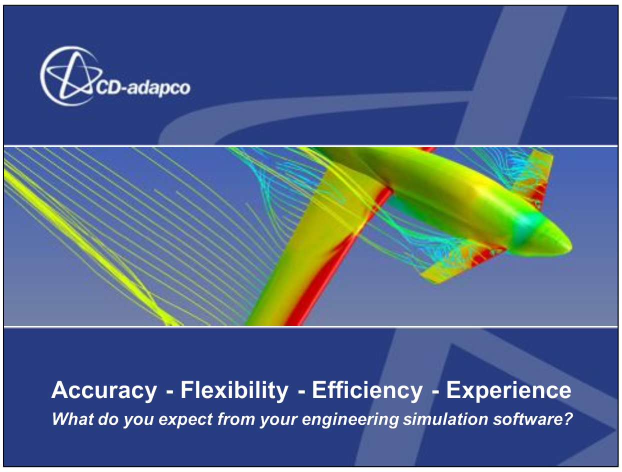
Lockheed Martin Public Release: ORL201102002


Automation/Customization
CD-adapco
Field Functions
Mathematical expressions to manipulate solver data
JAVA-based Macros
Pre-processing, Solution, Post-Processing
Interactive or batch runs
Custom GUIs
Engineering Document Generation
Directly build reports, spreadsheets, presentations, etc.
Combines "execution macros" with "user-functions" in the same code.
Import /Organise CAD
Set Continua / Boundaries
Mesh & Repair surface
Volume Mesh
Set-up post processing
Run
Output Results


Modern Software Architecture
CD-adapco
Parallel Meshing, Solution, Post-processing
Transparent to the user
Client-Server Software Model
Server runs on cluster
Client (GUI) runs on workstation
Interface/Graphics
Multi-user Collaboration
Designed for Multi-physics
Flow, thermal, stress, FSI, acoustics, electro-chemical, etc.


Optimate
CD-adapco
STAR-CCM+ Add-on
Automation without scripting
Design Exploration Mode
Optimization Mode
Inputs
One existing simulation file
Description of your design space
Outputs
Post-processing data from runs
Plots describing design trends
Special licensing suited for large clusters
Extract Variables From Simulation File
Design parameters
Part swapping
Boundary conditions
Reports
Identify Your Design Space
Define variable ranges, sampling
Couple with optimization
Design of Experiments
Specify Outputs
Report values
Scene images
Plot images
Submit Analyses to Queue
Run jobs sequentially or concurrent
Post-process
Plot design trends
Identify best designs


Conclusion
CD-adapco
STAR-CCM+ Integrated Workflow
CAD-to-Mesh times typically reduced by factor of 2-10
"Full-physics" Navier-Stokes solutions
Automation from CAD preparation through post-processing
Client-Server Architecture
Parallel Mesh/Solution/Post-Processing
Interaction with computing cluster
Multi-User Collaboration
Successful in significantly reducing CAD-to-Solution time
Native CAD Geometry
Geometry Cleanup
Unstructured Meshing
Solution and Post-Processing


CD-adapco
Accuracy - Flexibility - Efficiency - Experience
What do you expect from your engineering simulation software?

2.7 V. Morgenthaler (ANSYS France)

Abstract: Onera Scientific Day October 3, 2012

Simulation Driven Product Development with ANSYS Workbench platform

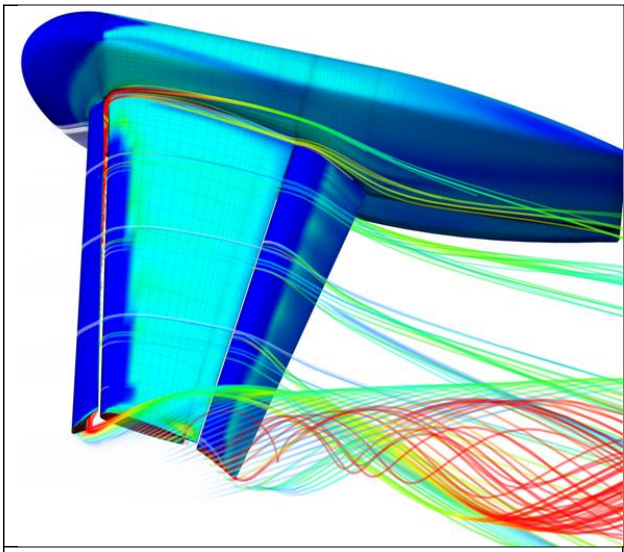
V. Morgenthaler
ANSYS France

In developing new products, organizations face complex and sometimes competing pressures like never before. It has to be affordable for customers, priced to turn a profit, and must work as promised. The challenges don’t stop there: Product lifecycles are shrinking, global competition is increasing and customer expectations are higher than ever.

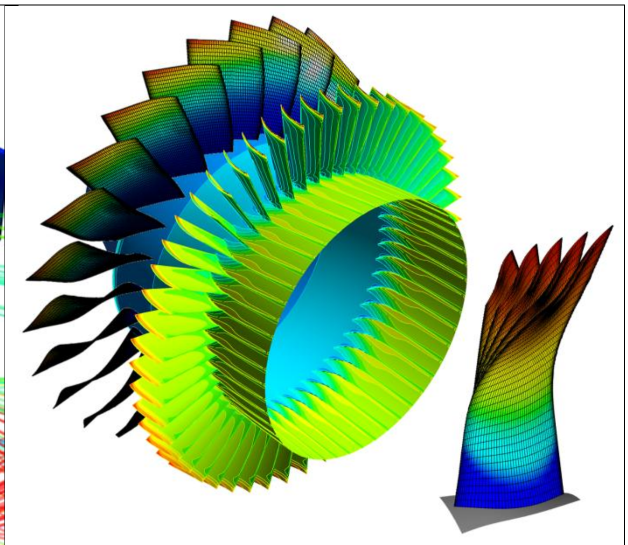
As products grow more complicated, the engineering challenges become harder to solve. And the methods and tools used to develop and test yesterday's products may no longer be applicable today.

Few product systems are affected by just one physical force. In the real world, products undergo a wide range of thermal, mechanical, electromagnetic and fluidic forces. Consequently, any system-level assessment must include multiphysics considerations to accurately predict performance.

This implies that the CFD workflow should be made effective and efficient for CFD simulations but also to perform multiphysic and multi-scale product design simulations.

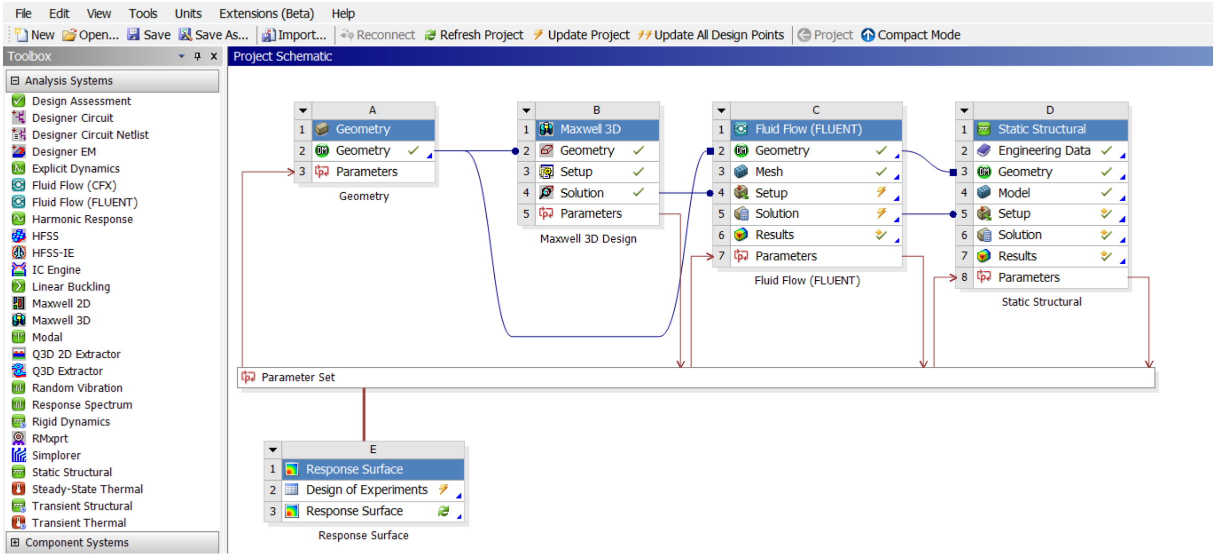


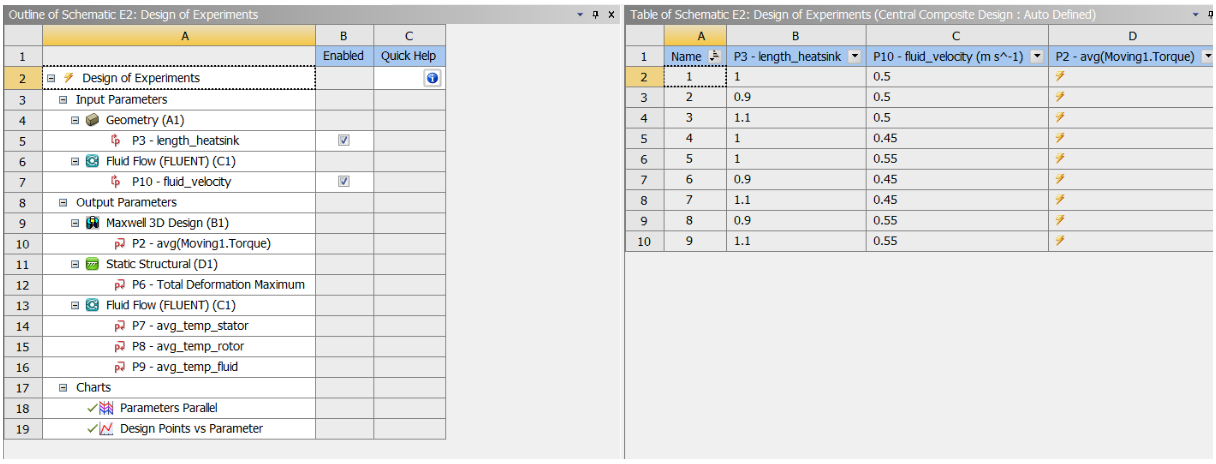
Trapezoidal (Trap) Wing Model



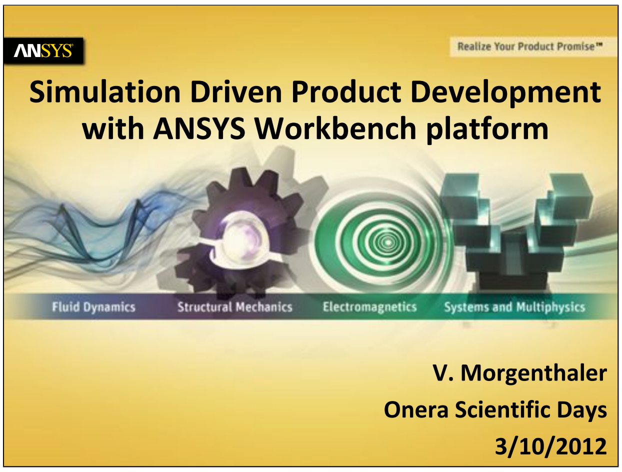
Aeroelastic simulation of air engine fan

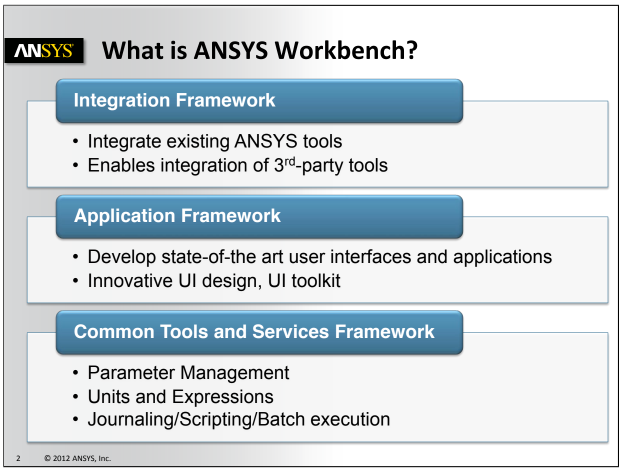
The ANSYS Workbench platform is the framework upon which the industry’s broadest and deepest suite of advanced engineering simulation technology is built. An innovative project schematic view ties together the entire simulation process, guiding the user through even complex multiphysics analyses. With bi-directional CAD connectivity, powerful highly-automated meshing, a project-level update mechanism, pervasive parameter management and integrated optimization tools the ANSYS Workbench platform delivers unprecedented productivity, enabling Simulation Driven Product Development.

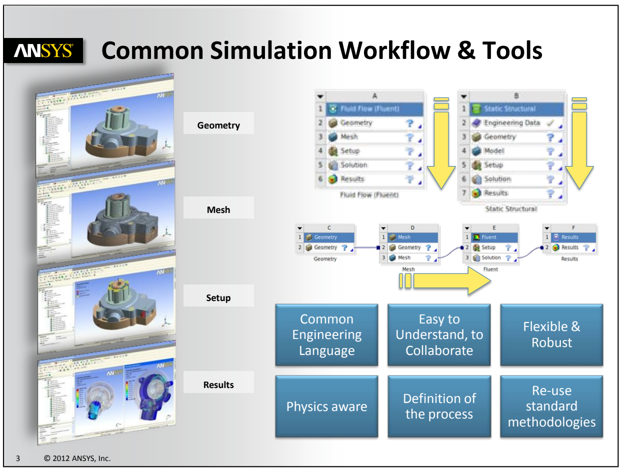


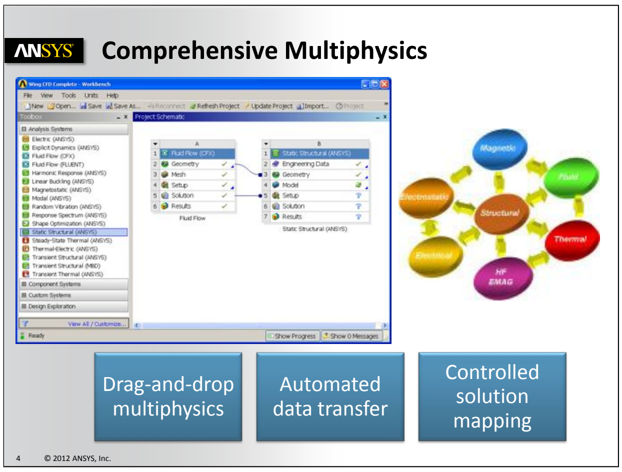


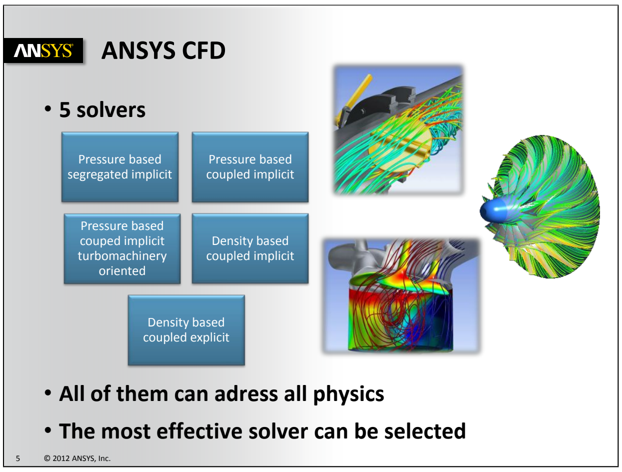
This presentation will first introduce the concept of ANSYS Workbench platform, then detail the CFD workflow with several key features and finally demonstrate how the CFD workflow fuel and fit into multiphysic, multiscale Simulation Driven Product Development.

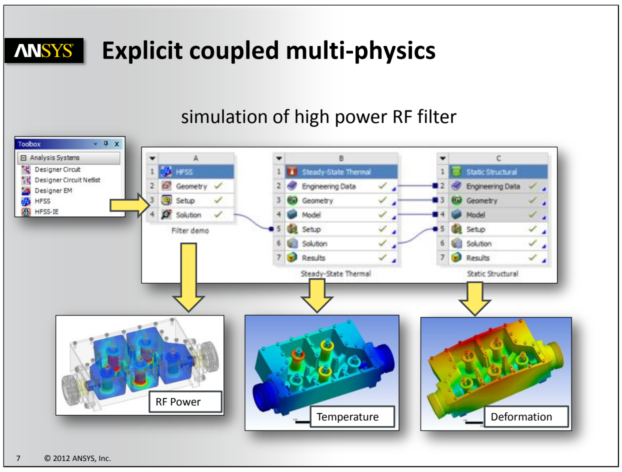

ANSYS
Realize Your Product Promise™
Simulation Driven Product Development with ANSYS Workbench platform
Fluid Dynamics
Structural Mechanics
Electromagnetics
Systems and Multiphysics
V. Morgenthaler
Onera Scientific Days
3/10/2012

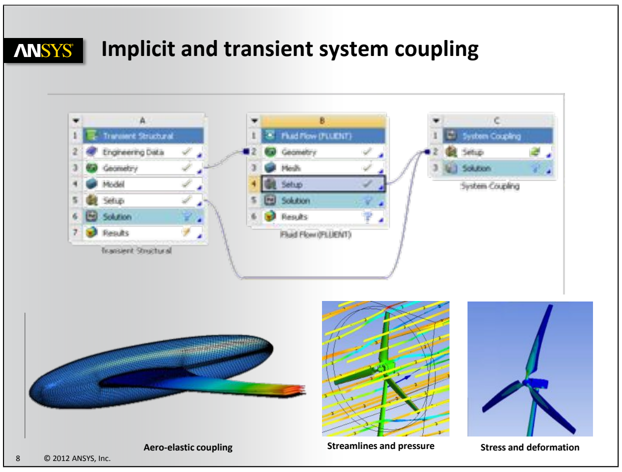

ANSYS
What is ANSYS Workbench?
Integration Framework
• Integrate existing ANSYS tools
• Enables integration of 3rd-party tools
Application Framework
• Develop state-of-the art user interfaces and applications
• Innovative UI design, UI toolkit
Common Tools and Services Framework
• Parameter Management
• Units and Expressions
• Journaling/Scripting/Batch execution
2 © 2012 ANSYS, Inc.

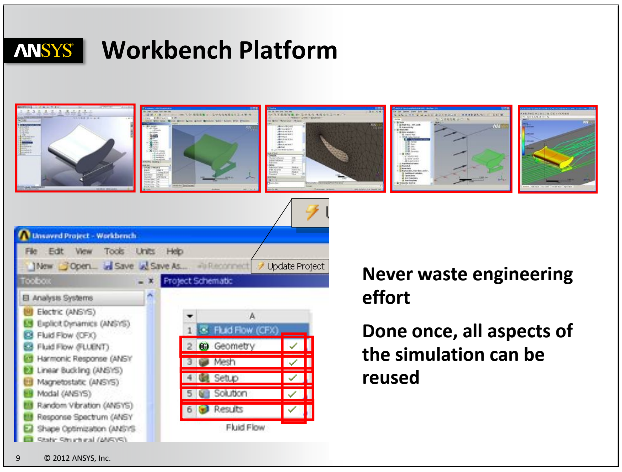

ANSYS
Common Simulation Workflow & Tools
Geometry
Mesh
Setup
Results
Common Engineering Language
Easy to Understand, to Collaborate
Flexible & Robust
Physics aware
Definition of the process
Re-use standard methodologies
3 © 2012 ANSYS, Inc.

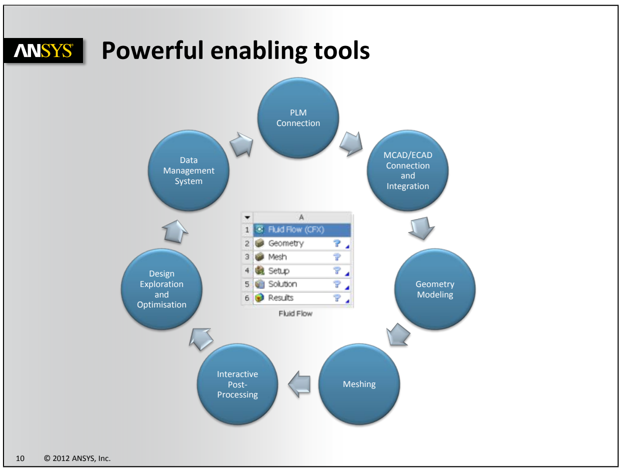

ANSYS
Comprehensive Multiphysics
Magnetic
Fluid
Electrostatic
Structural
Thermal
Electrical
HF EMAG
Drag-and-drop multiphysics
Automated data transfer
Controlled solution mapping
4 © 2012 ANSYS, Inc.

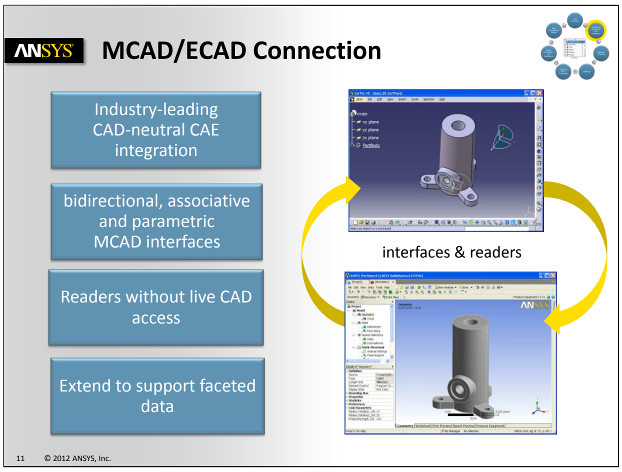

ANSYS
ANSYS CFD
• 5 solvers
Pressure based segregated implicit
Pressure based coupled implicit
Pressure based couped implicit turbomachinery oriented
Density based coupled implicit
Density based coupled explicit
• All of them can adress all physics
• The most effective solver can be selected
5 © 2012 ANSYS, Inc.

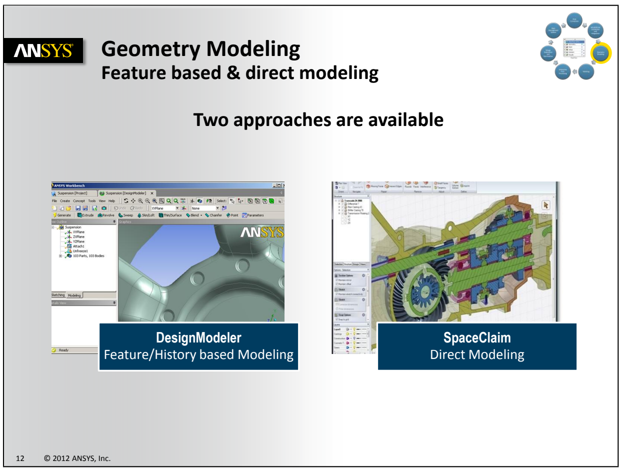

ANSYS
Explicit coupled multi-physics
simulation of high power RF filter
RF Power
Temperature
Deformation
7 © 2012 ANSYS, Inc.

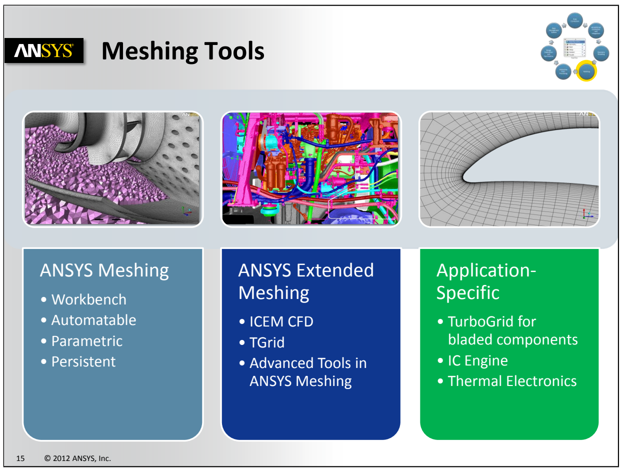

ANSYS
Implicit and transient system coupling
Transient Structural
Fluid Flow (FLUENT)
System Coupling
Aero-elastic coupling
Streamlines and pressure
Stress and deformation
8 © 2012 ANSYS, Inc.

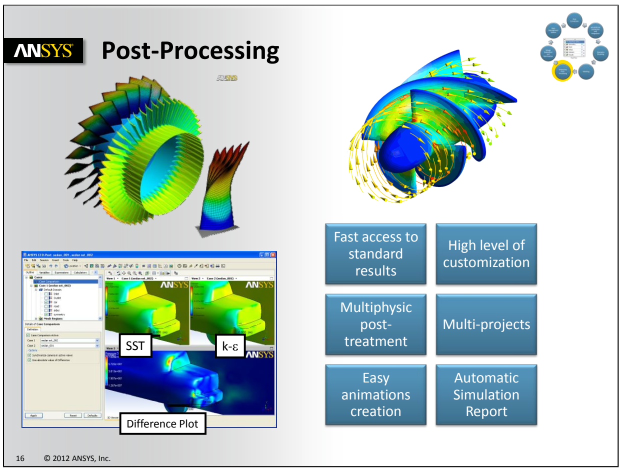

ANSYS
Workbench Platform
Never waste engineering effort
Done once, all aspects of the simulation can be reused
9 © 2012 ANSYS, Inc.

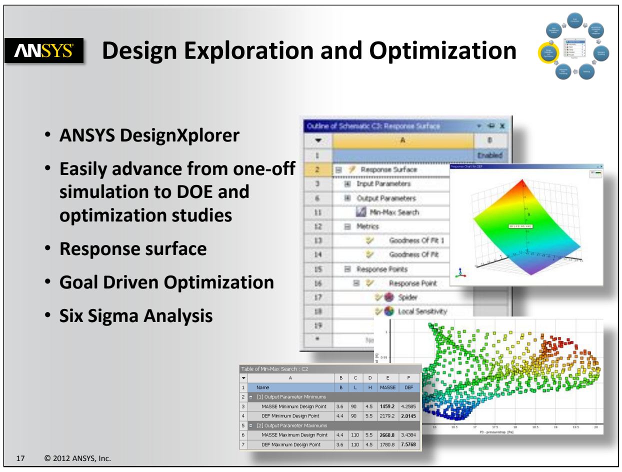

ANSYS
Powerful enabling tools
PLM Connection
MCAD/ECAD Connection and Integration
Geometry Modeling
Meshing
Interactive Post-Processing
Design Exploration and Optimisation
Data Management System
Fluid Flow (CFX)
Geometry
Mesh
Setup
Solution
Results
Fluid Flow
10 © 2012 ANSYS, Inc.

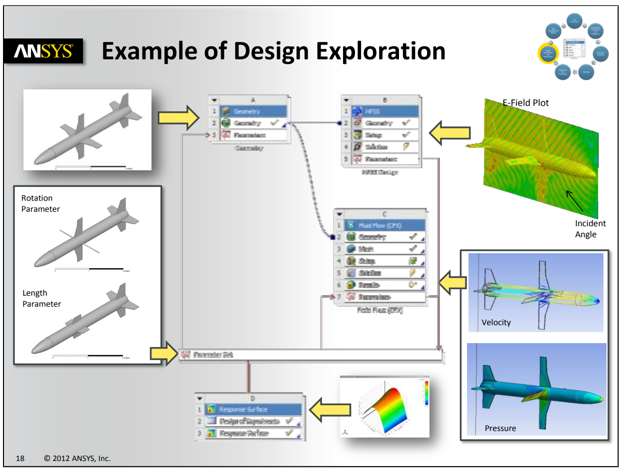

ANSYS
MCAD/ECAD Connection
Industry-leading CAD-neutral CAE integration
bidirectional, associative and parametric MCAD interfaces
Readers without live CAD access
Extend to support faceted data
interfaces & readers
11 © 2012 ANSYS, Inc.

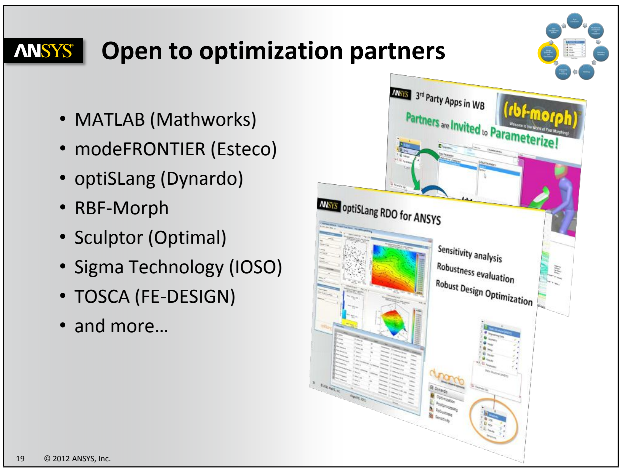

ANSYS
Geometry Modeling
Feature based & direct modeling
Two approaches are available
DesignModeler
Feature/History based Modeling
SpaceClaim
Direct Modeling
12 © 2012 ANSYS, Inc.


ANSYS
Meshing Tools
ANSYS Meshing
• Workbench
• Automatable
• Parametric
• Persistent
ANSYS Extended Meshing
• ICEM CFD
• TGrid
• Advanced Tools in ANSYS Meshing
Application-Specific
• TurboGrid for bladed components
• IC Engine
• Thermal Electronics
15 © 2012 ANSYS, Inc.

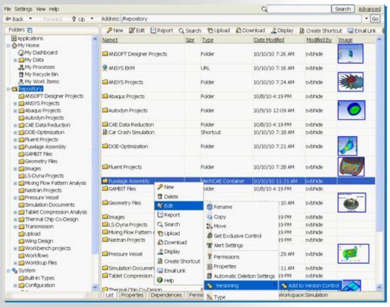

ANSYS
Post-Processing
Fast access to standard results
High level of customization
Multiphysic post-treatment
Multi-projects
Easy animations creation
Automatic Simulation Report
SST
k-ε
Difference Plot
16 © 2012 ANSYS, Inc.


ANSYS
Design Exploration and Optimization
• ANSYS DesignXplorer
• Easily advance from one-off simulation to DOE and optimization studies
• Response surface
• Goal Driven Optimization
• Six Sigma Analysis
17 © 2012 ANSYS, Inc.


ANSYS
Example of Design Exploration
Rotation Parameter
Length Parameter
E-Field Plot
Incident Angle
Velocity
Pressure
18 © 2012 ANSYS, Inc.


ANSYS
Open to optimization partners
• MATLAB (Mathworks)
• modeFRONTIER (Esteco)
• optiSLang (Dynardo)
• RBF-Morph
• Sculptor (Optimal)
• Sigma Technology (IOSO)
• TOSCA (FE-DESIGN)
• and more…
3rd Party Apps in WB
Partners are Invited to Parameterize!
optiSLang RDO for ANSYS
Sensitivity analysis
Robustness evaluation
Robust Design Optimization
19 © 2012 ANSYS, Inc.

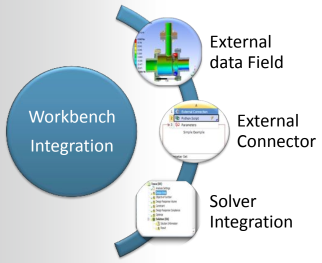
Knowledge Management and Collaborative plateform



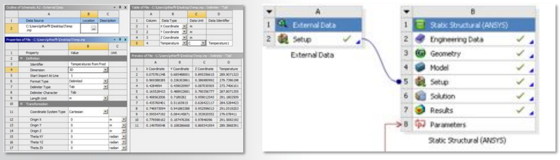
EKM

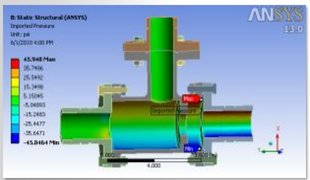
Link with PLM

External Data or In-House code Integration



External data fields





External Connector

Application launcher

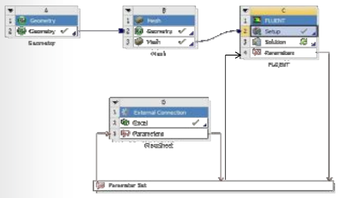
- OS executable command to launch the external application
- Definition of the command is done through a python script

Data exchange

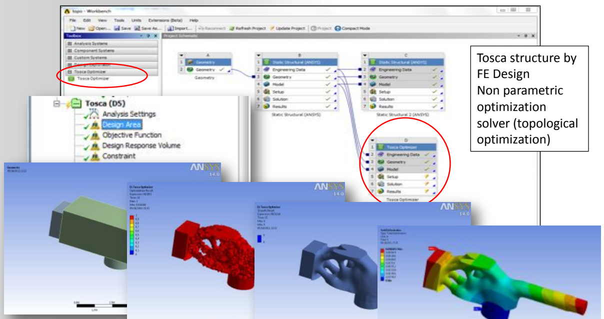
- Data passed via input and output files
- Input and output defined in an xml description file

Parameter setup

- Parameter echanged using files



In-house Solver Integration



Initial design

Optimal result

Optimal design

Verification

Shaping the future of simulation

ANSYS Workbench fully integrates ANSYS CFD Workflow but also enables :

- **Workflow Customization**
- **Multiphysics simulation (Structural Mechanics – Low & High Frequency Electromagnetism)**
- **Design Exploration and Optimization**
- **Simulation data Management and Capture Best Practices**
- **Clustering and HPC**

Our produts are following ANSYS Simulation Driven Product Development Strategy and accelerate Innovation.

2.8 C. Hirsch (Numeca) – presentation cancelled –

ONERA Scientific Day
CFD Workflow: Meshing, Solving, Visualizing ...

Components for an Integrated CFD Workflow for large scale Multidisciplinary Simulations

Charles Hirsch
Prof. Em. Vrije Universiteit Brussel
President, NUMECA Int.

ABSTRACT

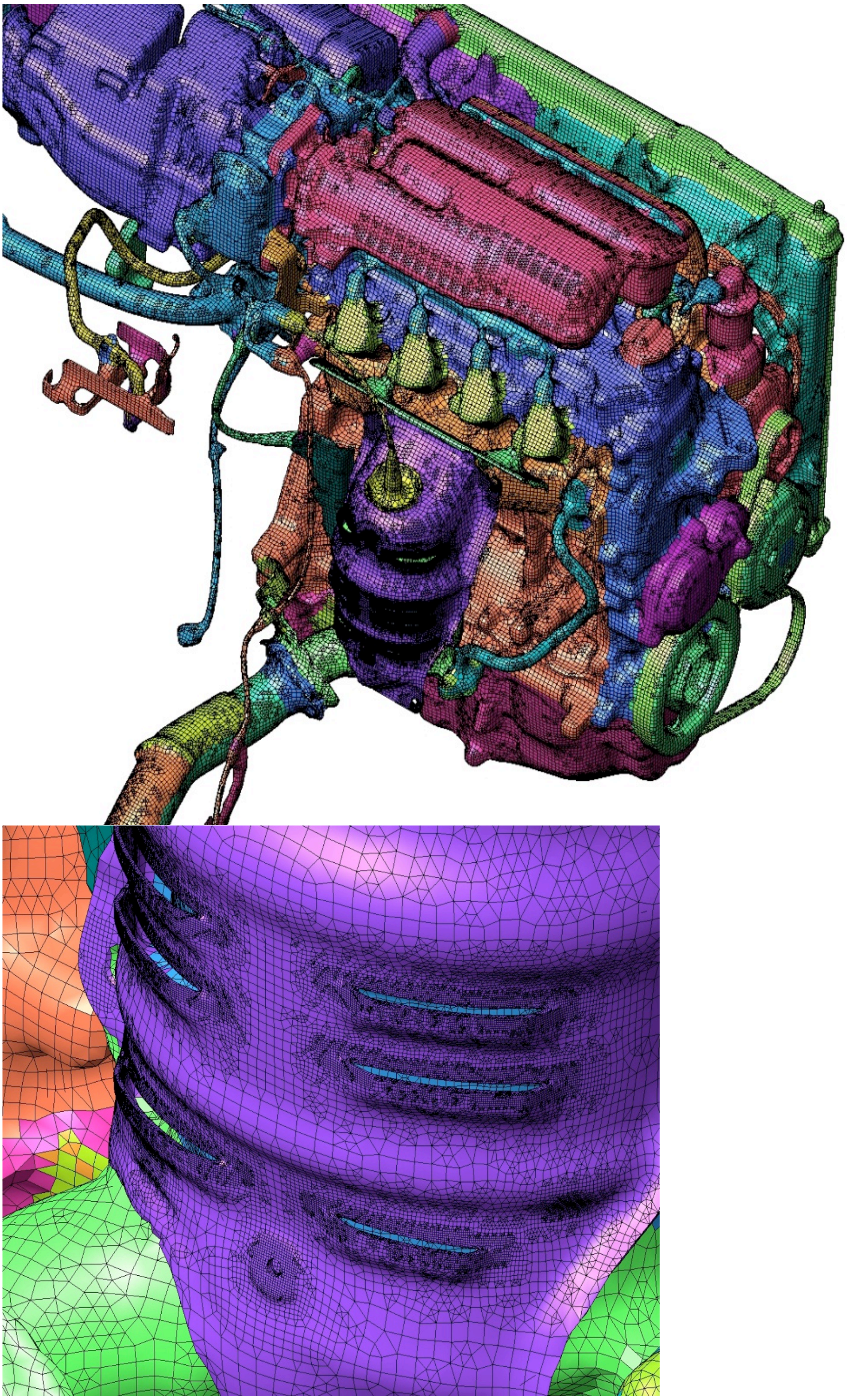
The current evolution in the aeronautical field towards high-fidelity simulations, including multi-physics, calls for a new approach of the complete CFD-multi-physics analysis and subsequent optimization chain.

High-fidelity simulations imply indeed a higher level of description of geometrical features, leading to very large meshes; to more accurate physics, such as required by broadband noise estimations, or fluid-structure couplings for aero-elastic stability. This leads to problem settings with large grids and large CPU times, even on large-scale parallel computers. In addition, pre-processing of large grids or post-processing of terabytes of data, such as resulting from large scale LES simulations, pose new challenges to the simulation chain.

A main objective of computer aided engineering (CAE) and virtual prototyping (VP), associated to multidisciplinary optimization, is the drastic reduction of the turnaround time needed for a given simulation. This is essential, since any design has to be achieved within a fixed, often quite restricted time framework and with a shorter turnaround time, more trials can investigated in the design space, increasing hereby the likelihood of coming up with a better final design.

This objective covers all aspects of the CFD Workflow: automatic high quality mesh generation, including CAD cleaning; very fast solvers; multiphysics analysis and optimization; highly efficient post-processing for large data sets.

Several of these components will be presented.



Example of high-fidelity grid generation with Hexpress/Hybrid™

2.9 C. Geuzaine (Univ. Liège)

Abstract: Onera Scientific Day 2012

Recent Advances in Quad Meshing

C. Geuzaine
Université de Liège, Belgium

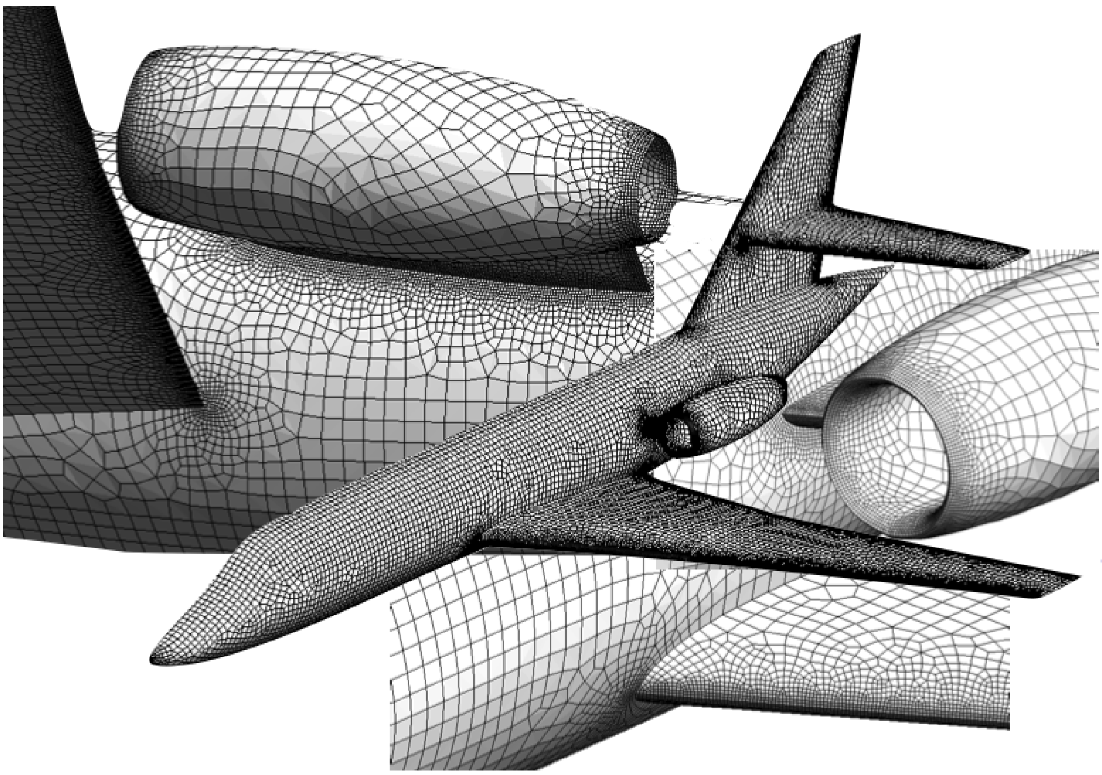
J.-F. Remacle, E. Marchandise
Université Catholique de Louvain, Belgium

There exist essentially two approaches to automatically generate quadrilateral finite element meshes. With *direct methods*, quadrilaterals are constructed at once, using either advancing front techniques [1] or regular grid-based methods (quadtrees) [2]. *Indirect methods*, on the other hand, rely on an initial triangular mesh and apply merging techniques to recombine the triangles of the initial mesh into quadrangles [3, 4]. Other more sophisticated indirect methods use a mix of advancing front and recombination [5].

In this talk we will report on our recent efforts [6, 7] to design an indirect quad meshing algorithm that can produce high-quality quadrangular meshes, with speed and robustess that approach those of classical triangulation algorithms. The resulting algorithm is readily available for testing in the open source mesh generator Gmsh [8].

The talk will focus on the following elements:

- Reparametrization of surfaces and construction of cross fields [9, 10];
- Generation of triangular meshes in the L^∞ norm, suitable for recombination into quads [7];
- Recombination of triangular meshes into fully quadrangular meshes using graph-theoretical approaches [6].



References

[1] T. D. Blacker and M. B. Stephenson. Paving: A new approach to automated quadrilateral mesh generation. *International Journal for Numerical Methods in Engineering*, 32:811–847, 1991.

[2] P.J. Frey and L. Marechal. Fast adaptive quadtree mesh generation. In *in: Proceedings of the Seventh International Meshing Roundtable*. Citeseer, 1998.

[3] C. K. Lee and S. H. Lo. A new scheme for the generation of a graded quadrilateral mesh. *Computers and Structures*, 52:847–857, 1994.

[4] H. Borouchaki and P.J. Frey. Adaptive triangular–quadrilateral mesh generation. *International Journal for Numerical Methods in Engineering*, 45(5):915–934, 1998.

[5] S. J. Owen, M. L. Staten, S. A. Canann, and S. Saigal. Q-morph: An indirect approach to advancing front quad meshing. *International Journal for Numerical Methods in Engineering*, 9:1317–1340, 1999.

[6] J.-F. Remacle, J. Lambrechts, B. Seny, E. Marchandise, A. Johnen, and C. Geuzaine. Blossom-quad: a non-uniform quadrilateral mesh generator using a minimum cost perfect matching algorithm. 89:1102–1119, 2012. doi: 10.1002/nme.3279.

[7] J. F. Remacle, F. Henrotte, T. Carrier-Baudouin, E. Béchet, E. Marchandise, C. Geuzaine, and T. Mouton. A frontal Delaunay quad mesh generator using the l^∞ norm. *International Journal for Numerical Methods in Engineering*, In press, 2012.

[8] C. Geuzaine and J.-F. Remacle. Gmsh: a three-dimensional finite element mesh generator with built-in pre- and post-processing facilities. *International Journal for Numerical Methods in Engineering*, 79(11):1309–1331, 2009.

[9] E. Marchandise, C.C. de Wiart, WG Vos, C. Geuzaine, and J.F. Remacle. High-quality surface remeshing using harmonic maps–part ii: Surfaces with high genus and of large aspect ratio. *International Journal for Numerical Methods in Engineering*, 86:1303–1321, 2011.

[10] B. Lévy and Y. Liu. Lp centroidal voronoi tesselation and its applications. In *ACM Transactions on Graphics (SIGGRAPH conference proceedings)*, 2010.

Recent Advances in Quad (Re)Meshing

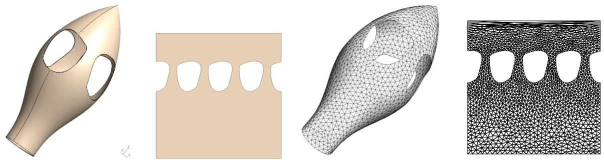
C. Geuzaine
University of Liège
Dept. of Electrical Engineering and Computer Science
Institut Montefiore B28

October 3, 2012

Surface meshes

Surface triangulations can be generated

- either directly in the embedding 3-D Euclidean space
- or in the parametric plane of the surface, which is far more robust (Delaunay and variants)



Unfortunately...

In many cases we don't have a (good) parametrization of the surfaces

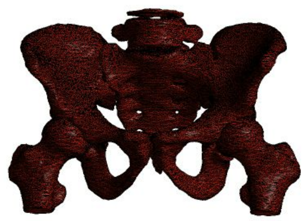
Example 1: biomedical simulations

- Require high-quality meshes

Unfortunately...

In many cases we don't have a (good) parametrization of the surfaces

Example 1: biomedical simulations

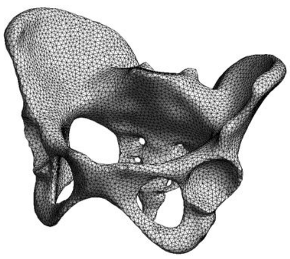


- Require high-quality meshes
- Triangulations (STL) obtained from imaging techniques are of low quality:
  - oversampled
  - non-Delaunay

Unfortunately...

In many cases we don't have a (good) parametrization of the surfaces

Example 1: biomedical simulations

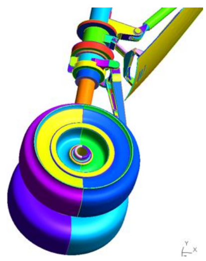


- Require high-quality meshes
- Triangulations (STL) obtained from imaging techniques are of low quality:
  - oversampled
  - non-Delaunay
- Recover high quality surface mesh from low-quality inputs

Unfortunately...

In many cases we don't have a (good) parametrization of the surfaces

Example 2: CAD data

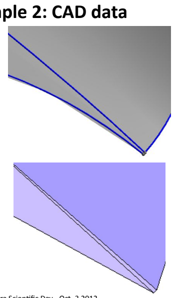


- Geometric models contain many patches (e.g. 852 in CATIA model of landing gear)

Unfortunately...

In many cases we don't have a (good) parametrization of the surfaces

Example 2: CAD data

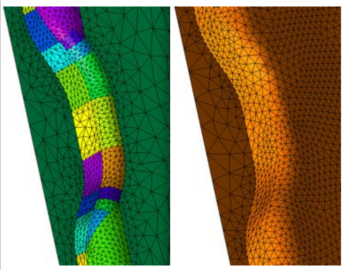


- Geometric models contain many patches (e.g. 852 in CATIA model of landing gear)
- Not suitable for FE analysis:
  - small model edges/faces
  - lead to poor quality triangles

Unfortunately...

In many cases we don't have a (good) parametrization of the surfaces

Example 2: CAD data

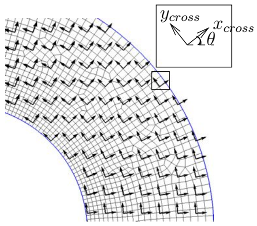


- Geometric models contain many patches (e.g. 852 in CATIA model of landing gear)
- Not suitable for FE analysis:
  - small model edges/faces
  - lead to poor quality triangles
- Mesh group of patches instead of single patches

Unfortunately...

In many cases we don't have a (good) parametrization of the surfaces

Example 3: Quad meshes need orientation information



- Contrary to triangles, quads are intrinsically "oriented" (they follow some main "directions")

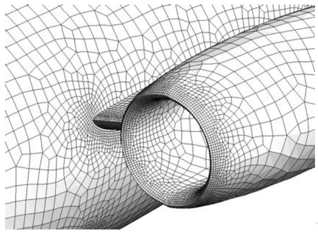
C. Geuzaine - Onera Scientific Day - Oct. 3 2012 5

jeudi 4 octobre 12

Unfortunately...

In many cases we don't have a (good) parametrization of the surfaces

Example 3: Quad meshes need orientation information



- Contrary to triangles, quads are intrinsically "oriented" (they follow some main "directions")
- If meshing in parameter space, need conformal mapping that preserves angles

C. Geuzaine - Onera Scientific Day - Oct. 3 2012 5

jeudi 4 octobre 12

In this talk

Remesh using **triangles** and **quads** by solving **PDEs** to compute a **conformal parametrization** on an **initial mesh**

Joint work with:

- J.-F. Remacle
- E. Marchandise
- E. Bechet
- T. Mouton
- T. Carrier-Baudouin
- A. Johnen
- E. Sauvage

C. Geuzaine - Onera Scientific Day - Oct. 3 2012 6

jeudi 4 octobre 12

In this talk

Remesh using **triangles** and **quads** by solving **PDEs** to compute a **conformal parametrization** on an **initial mesh**

Related papers:

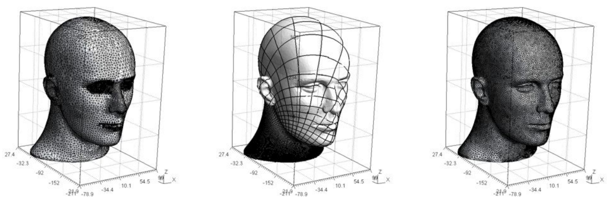
- *Gmsh: a finite element mesh generator with built-in pre- and post-processing facilities*, International Journal for Numerical Methods in Engineering, 79(11), 130901331, 2009.
- *High-Quality Surface Remeshing Using Harmonic Maps*, International Journal for Numerical Methods in Engineering, 83(4), pp. 403–425, 2010.
- *Quality meshing based on STL triangulations for biomedical simulations*. International Journal for Numerical Methods in Biomedical Engineering, 26(7), pp. 876–889, 2010.
- *High Quality Surface Remeshing Using Harmonic Maps. Part II: Surfaces with High Genus and of Large Aspect Ratio*. International Journal for Numerical Methods in Engineering, 86(11), 1303-1321, 2011.
- *Quality open source mesh generation for cardiovascular flow simulations*, Springer Series on Modeling, Simulation and Applications, to appear, 2011.
- *Blossom-Quad: a non-uniform quadrilateral mesh generator using a minimum cost perfect matching algorithm*, International Journal for Numerical Methods in Engineering, 2011.
- *A frontal Delaunay quad mesh generator using the L^∞ norm*, International Journal for Numerical Methods in Engineering, 2012.

C. Geuzaine - Onera Scientific Day - Oct. 3 2012 6

jeudi 4 octobre 12

The basic idea

(1) (2) (3)

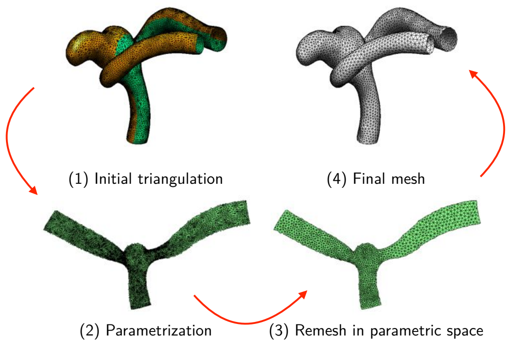


1. Start from an initial triangulation
2. Compute an adequate mapping
3. Use robust planar algorithms to build a finite element mesh

C. Geuzaine - Onera Scientific Day - Oct. 3 2012 7

jeudi 4 octobre 12

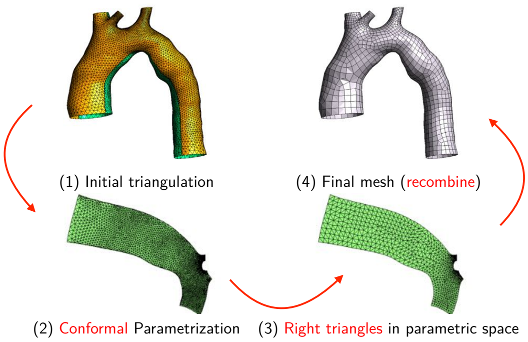
Triangular meshing with parametrization



C. Geuzaine - Onera Scientific Day - Oct. 3 2012 8

jeudi 4 octobre 12

Quadrangular meshing with parametrization



C. Geuzaine - Onera Scientific Day - Oct. 3 2012 9

jeudi 4 octobre 12

Outline

- Computing maps
- Automatic triangular remeshing
- Automatic quadrangular remeshing

C. Geuzaine - Onera Scientific Day - Oct. 3 2012 10

jeudi 4 octobre 12

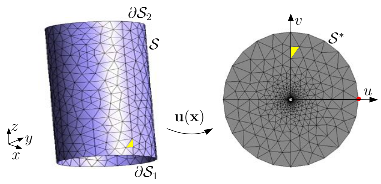
Outline

- Computing maps
- Automatic triangular remeshing
- Automatic quadrangular remeshing

Reparametrization of triangulated surfaces

A reparametrization procedure aims at finding an diffeomorphism that maps a complex triangulated surface $\mathcal{S}$ into another one $\mathcal{S}^*$ that has a well-known parametrization (e.g. the plane):

$$\mathbf{x} \in \mathcal{S} \subset \mathcal{R}^3 \mapsto \mathbf{u}(\mathbf{x}) \in \mathcal{S}^* \subset \mathcal{R}^2$$



Compute $\mathbf{u}(\mathbf{x})$ with conformal harmonic map

Minimize the conformal energy:

$$E_{\mathrm{LSCM}}(\mathbf{u}) = \int_{\mathcal{S}} \frac{1}{2} \left| \nabla u^{\perp} - \nabla v \right|^2 ds,$$

where $^{\perp}$ denotes a counterclockwise $90°$ rotation in $\mathcal{S}$

The discrete minimization problem is equivalent to solving

$$\underbrace{\begin{pmatrix} \bar{\bar{A}} & \bar{\bar{C}} \\ \bar{\bar{C}}^T & \bar{\bar{A}} \end{pmatrix}}_{L_C} \begin{pmatrix} \bar{U} \\ \bar{V} \end{pmatrix} = \begin{pmatrix} \bar{0} \\ \bar{0} \end{pmatrix}$$

where $\bar{\bar{A}}$ is a symmetric positive definite matrix $A_{kj} = \int_{\mathcal{S}} \nabla\phi_k \cdot \nabla\phi_j \, ds$

and $\bar{\bar{C}}$ is an antisymmetric matrix $C_{kj} = \int_{\mathcal{S}} \mathbf{n} \cdot (\nabla\phi_k \times \nabla\phi_j) \, ds$

Compute $\mathbf{u}(\mathbf{x})$ with conformal harmonic map

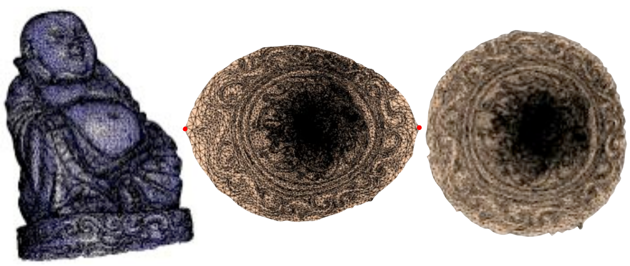
How to assign proper boundary conditions?

1. pin down two vertices (Levy 2001)
2. constrain the system and solve it using spectral theory (Mullen 2008)

The eigenvector $\mathbf{u}^*$ (Fiedler vector) associated with the second smallest eigenvalue λ of $L_C\mathbf{u}^* = \lambda\mathbf{u}^*$ is solution to the constrained minimization problem:

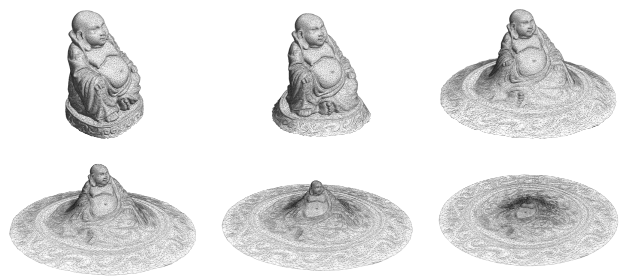
$$\mathbf{u}^* = \underset{\mathbf{u}, \mathbf{u}^t\mathbf{e}=0, \mathbf{u}^t\mathbf{u}=1}{\arg\min} \; \mathbf{u}^t L_C \mathbf{u}$$

Compute $\mathbf{u}(\mathbf{x})$ with conformal harmonic map



Pinned down Spectral

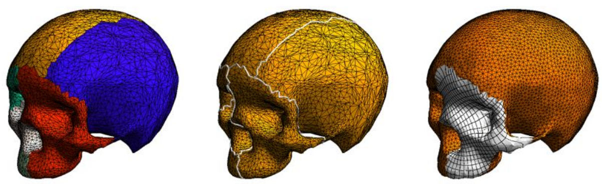
Compute $\mathbf{u}(\mathbf{x})$ with Laplacian harmonic map



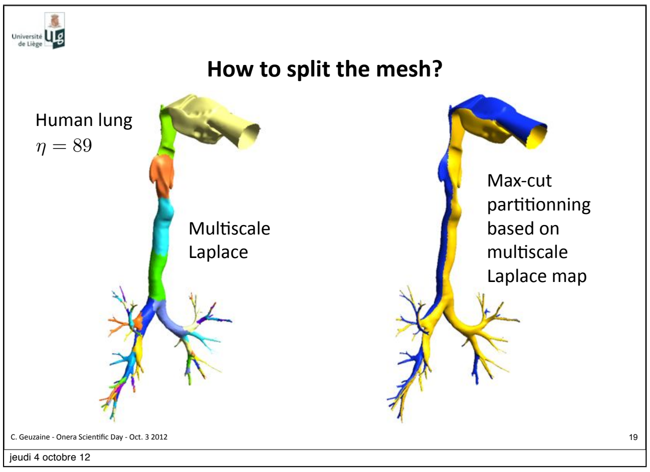
Outline

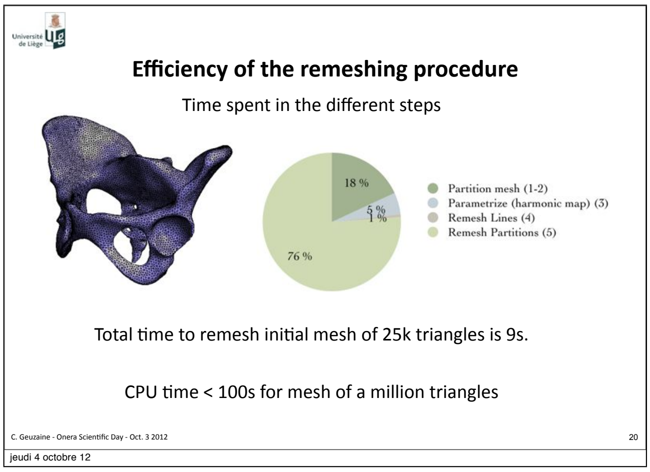
- Computing maps
- Automatic triangular remeshing
- Automatic quadrangular remeshing

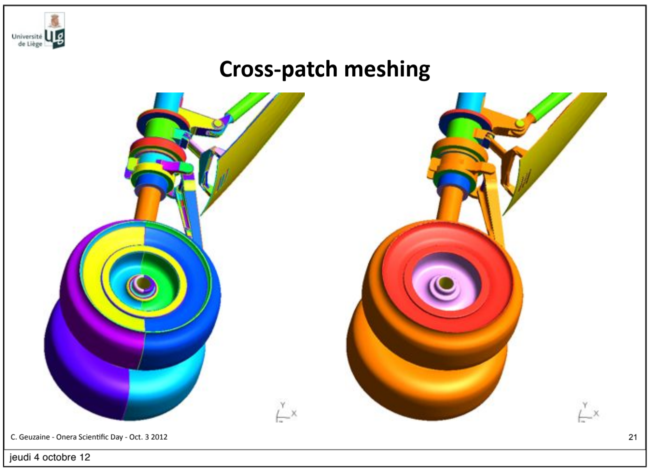
Automatic remeshing algorithm

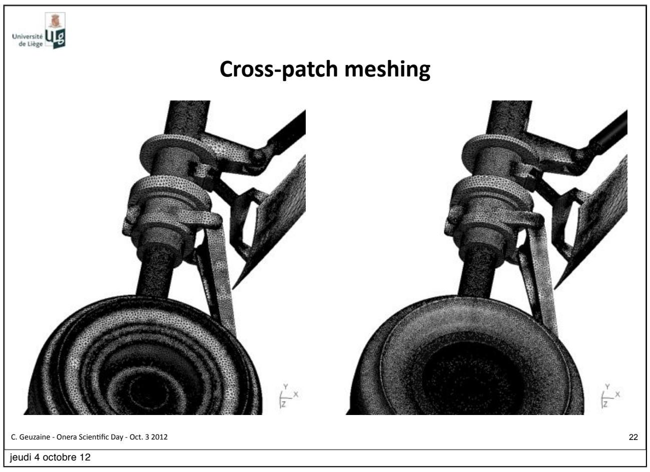


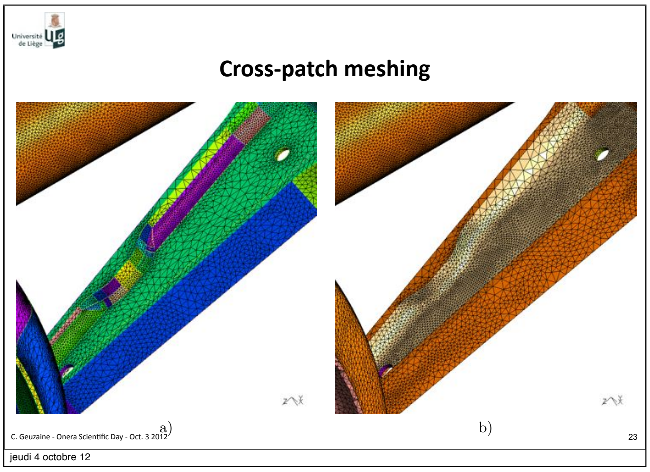
1. Compute $G = 2, N_B = 0, \eta|_{G=0}$
2. If needed split/cut mesh into different partitions of zero genus
3. Remesh the lines at the interfaces between partitions
4. Compute mapping for every partition
5. Remesh the partition in the parametric space

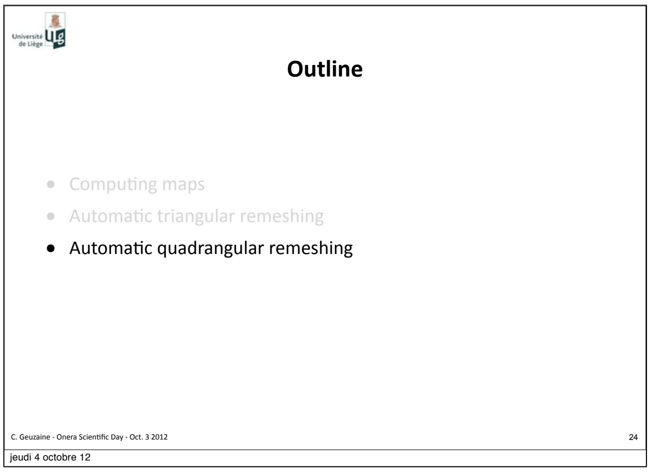

How to split the mesh?
Human lung
$\eta = 89$
Multiscale
Laplace
Max-cut
partitionning
based on
multiscale
Laplace map
C. Geuzaine - Onera Scientific Day - Oct. 3 2012
19
jeudi 4 octobre 12

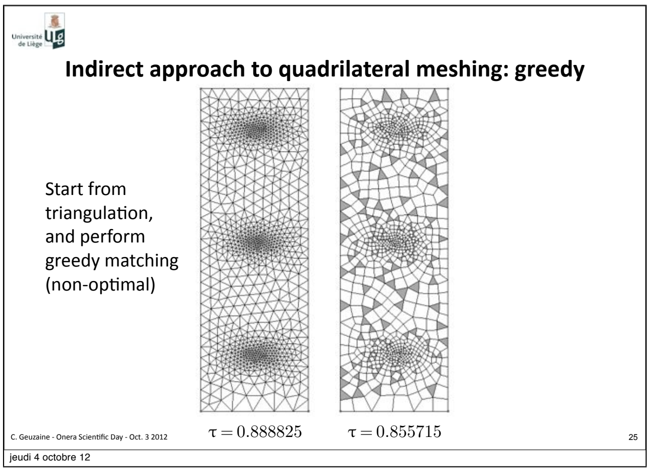

Efficiency of the remeshing procedure
Time spent in the different steps
18 %
5 %
1 %
76 %
Partition mesh (1-2)
Parametrize (harmonic map) (3)
Remesh Lines (4)
Remesh Partitions (5)
Total time to remesh initial mesh of 25k triangles is 9s.
CPU time < 100s for mesh of a million triangles
C. Geuzaine - Onera Scientific Day - Oct. 3 2012
20
jeudi 4 octobre 12

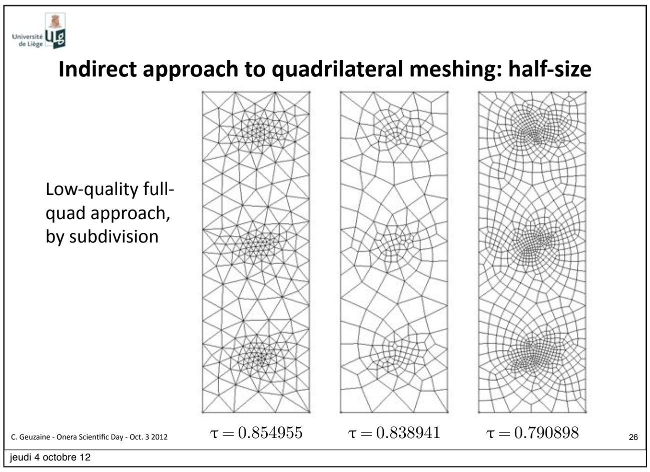

Cross-patch meshing
C. Geuzaine - Onera Scientific Day - Oct. 3 2012
21
jeudi 4 octobre 12


Cross-patch meshing
C. Geuzaine - Onera Scientific Day - Oct. 3 2012
22
jeudi 4 octobre 12


Cross-patch meshing
a)
b)
C. Geuzaine - Onera Scientific Day - Oct. 3 2012
23
jeudi 4 octobre 12

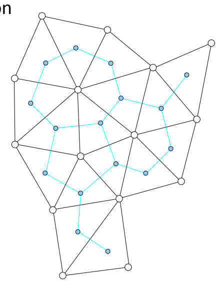

Outline
• Computing maps
• Automatic triangular remeshing
• Automatic quadrangular remeshing
C. Geuzaine - Onera Scientific Day - Oct. 3 2012
24
jeudi 4 octobre 12


Indirect approach to quadrilateral meshing: greedy
Start from
triangulation,
and perform
greedy matching
(non-optimal)
$\tau = 0.888825$
$\tau = 0.855715$
C. Geuzaine - Onera Scientific Day - Oct. 3 2012
25
jeudi 4 octobre 12


Indirect approach to quadrilateral meshing: half-size
Low-quality full-
quad approach,
by subdivision
$\tau = 0.854955$
$\tau = 0.838941$
$\tau = 0.790898$
C. Geuzaine - Onera Scientific Day - Oct. 3 2012
26
jeudi 4 octobre 12

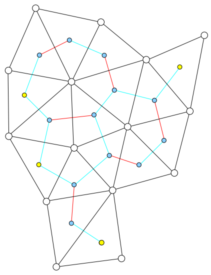
Indirect approach to quadrilateral meshing: Blossom

Build $G(V, E, c)$ an undirected weighted graph, with V the set of n_V vertices, E the set of n_E undirected edges and $c(E) = \sum c(e_{ij})$ an edge-based cost function



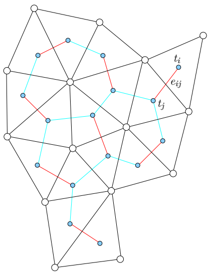
Indirect approach to quadrilateral meshing: Blossom

A matching is a subset $E' \subseteq E$ such that each node of V has at most one incident edge in E'



Indirect approach to quadrilateral meshing: Blossom

A matching is perfect if each node of V has exactly one incident edge in E'. A matching is optimum if $c(E')$ is minimum among all possible perfect matchings



Indirect approach to quadrilateral meshing: Blossom

The Blossom algorithm (Edmonds 1965) solves the problem of optimum perfect matching in polynomial time $\mathcal{O}(n_V^2 n_E)$

The worst-case complexity of Blossom has been steadily improving since then: Lawler and Gabow achieved $\mathcal{O}(n_V^3)$, Galil, Micali and Gabow improved it to $\mathcal{O}(n_V n_E \log(n_V))$. The current best known result in terms of n_V and n_E is $\mathcal{O}(n_V(n_E + \log(n_V)))$

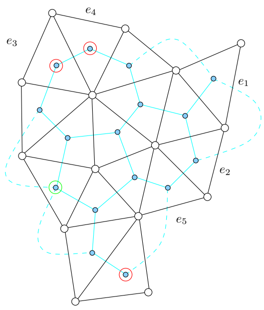
There is also a long history of computer implementations of the Blossom algorithm, starting with the Blossom I code of Edmonds, Johnson and Lockhart. We use the Blossom IV code of Cook and Rohe, that has been considered for several years as the fastest available (http://www2.isye.gatech.edu/~wcook/blossom4/)

Indirect approach to quadrilateral meshing: Blossom

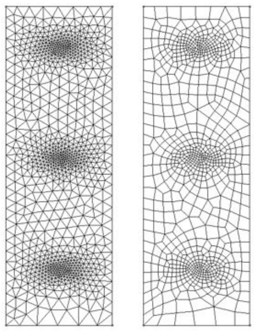
Finding if a perfect matching **exists** is (sharp) NP complete

However, a perfect matching is guaranteed to exist for cubic graphs

Engineer's solution: add extra edges to make our graphs closer to cubic

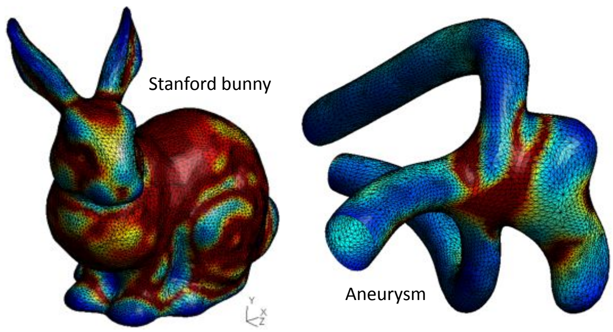


Indirect approach to quadrilateral meshing: Blossom

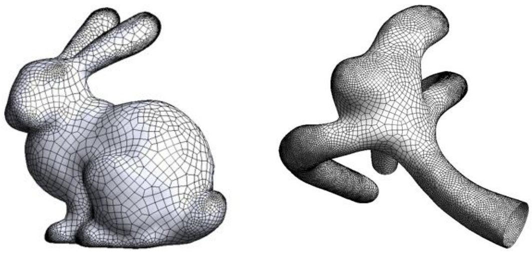


$\tau = 0.888825$ $\tau = 0.839313$

Reparametrization + Blossom



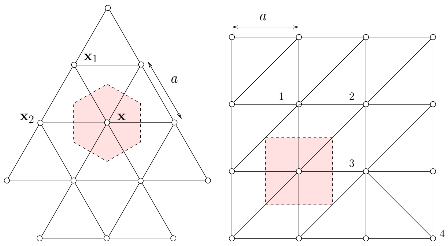
Reparametrization + Blossom



$\tau = 0.842549$ $\tau = 0.856247$

CPU time: approx. 20s (5s for Blossom)

Major problem for indirect approach

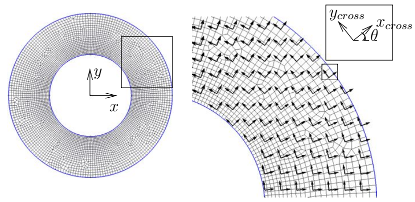


A "perfect" equilateral triangular mesh contains about $2/\sqrt{3}$ times too many vertices to make a "perfect" quadrilateral mesh

Idea: mesh generation in L^∞ norm

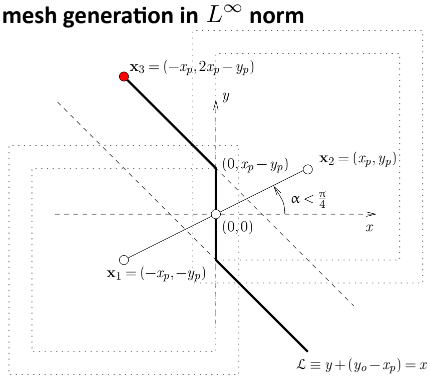
The L^∞ norm is not invariant by rotation $\Rightarrow$ quads should be oriented

We compute a cross-field to give the orientation of a local system of axes in the parameter plane. The local value of the cross-field is defined up to rotations by $\pi/2 \Rightarrow$ propagate $\alpha(u,v) = a(u,v) + ib(u,v) = e^{4i\theta(u,v)}$ through a harmonic map



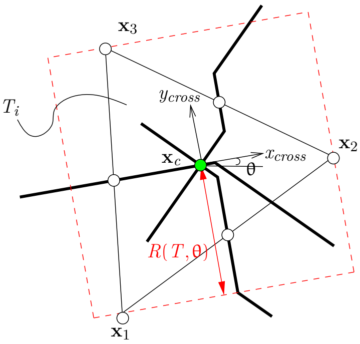
Idea: mesh generation in L^∞ norm

Perpendicular bisectors in the L^∞ norm



Idea: mesh generation in L^∞ norm

Circumcenter, circumradius and circumcircle using the L^∞ norm



Frontal-Delaunay in L^∞ norm: DelQuad

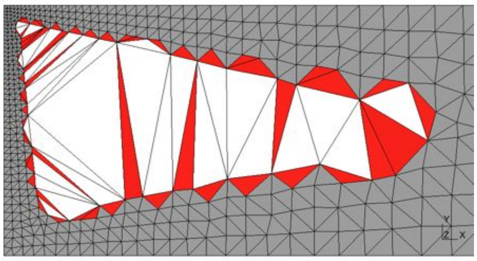


Illustration of the frontal-Delaunay: Grey triangles are "resolved", red triangles are "active", white triangles are "waiting"

Frontal-Delaunay in L^∞ norm: DelQuad

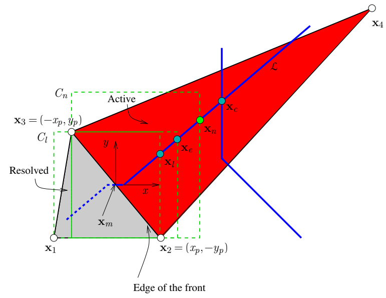
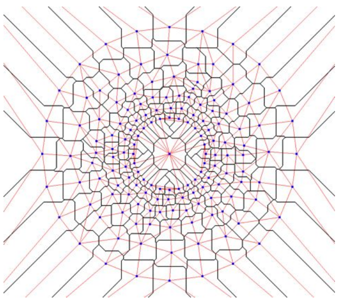
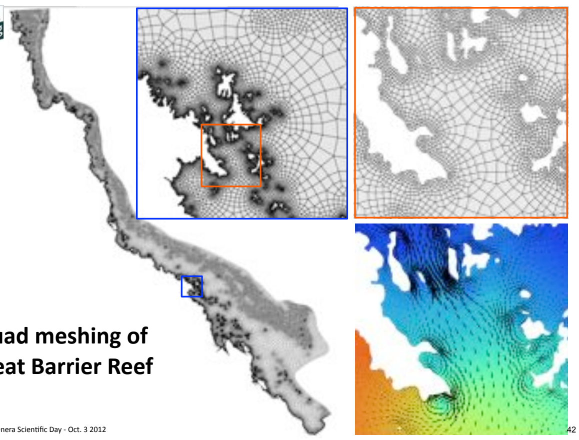


Illustration of the point insertion algorithm

Frontal-Delaunay in L^∞ norm: DelQuad

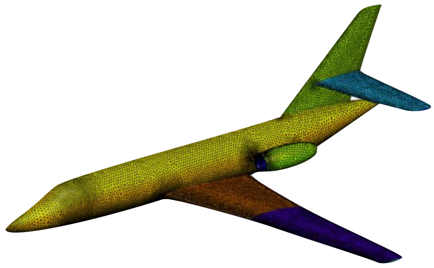


Voronoi diagram and Delaunay triangulation in the L^∞ norm



Quad meshing of Great Barrier Reef

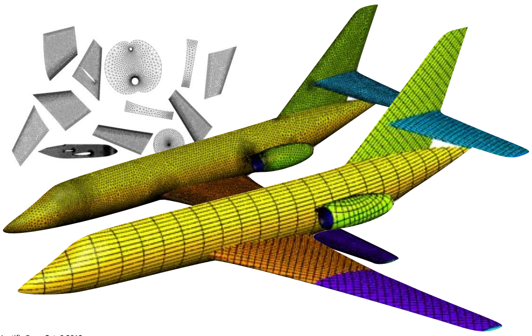
Quad meshing of Falcon



C. Geuzaine - Onera Scientific Day - Oct. 3 2012 43

jeudi 4 octobre 12

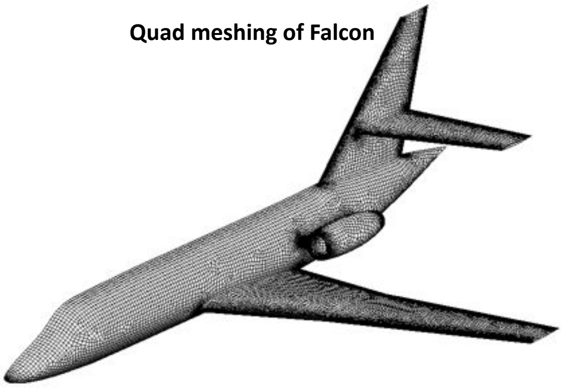
Quad meshing of Falcon



C. Geuzaine - Onera Scientific Day - Oct. 3 2012 43

jeudi 4 octobre 12

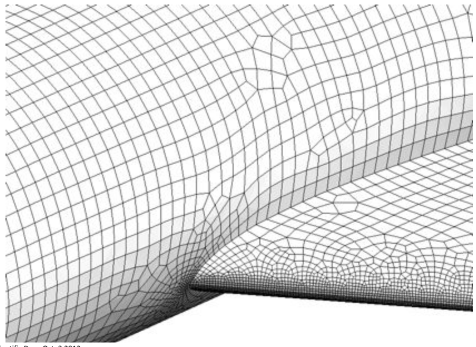
Quad meshing of Falcon



C. Geuzaine - Onera Scientific Day - Oct. 3 2012 44

jeudi 4 octobre 12

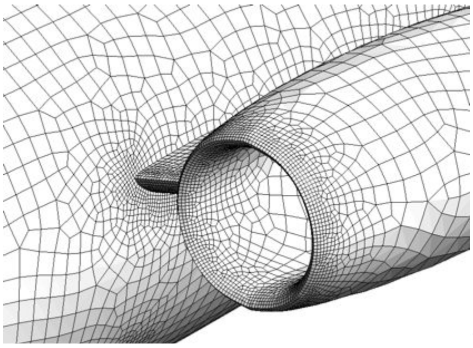
Quad meshing of Falcon



C. Geuzaine - Onera Scientific Day - Oct. 3 2012 45

jeudi 4 octobre 12

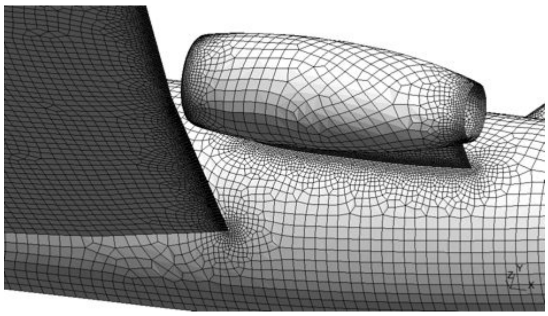
Quad meshing of Falcon



C. Geuzaine - Onera Scientific Day - Oct. 3 2012 46

jeudi 4 octobre 12

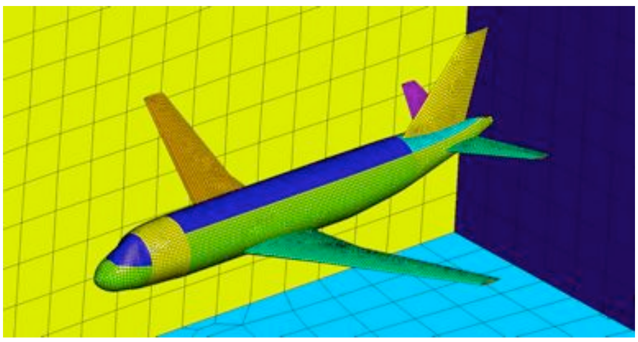
Quad meshing of Falcon



C. Geuzaine - Onera Scientific Day - Oct. 3 2012 47

jeudi 4 octobre 12

Quad meshing of A319



C. Geuzaine - Onera Scientific Day - Oct. 3 2012 48

jeudi 4 octobre 12

Take-home message

Conformal surface parametrization through PDEs allows to:

- create high quality meshes for FE simulations
- remesh CAD patches together for avoiding small elements
- generate quad meshes

Presented methods are :

- automatic
- efficient
- and lead to high quality surface meshes (triangles and quads)

C. Geuzaine - Onera Scientific Day - Oct. 3 2012 49

jeudi 4 octobre 12

Take-home message

All the presented algorithms are available in the open source mesh generator Gmsh (http://geuz.org/gmsh)

Documentation is available in the wiki (https://geuz.org/trac/gmsh) with the username/password: gmsh/gmsh

Current work: hexahedra & high-order meshes

Take-home message

All the presented algorithms are available in the open source mesh generator Gmsh (http://geuz.org/gmsh)

Documentation is available in the wiki (https://geuz.org/trac/gmsh) with the username/password: gmsh/gmsh

Current work: hexahedra & high-order meshes

Thank you for your attention...

cgeuzaine@ulg.ac.be

PS: Doing open source is rewarding

Comment about Gmsh that we found on http://www.fltk.org a few years ago (sic):

```
>From Anonymous, 20:33 May 18, 2004 (score=1)
Je suis outre du programme pour des intelectuels
vous devrez avoir plus d'imagination vous faite
onte au genie informatique
```

2.10 P. Brenner (ASTRIUM ST) - 1 -

Recent developments about overlapping grids for unstructured meshes

P. Brenner[1]
ASTRIUM ST – 66, route de Verneuil – 78133 Les Mureaux – France

ABSTRACT

Overlapping grid techniques are very attractive for the simulation of flows around bodies in relative motion. Nevertheless for supersonic flows where strong shocks are expected, conventional CHIMERA methods quickly become ineffective since they are based on interpolation techniques between meshes. It is the reason why we have developed intersection algorithms: one grid is embedded in the other by calculating the exact geometric intersection at the border of the embedded grid. In our method [4], some cells are fully covered and are therefore excluded from the computation, others are fully uncovered and the remaining class consists of cells that are partially covered and that are connected to the overlapping grid through the intersection surface. The use of a general unstructured finite volume solver (i.e. that can work on any kind of cells) finally allows ensuring a transparent transfer of information between meshes: that simply acts to compute the numerical fluxes on the intersection surface which becomes in fact a simple interface between several cells of a same composite grid.
So, two very important ingredients are needed for the correct working of the code: a technique for calculating efficiently a geometric intersection between several grids and an unstructured solver which is robust and accurate on all kinds of meshes. Very recently, we improved these two components of our software:

- We have generalized the calculation of intersection with an arbitrary number of nested levels (a grid can be overlapped by several other grids which themselves overlap one another) whereas before, each grid could overlap only one other grid... We also optimized the algorithm to reduce CPU consumption to “the minimum”.
- We have implemented into the aerodynamic solver an algorithm to obtain an arbitrary order of accuracy. This technique relies on a compact numerical scheme (i.e. that uses only the direct neighborhood of each cell control) and works on the primitive variables which ensures a great robustness for the simulation of high enthalpy flows when upwind [2] fluxes are used.

These two major developments in the code are based on two simple ideas:

- For the generalization of intersections, it suffices to decompose a multiple intersection in a sum (or a difference) of simple intersections. Therefore, if one [1] already knows how to calculate a simple intersection, the determination of multiple intersections is only to make the algorithm recursive.
- Regarding the aerodynamic solver, it is based on the MUSCL method [3]: accuracy depends on the order of the reconstruction and thus on the estimation of the multiple derivatives of the variables to be reconstructed. To determine these derivatives, the basic idea is that a second derivative is just the derivative of a first derivative... Therefore only the direct neighbors are required but the process becomes iterative. Moreover, an original method based on the concept of the K-Exact reconstructions [5] was developed to make consistent these different derivatives for the conservative variables [6]. Finally, an adaptation of the algorithm allows the use of primitive variables without loss of precision which ensures a high robustness for high-enthalpy flow simulation. At the moment only the third order of accuracy has been implemented but the method is fully universal. Ongoing studies will enable us to determine the required order of accuracy to implement efficiently VLES models.

REFERENCES

[1] G. Monge. « Géométrie Descriptive », Baudouin Imprimeur du Corps législatif et de l’Institut national, Paris, An VII (1799).

[2] S.K Godounov - A. Zabrodine - M. Ivanov - A. Kraiko - G. Prokopov. « Résolution numérique des problèmes multidimensionnels de la dynamique des gaz », Editions MIR, Moscou, 1976.

[3] B. Van Leer. “Towards Ultimate Conservative Difference Scheme V: A second order sequel of Godunov's Method”, Journal of Computational Physics, vol. 32, 1979.

[4] P. Brenner. ”Three Dimensional Aerodynamics with Moving Bodies applied to Solid Propulsion”, AIAA paper 91-2304, 1991.

[5] T. J. Barth. “Recent Developments in High Order K-Exact Reconstruction on Unstructured Meshes”, AIAA paper 93-0668, 1993.

[6] F. Haider - P. Brenner - B. Courbet - J-P. Croisille. “Efficient Implementation of High Order Reconstruction in Finite Volume Methods”, 6th International Symposium on Finite Volume for Complex Applications, Prague, 2011.

[1] pierre.brenner@astrium.eads.net

Recent developments about overlapping grids for unstructured meshes

ONERA Scientific Day, Oct 3, 2012

Astrium Space Transportation
Les Mureaux France

All the space you need

Outline

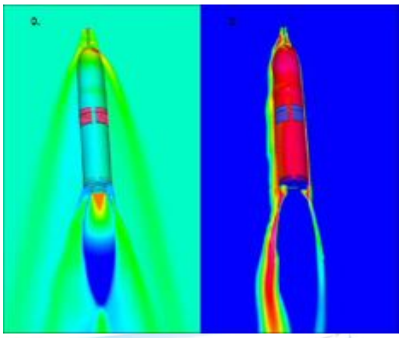
- ➢ **The FLUSEPA code**
  - Core competencies
  - CFD choices
  - Recent developments
- Geometric overlapping
  - Principle
  - Mono/multi-overlapping
- High order compact reconstruction schemes
  - Principles
  - Results
- Conclusions & Perspectives

FLUSEPA *Core competencies*

- Transient/unsteady flows
  - Engine Ignition (Side Loads...)
  - Launcher Takeoff (Blast Waves...)
  - Stages separation
  - Stability at re-entry
  - Security (explosions..)
  - …
- High enthalpy flows
  - Hypersonic (re-entry...)
  - Propulsive flows (engines, nozzles, plumes)

Stages separation

- 2005
  - ✓ **1st stage empty**
  - ✓ **Interstage cut**
  - ✓ **Engine ignition**



FLUSEPA *CFD choices*

- **Unstructured Cell Centred Finite Volume**
  - Hybrid meshes (tetrahedrons, prisms, pyramids, hexahedrons…)
  - General unstructured meshes (any polyhedral cells)
- **Complex physics**
  - Moving bodies (ALE + overlapping meshes)
  - Turbulent models (RANS-VLES)
  - Particles coupling ($H2O$, $AL2O3$...)
  - Chemical modelling
  - ...
- **Accurate & robust numerical schemes**
  - **Conservative CHIMERA method (geometrical intersection)**
  - **High order reconstruction (compact scheme)**
  - Adaptive Temporal Integration (consistent local time step)

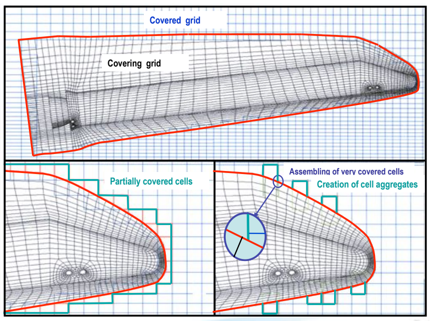
FLUSEPA *Recent Developments*

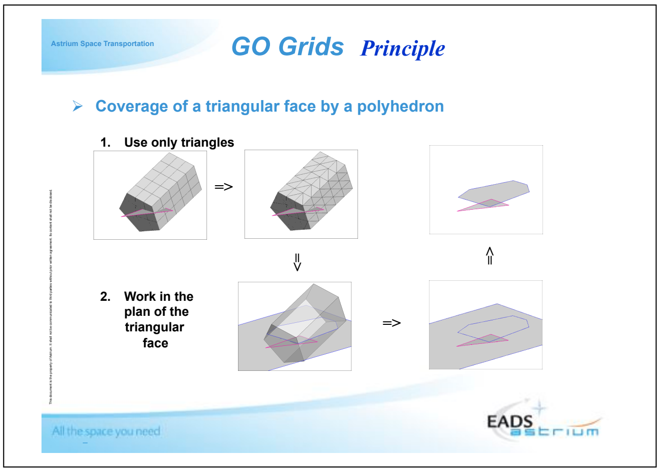
- **Technical developments:**
  - 2010
    - ✓ Implementation of high order compact scheme (3rd order - SC)
    - ✓ Validation of the reconstruction robustness (without quadrature)
  - 2012
    - ✓ Implementation of multi-intersection algorithms — August
    - ✓ Parallel Computing (Domain Decomposition) — September
    - ➢ Validation of high order numerical schemes — End-2012
      - ✓ Subsonic => Isentropic Vortex
      - ➢ Supersonic => Ringleb's Problem

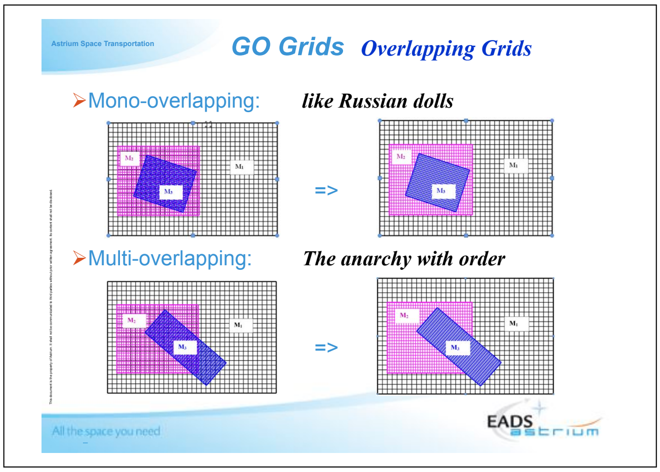
Outline

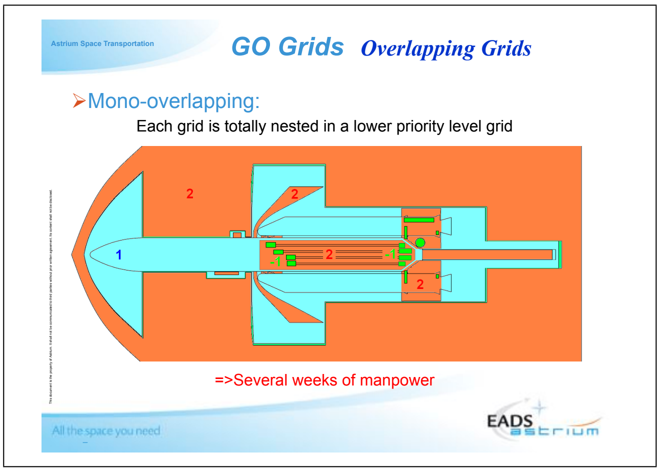
- The FLUSEPA code
  - Core competencies
  - CFD choices
  - Recent developments
- ➢ **Geometric Overlapping**
  - Principle
  - Mono/multi-overlapping
- High order compact reconstruction schemes
  - Principles
  - Results
- Conclusions & Perspectives

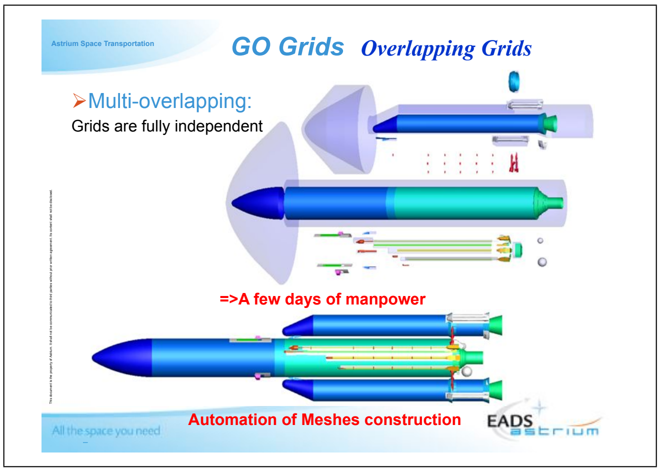
GO Grids *Principle*

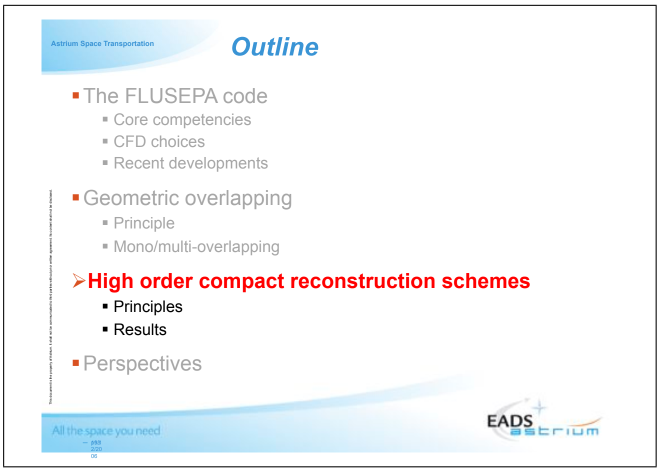


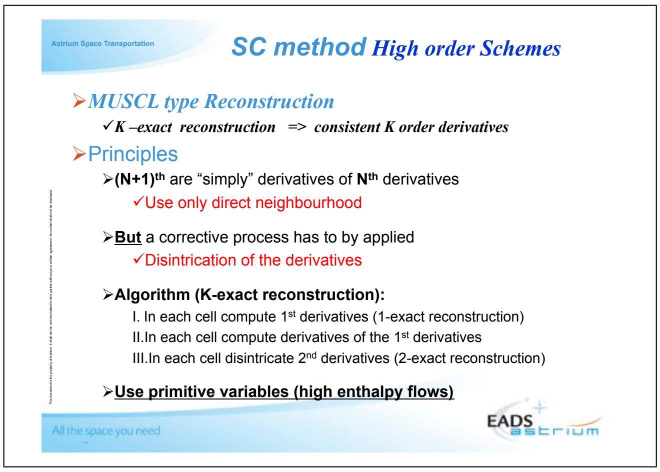

Astrium Space Transportation
GO Grids Principle
Coverage of a triangular face by a polyhedron
1. Use only triangles
=>
2. Work in the plan of the triangular face
=>
All the space you need
EADS astrium

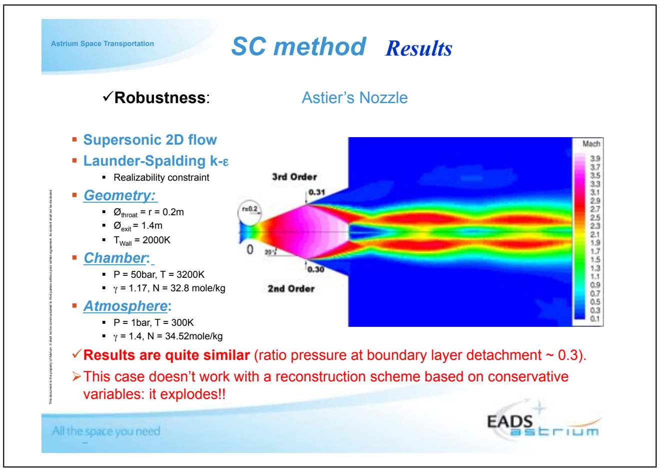

Astrium Space Transportation
GO Grids Overlapping Grids
Mono-overlapping: like Russian dolls
=>
Multi-overlapping: The anarchy with order
=>
All the space you need
EADS astrium

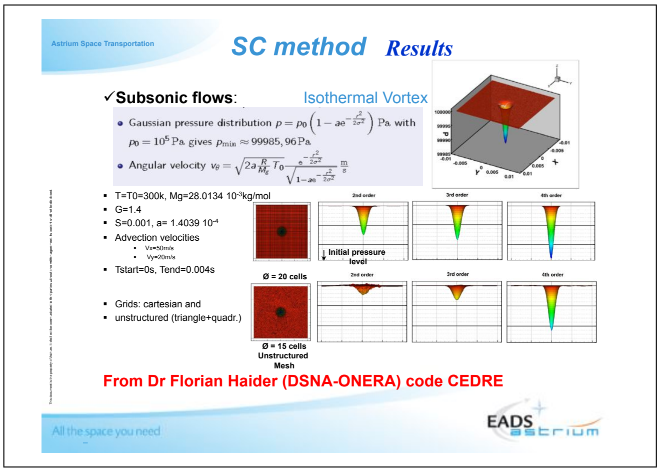

Astrium Space Transportation
GO Grids Overlapping Grids
Mono-overlapping:
Each grid is totally nested in a lower priority level grid
=>Several weeks of manpower
All the space you need
EADS astrium

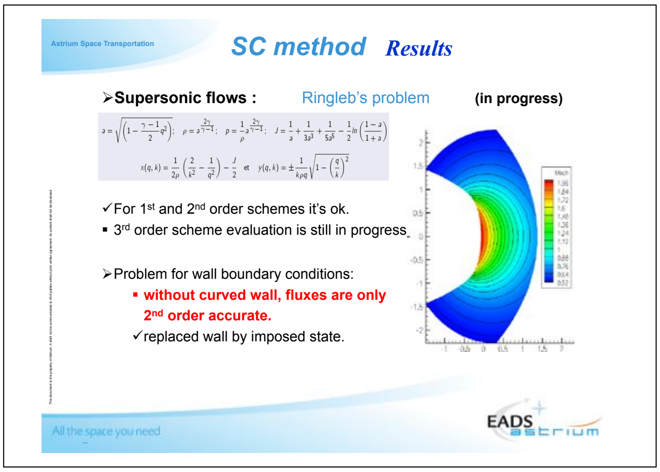

Astrium Space Transportation
GO Grids Overlapping Grids
Multi-overlapping:
Grids are fully independent
=>A few days of manpower
Automation of Meshes construction
All the space you need
EADS astrium

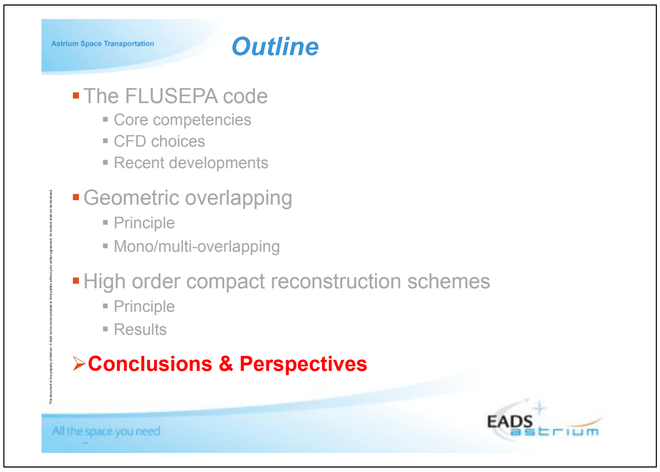

Astrium Space Transportation
Outline
The FLUSEPA code
Core competencies
CFD choices
Recent developments
Geometric overlapping
Principle
Mono/multi-overlapping
High order compact reconstruction schemes
Principles
Results
Perspectives
All the space you need
EADS astrium

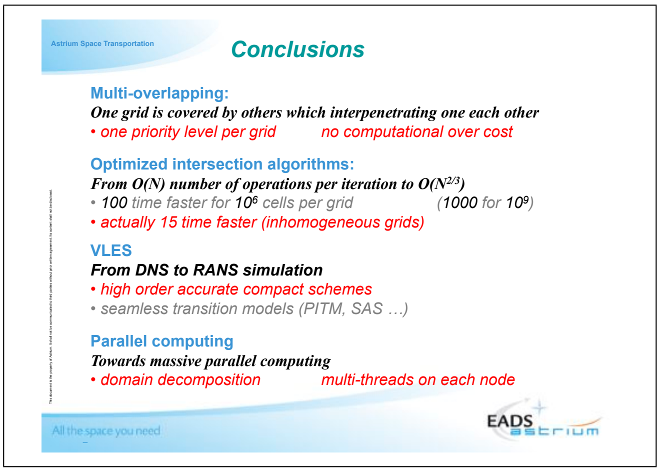

Astrium Space Transportation
SC method High order Schemes
MUSCL type Reconstruction
✓K –exact reconstruction => consistent K order derivatives
Principles
(N+1)th are "simply" derivatives of Nth derivatives
✓Use only direct neighbourhood
But a corrective process has to by applied
✓Disintrication of the derivatives
Algorithm (K-exact reconstruction):
I. In each cell compute 1st derivatives (1-exact reconstruction)
II.In each cell compute derivatives of the 1st derivatives
III.In each cell disintricate 2nd derivatives (2-exact reconstruction)
Use primitive variables (high enthalpy flows)
All the space you need
EADS astrium

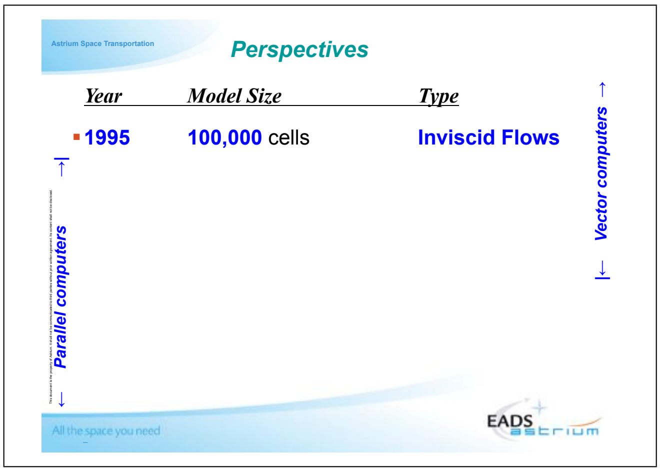

Astrium Space Transportation
SC method Results
✓Robustness: Astier's Nozzle
Supersonic 2D flow
Launder-Spalding k-ε
Realizability constraint
Geometry:
Ø throat = r = 0.2m
Ø exit = 1.4m
T Wall = 2000K
Chamber:
P = 50bar, T = 3200K
γ = 1.17, N = 32.8 mole/kg
Atmosphere:
P = 1bar, T = 300K
γ = 1.4, N = 34.52mole/kg
3rd Order
2nd Order
Mach
✓Results are quite similar (ratio pressure at boundary layer detachment ~ 0.3).
This case doesn't work with a reconstruction scheme based on conservative variables: it explodes!!
All the space you need
EADS astrium

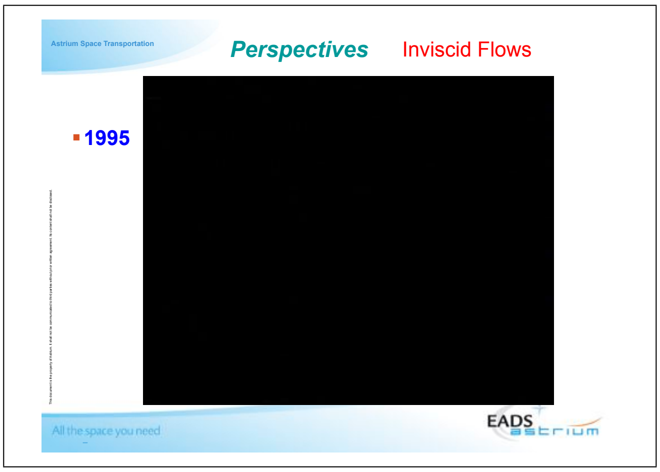

Astrium Space Transportation
SC method Results
✓Subsonic flows: Isothermal Vortex
T=T0=300k, Mg=28.0134 10-3kg/mol
G=1.4
S=0.001, a= 1.4039 10-4
Advection velocities
Tstart=0s, Tend=0.004s
Grids: cartesian and unstructured (triangle+quadr.)
Initial pressure level
Ø = 20 cells
Ø = 15 cells Unstructured Mesh
2nd order
3rd order
4th order
From Dr Florian Haider (DSNA-ONERA) code CEDRE
All the space you need
EADS astrium

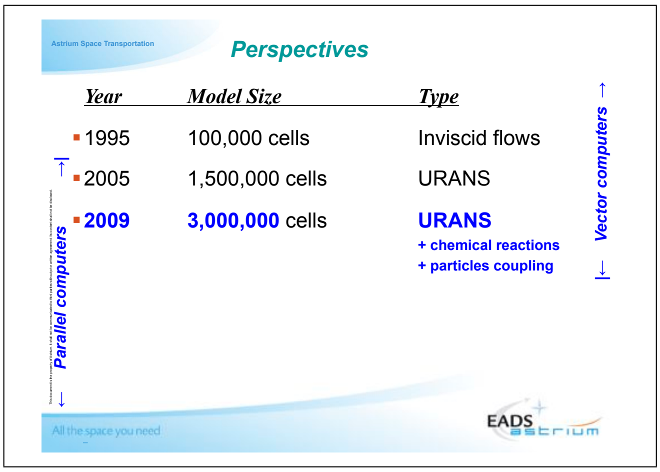

Astrium Space Transportation
SC method Results
➢Supersonic flows : Ringleb's problem (in progress)
✓For 1st and 2nd order schemes it's ok.
▪ 3rd order scheme evaluation is still in progress
➢Problem for wall boundary conditions:
▪ without curved wall, fluxes are only 2nd order accurate.
✓replaced wall by imposed state.
All the space you need
EADS astrium

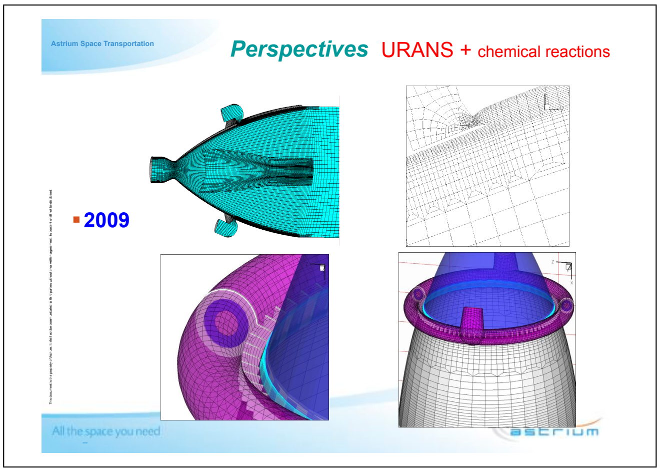

Astrium Space Transportation
Outline
▪ The FLUSEPA code
▪ Core competencies
▪ CFD choices
▪ Recent developments
▪ Geometric overlapping
▪ Principle
▪ Mono/multi-overlapping
▪ High order compact reconstruction schemes
▪ Principle
▪ Results
➢Conclusions & Perspectives
All the space you need
EADS astrium

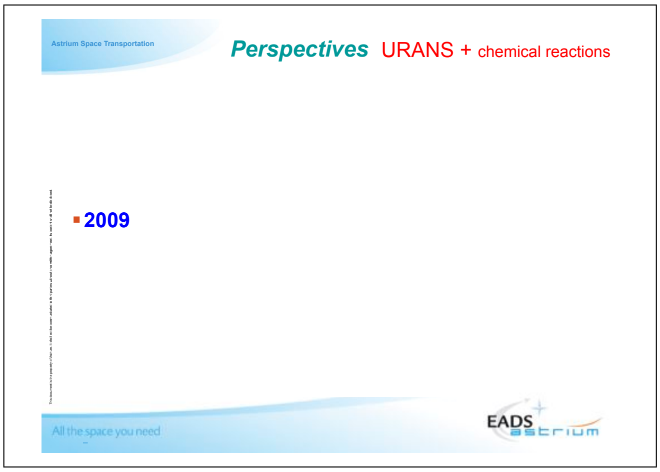

Astrium Space Transportation
Conclusions
Multi-overlapping:
One grid is covered by others which interpenetrating one each other
• one priority level per grid no computational over cost
Optimized intersection algorithms:
From O(N) number of operations per iteration to O(N2/3)
• 100 time faster for 106 cells per grid (1000 for 109)
• actually 15 time faster (inhomogeneous grids)
VLES
From DNS to RANS simulation
• high order accurate compact schemes
• seamless transition models (PITM, SAS …)
Parallel computing
Towards massive parallel computing
• domain decomposition multi-threads on each node
All the space you need
EADS astrium


Astrium Space Transportation
Perspectives
Year Model Size Type
▪ 1995 100,000 cells Inviscid Flows
Parallel computers
Vector computers
All the space you need
EADS astrium


Astrium Space Transportation
Perspectives Inviscid Flows
▪ 1995
All the space you need
EADS astrium


Astrium Space Transportation
Perspectives
Year Model Size Type
▪ 1995 100,000 cells Inviscid flows
▪ 2005 1,500,000 cells URANS
▪ 2009 3,000,000 cells URANS
+ chemical reactions
+ particles coupling
Parallel computers
Vector computers
All the space you need
EADS astrium


Astrium Space Transportation
Perspectives URANS + chemical reactions
▪ 2009
All the space you need
astrium


Astrium Space Transportation
Perspectives URANS + chemical reactions
▪ 2009
All the space you need
EADS astrium

Perspectives

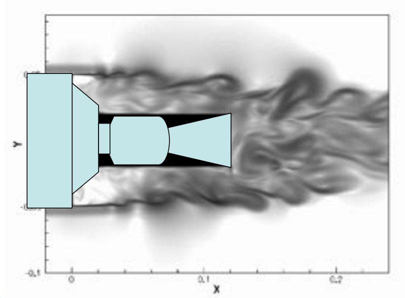
Year	Model Size	Type
1995	100,000 cells	Inviscid flows
2005	1,500,000 cells	URANS
2009	**3,000,000** cells	URANS + chemical reactions + particles coupling **VLES small scale**

↑ Vector computers ↓

↑ Parallel computers ↓

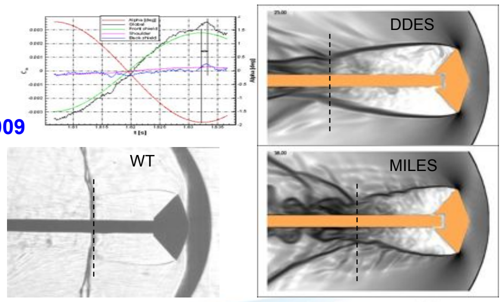
Perspectives Buffeting small scale

- **2009**



Perspectives Stability at Re-entry

- **2009**



Perspectives

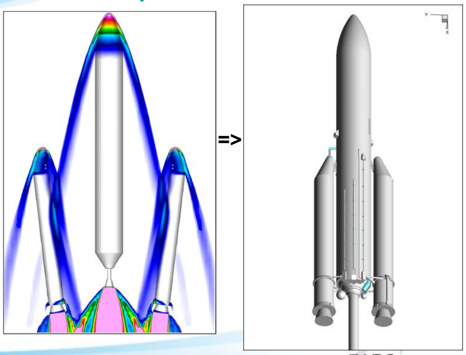
Year	Model Size	Type
1995	100,000 cells	Inviscid flows
2005	1,500,000 cells	URANS
2009	3,000,000 cells	URANS + chemical reactions + particles coupling VLES small scale
2013	**≥ 50,000,000** cells	**VLES/URANS ++**

↑ Vector computers ↓

↑ Parallel computers ↓

Perspectives URANS + chemical reactions

- **2013**



Perspectives

Year	Model Size	Type
1995	100,000 cells	Inviscid flows
2005	1,500,000 cells	URANS
2009	3,000,000 cells	URANS + chemical reactions + particles coupling VLES small scale
2013	≥ 50,000,000 cells	VLES/URANS ++
201?	**≥ 1,000,000,000** cells	**VLES full scale**

↑ Vector computers ↓

↑ Parallel computers ↓

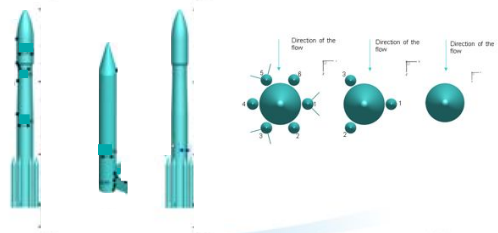
Perspectives

- 2012 **VLES** methods
  - numerical scheme requirement
  - models (SAS, PANS, PITM, …)
- 2013 **HPC** for CFD with overlapping grids
  - Parallel computing MPI-openMP
  - parallel computing for large grids ($\geq 10^9$ cells)
- 2014(?) Mesh adaptation methods
  - **A**utomatic **M**esh **R**efinement
  - deforming grids
  - mesh displacement (following of shocks & vortexes)

Other activities

Automated Preliminary Design Tool : **CASCADES**

- **use of overlapping grid technique to simplify meshing**
- **Mesh automation with bow shock fitting**
- **Automation of the post-treatments**



2.11 S. Péron, C. Benoit, P. Raud (ONERA)

Abstract proposed to the ONERA Scientific Day, 3^{rd} October 2012

Cassiopée: pre- and post-processing for CFD python/CGNS workflow

S. Péron, C. Benoit, P. Raud

ONERA -The French Aerospace Lab, CFD and Aeroacoustics Department
BP 72, 92322 Châtillon Cedex, France

Due to the increasing complexity of numerical simulations, the number of numerical components intervening in a simulation is growing. Today, it is common to run CFD, CSM, Optimizer,... together. To achieve this goal, the industry paradigm was up to now directed by the concept of chaining numerical simulations, where the output of each code was reintroduced in the following code, generally after home-brewed routines of data transformation. Nevertheless, with the application diversifications and the number of problems now submitted to numerical analysis, the concept of a fixed chain becomes very restrictive. To be able to adapt the numerical tools for different applications, the concept of workflow has been introduced. In this frame, each numerical component relies on a single data model, which is modified by a numerical component that outputs a fully compatible data model. ONERA, through the work of M. Poinot [8], chooses CGNS [1] as data model and Python to carry the workflow. The standard CGNS data model is represented in Python [3] and each numerical function is interfaced to Python. Cassiopée (for CFD Advanced Set of Services In an Open Python EnvironmEnt) [2] has then been developped to provide pre- and post-processing functions in this Python/CGNS environment. Its range of application is:

+ Meshing and remeshing:
  - surface mesh generation by orthogonal walk [4],
  - extrusion and transfinite interpolation methods for volume mesh generation,
  - unstructured octree mesh generation and derivation to off-body Cartesian mesh generation [7],
  - collar grid mesh generation for intersecting grids [5],
  - remeshing: splitting, merging, coarsening, refining, densifying a mesh,
  - unstructured octree and structured Cartesian mesh adaptation[7].
+ Multiblock and overset grid assembly:
  - Automatic computation of abutting (1-to-1 and 1-to-n) connectivity between structured blocks,
  - Chimera hole-cutting: blanking of cells lying inside bodies and overlap optimization [6],
  - Computation of Chimera connectivity (interpolation coefficients and donors).
+ CFD solution post-processing:
  - Signed distance field computation,
  - Computation of aerodynamics variables (pressure, Mach number,...), gradient of a given field, curl of a vector, taking into account the abutting connectivity,
  - Slices, isoline and isosurface extraction, taking into account the Chimera nature of points (blanked, interpolated, computed),
  - High-order interpolation from a solution defined on a mesh to another mesh, taking into account the Chimera nature of points.

In our talk, we will illustrate our approach on two examples. The first example concerns the grid assembly for arbitrary intersecting bodies, with automatic grid generation of overset grids at body junction. In the second example, we will present a workflow consisting of generating and adapting an octree mesh periodically according to the solution of a CFD solver.

References

[1] http://cgns.sourceforge.net.

[2] http://elsa.onera.fr/Cassiopee/Userguide.html.

[3] The CGNS Steering Committee. SIDS-to-Python (CGNS/Python). Release 3.1.2. http://www.grc.nasa.gov/WWW/cgns/CGNS_docs_current/python/sidstopython.pdf.

[4] W. M. Chan and P. G. Buning. Surface grid generation for Overset grids. *Computers and Fluids*, 24(5):509–522, 1995.

[5] S. J. Parks, P. G. Buning, W. M. Chan, and J. L. Steger. Collar grids for intersecting geometric components within the Chimera overlapped grid scheme. pages 672–682, 1991.

[6] S. Péron and C. Benoit. A python module for Chimera assembly. 10^{th} Overset Grid Symposium, 2010. http://2010.oversetgridsymposium.org/assets/pdf/presentations/1_1/Peron_OversetSymposium2010.pdf.

[7] S. Péron and C. Benoit. Off-body Overset adaptive Cartesian mesh method based on an octree approach. AIAA paper 2011-3050, 2011.

[8] M. Poinot. Five Good Reasons to Use the Hierarchical Data Format. *Computing in Science Engineering*, 12(5):84 –90, 2010.

- Cassiopée -
Pre- and Post-processing for
CFD python CGNS workflow

S. Péron, C. Benoit, P. Raud, S. Landier
Onera / DSNA

CFD Workflow : Meshing, Solving, Visualizing...
OSD : Onera Scientific Day – October 3rd, 2012

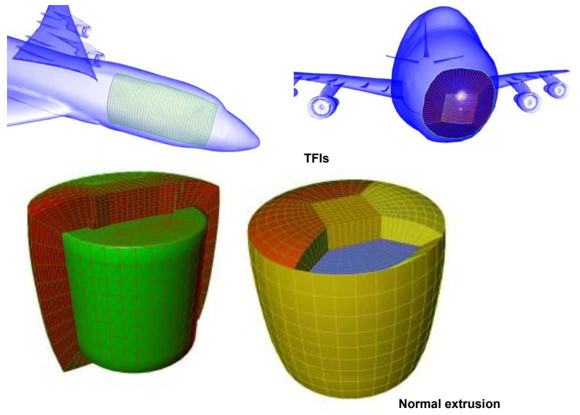
Python/CGNS

- Based on Python/CGNS
  - CGNS: standard/well established data model
  - Python: high level script language, easy to use
  - Python/CGNS standard (M. Poinot)

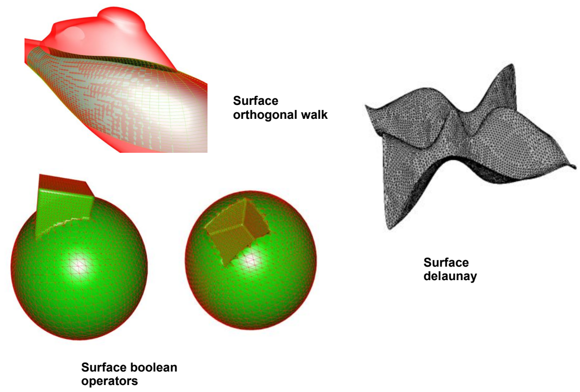
Python/CGNS

- Full CFD computation case is stored in a tree
  - Meshes, BCs, settings...
- Tree is stored as an imbricated set of python lists
- Cassiopée: a set of functions (python modules)
  - t' = f(t), t is the python CGNS tree
  - Generator : Mesh generation module
  - Transform : block transformation module
  - Connector : connectivity module
  - Post : solution post-processing module

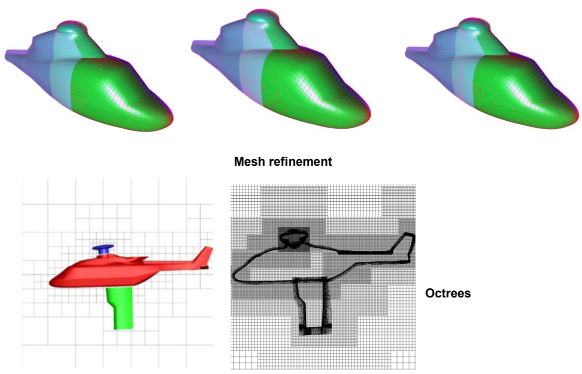
Generator



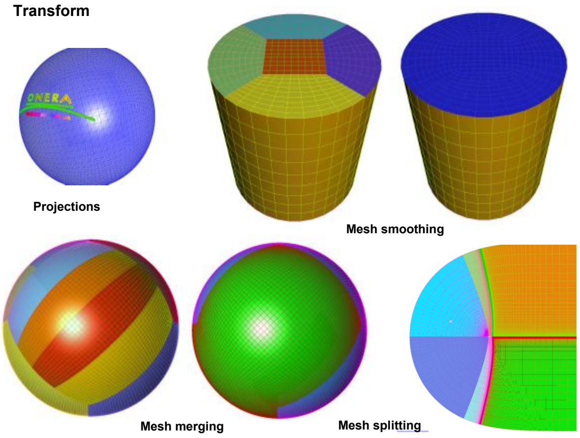
Generator



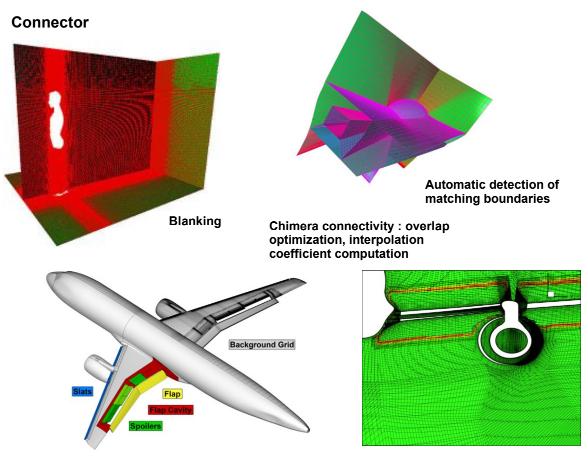
Generator



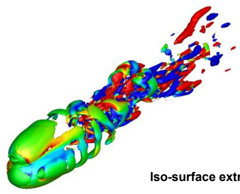
Transform



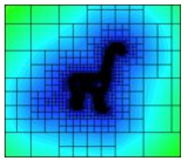
Connector



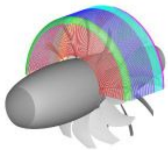
Post

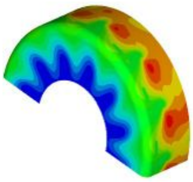


Iso-surface extraction



Signed distance field

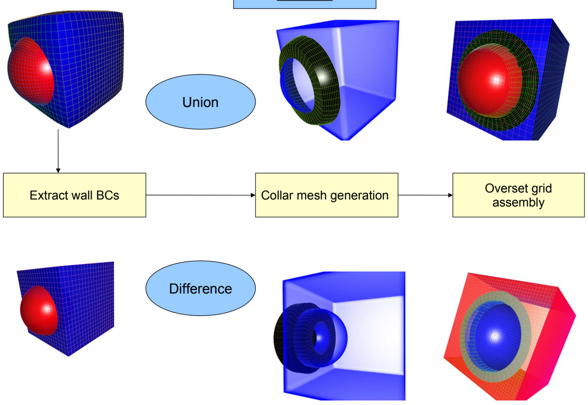




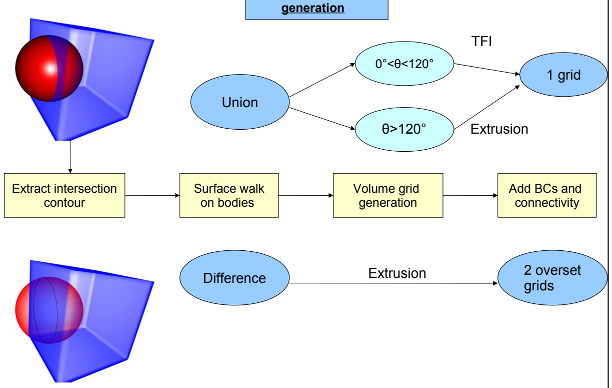
Surface extraction

Application to automatic grid assembly
(collar grids + chimera)

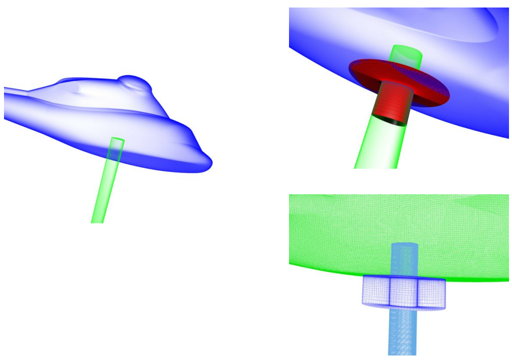
Workflow



Collar mesh generation



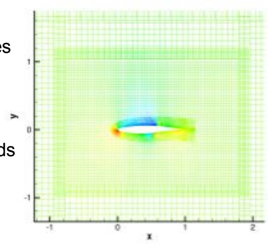
Example : DGV fuselage with a strut

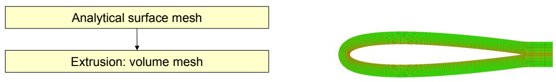


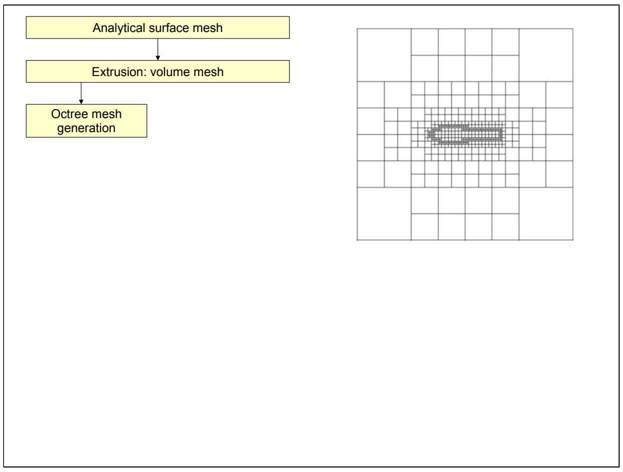
Application to Cartesian mesh generation
and adaptation

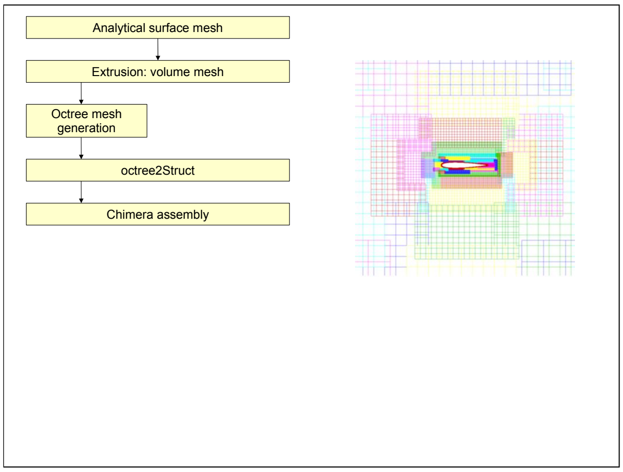
Framework

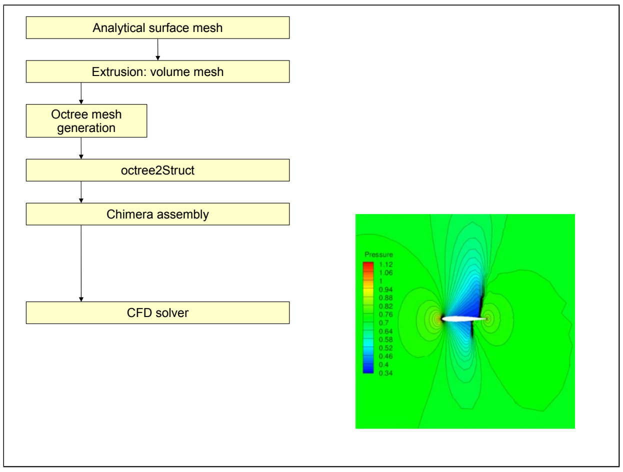
- The computational domain is partitioned into:
  - near-body regions around bodies (fuselage, wing, ...)
  - off-body regions
- Each geometrical component is meshed independently by a set of grids extending a short distance in the domain
- Off-body regions are described by a set of adaptive Cartesian grids, overlapping near-body grids

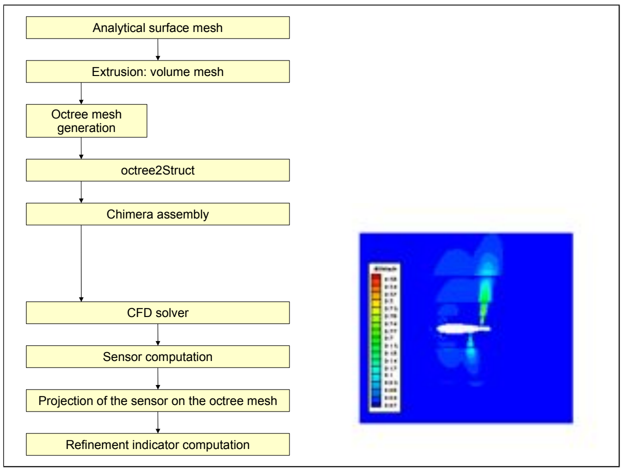


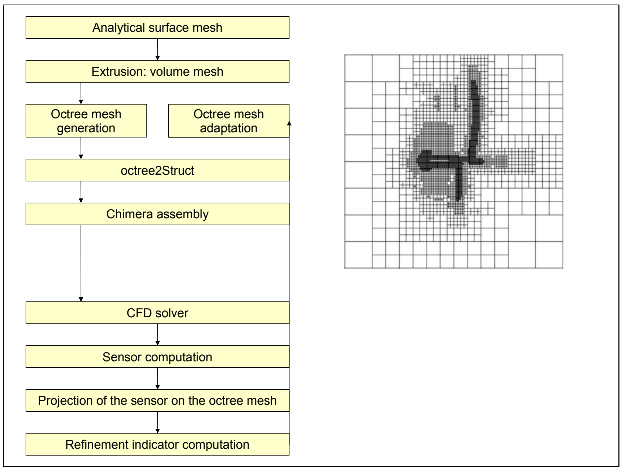


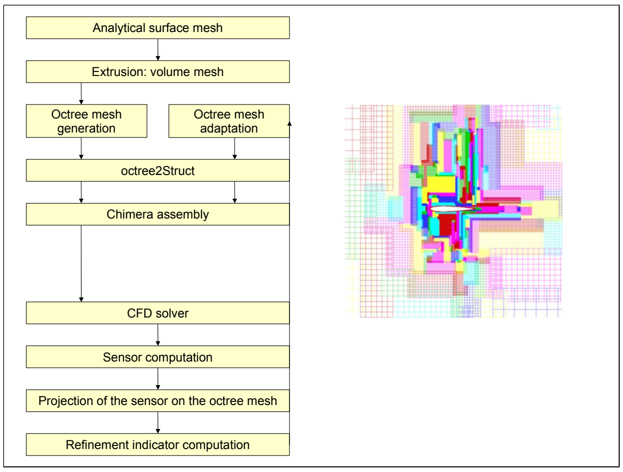

Analytical surface mesh
Extrusion: volume mesh
Octree mesh generation

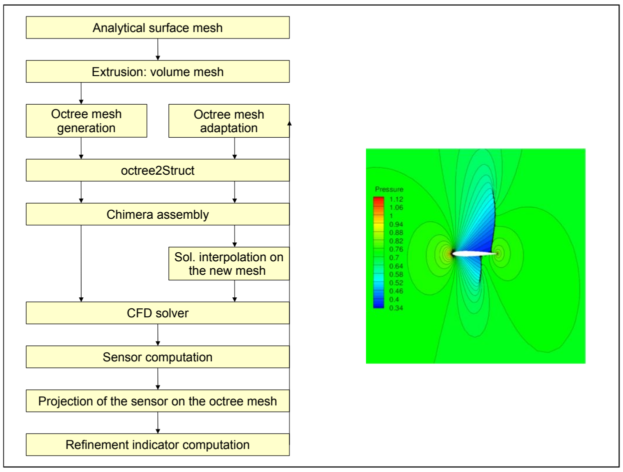

Analytical surface mesh
Extrusion: volume mesh
Octree mesh generation
octree2Struct
Chimera assembly

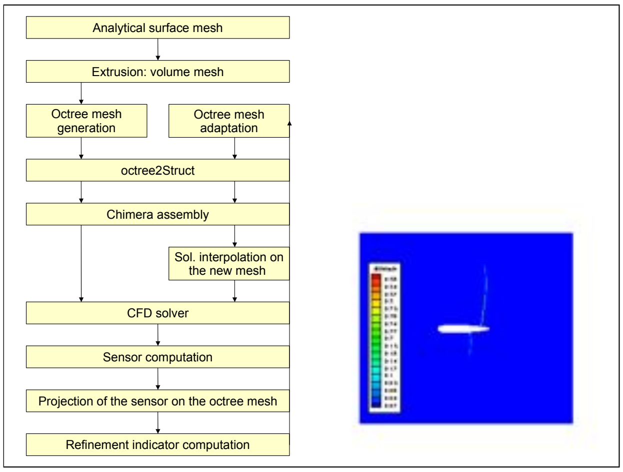

Analytical surface mesh
Extrusion: volume mesh
Octree mesh generation
octree2Struct
Chimera assembly
CFD solver


Analytical surface mesh
Extrusion: volume mesh
Octree mesh generation
octree2Struct
Chimera assembly
CFD solver
Sensor computation
Projection of the sensor on the octree mesh
Refinement indicator computation


Analytical surface mesh
Extrusion: volume mesh
Octree mesh generation
Octree mesh adaptation
octree2Struct
Chimera assembly
CFD solver
Sensor computation
Projection of the sensor on the octree mesh
Refinement indicator computation

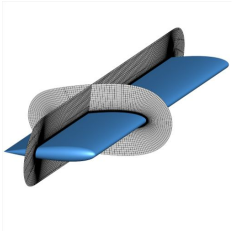

Analytical surface mesh
Extrusion: volume mesh
Octree mesh generation
Octree mesh adaptation
octree2Struct
Chimera assembly
CFD solver
Sensor computation
Projection of the sensor on the octree mesh
Refinement indicator computation


Analytical surface mesh
Extrusion: volume mesh
Octree mesh generation
Octree mesh adaptation
octree2Struct
Chimera assembly
Sol. interpolation on the new mesh
CFD solver
Sensor computation
Projection of the sensor on the octree mesh
Refinement indicator computation


Analytical surface mesh
Extrusion: volume mesh
Octree mesh generation
Octree mesh adaptation
octree2Struct
Chimera assembly
Sol. interpolation on the new mesh
CFD solver
Sensor computation
Projection of the sensor on the octree mesh
Refinement indicator computation

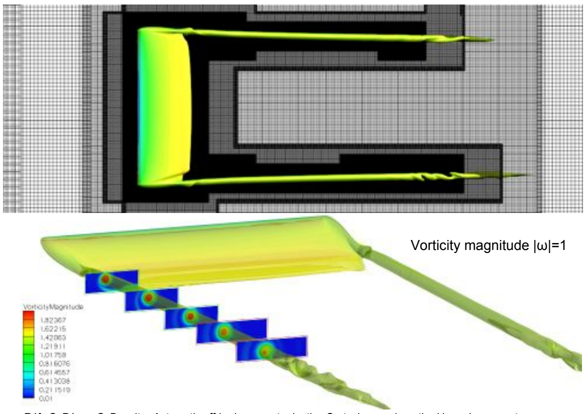
Example: RANS simulation of the flow on a NACA0015 rounded tip wing

- Simulation:
  - Exp. by McAlister & Takahashi
  - M=0.1235 , AoA=12°, Re = 2 million
  - Rectangular wing, rounded tip and root, blunt trailing edge
- Near-body mesh:
  - Near-body mesh obtained by extrusion from the analytical surface mesh: 297x101x90 points
  - Overlap BCs applied at external borders
  - Spacing at external borders ~ 2%c



Workflow

- Initial octree mesh refined in the vicinity of external surfaces of the near-body mesh
- Derivation to Cartesian grids, with dx_{min}=2%c (8MPts over 72 blocks)
- Chimera assembly with overlap optimization
- RANS simulation using AUSM+ scheme, Wilcox k-ω turbulence model
- Cartesian off-body mesh adaptation performed according to the previous workflow:
  - Sensor field: streamwise component of the vorticity
  - Adaptation performed every 500 iteration (9 times)



Réf : S. Péron, C. Benoit, «*Automatic off-body overset adaptive Cartesian mesh method based on an octree approach*», Journal of Computational Physics, 2012, http://dx.doi.org/10.1016/j.jcp.2012.07.029 (on line)

Conclusion

- Cassiopée contains a set of pre- and post-processing functions
- All the functions operate on the same data (CGNS/Python tree)
- This enables to quickly design solutions for mesh generation/adaptation/assembly and post-processing.

2.12 M. Poinot (ONERA)

Numerical **S**imulation **C**omponents in **O**pen **P**ython **E**nvironment

Marc Poinot
ONERA
THE FRENCH AEROSPACE LAB
Computational Fluid Dynamics and Aeroacoustics Dept.

marc.poinot@onera.fr

NSCOPE is an Onera self-funded project which aims at defining and spreading software technology for numerical simulation interoperability. NSCOPE takes existing technologies and tries to define simple rules to make software component interface compliant to Open Systems requirements.

Multi-physics simulations

The CFD dept. of Onera has/had to integrate the elsA CFD solver[1] and its pre/post tools in many proprietary or third party frameworks (FSDM/Airbus, Canelle/SAFRAN, GANESH/Eurocopter, Salome/EDF-CEA, MpCCI/Fraunhofer, OpenPALM/Cerfacs...). These frameworks are used for home-defined data-flow processes or sometimes for actual process-based workflow. Most of them are MPI-based and have a low-level system view of the application rather than an high level (data model, algorithm). All these integration experiences helped us to learn how to insulate/isolate our software components from dedicated technologies or even algorithms (for example time or space interpolation methods, when possible...).

Open System

The use of a proprietary integration framework is a strong requirement in large companie, the overall system is then well-defined for both applications and underlaying computer facilities. We cannot subsitute the proprietary environment with ours or any other industrial, third-party or open source environment. At the same time we want to reduce development and maintenance cost, as well as learning curve for new scientists and engineer for whom the software is not the main concern.

Thanks the world wide web, most engineers can now quickly and efficiently learn new software technologies, including applications, middleware, libraries or even algorithms (for example in the context of parallel computations). The selection of widely used techs helps to find accurate documentations and examples, and as a side effect also helps to find new engineers with required skills as well as new software with required interfaces.

Rather than producing 'yet another framework', the amount of integration experience we have with elsA and its tools lead us to a non-intrusive approach of the software component interface.

NSCOPE

The project has three themes: the **software engineering**, the **component repository** and the **application assembly**. The project tasks are expert meetings and guide/documentation editing. The actual result is a set of recommandations with many code examples. The top requirement is all deliverable can be implemented and used on any platform without any dependancy on a proprietary item.

- Software engineering deals with component environment and life cycle, such as software distribution to support for components, how to document, how to test, how to manage binary release for proprietary software, how to integrate in a framework using stubs instead of actual component, how to detext and define dependencies with other components, how to release a patch...

- Component repository is a set of component skelettons or templates, with examples of use of specific techs such as how to make a component from an existing Fortran code + Cython/Numpy[7][8], or how to create an asynchroneous server that exchanges CGNS/Python[3][6] trees between a cluster and a remote server, how to get data in memory to monitor application in run-time, how to load/save partial and shared data using HDF5[4][9], how to integrate the component in specific proprietary frameworks...
- Application assembly deals with high level application concerns such as how to define a CGNS data model for time dependant simulations with time or space interpolation, how to distribute parts of data amongst components, how to manage errors/failures with many components, how to select a correct component depending on application criteria such as large/small memory, large/small/no disk usage, how to manage simulation with more than 10Gb of data generated per iteration.... This theme also has a research topic on interface definition assembly proof[2][5], the goal is to simulate an off-line integration to check interfaces and algorithm before (or at the same time) the actual development.

Status

NSCOPE is a five years project involving eight departements in Onera, the first year is 2012 and the main deliverable is a roadmap defining and priorizing processes and technologies to document. We eventually plan to open some NSCOPE meeting to recognized experts in CFD workflows, including DLR, Cenaero, Cerfacs, AIRBUS, SAFRAN, Eurocopter... in 2013/2014.

References:

[1].“Multi-Physics Coupling Approaches for Aerospace Numerical Simulations” , M. Errera, A. Dugeai, Ph. Girodroux-Lavigne, J.-D. Garaud, M. Poinot, S. Cerqueira, G. Chaineray, Aerospace Lab issue 2 march 2011.

[2]."Impact of CGNS on CFD workflow" (AIAA 2004-2142), Poinot M., Rumsey C., Bush R., 34th AIAA Fluid Dynamics Conference, June 2004, Portland (OR), U.S.A.

[3]."Recent Updates to the CFD General Notation System (CGNS)," C. Rumsey, B. Wedan, T. Hauser, M. Poinot (AIAA Paper 2012-1264).

[4]."Five Good Reasons to Use the Hierarchical Data Format", Marc Poinot, Computing in Science and Engineering, vol. 12, no. 5, pp. 84-90, Sep./Oct. 2010, doi:10.1109/MCSE.2010.107

[5]."Checking of CFD interfaces in a multi-disciplinary workflow with an XML/CGNS compiler" (AIAA 2005-1155), Poinot M., Montreuil E., Hénaux E. 43rd AIAA

[6].http://www.cgns.org

[7].http://www.python.org

[8].http://numpy.scipy.org

[9].http://www.hdfgroup.org

NSCOPE

Numerical Simulation Components in Open Python Environment

Marc Poinot
DSNA/CS2A

ONERA
THE FRENCH AEROSPACE LAB

retour sur innovation

Context

- Onera
  - Component assembly know-how & Component provider
  - Lot of various numerical simulation codes
- Industrial requirements
  - Each dedicated platform is the best for its own purpose
  - Components have to be integrated in each best platform

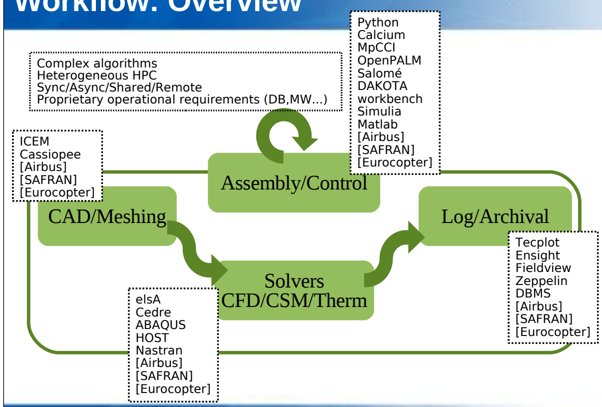
Define a common interface for Onera numerical simulation components to make them the most integrable/portable into industrial platforms and to insure lowest development/maintenance cost for Onera

  - Use Open Systems approach
  - Components declare/use only public standard interfaces
- Strategy
  - Gather Onera depts experts
  - Factorize existing interfaces

elsA and Workflows

- Experience in workflows and Python
  - Onera elsA CFD solver
    - Actual multi-physics simulations
    - Onera own computations
    - Support large companies
    - Many workbenches/frameworks
  - CGNS/Python interface
    - Onera Open Source
    - French community actors since 1999
    - Techno spreading in Systematic/OpenHPC 2009
  - CGNS Steering Committee member since 2001
- How to focus on Onera know-how?
- How to increase codes availability for frameworks?

Workflow: Overview

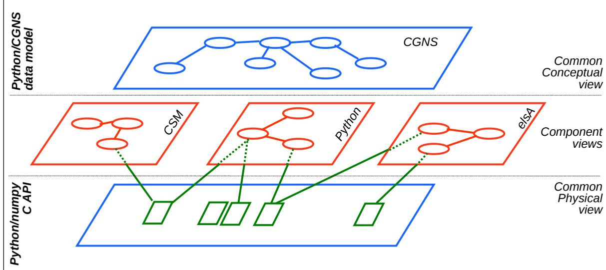


Workflow: Usual framework services

- Control
  - Code identification
  - Distributed systems services
    - Initialization/Finalization/Synchronization
    - Algorithm
  - Communication inter-codes
    - Memory/Network/File
- Data
  - Data identification
  - Serialization/Marshalling/Translation
    - Interpolation wrt space/time

Proprietary language
MPI-based
Proprietary intrusive control
No algo representation
No automata model
Proprietary language
raw data
MPI-based
No time representation
Not-open algorithm/interface
No data model

Workflow: Data Model vs Implementation



ARF NSCOPE

Numerical Simulation Components in Open Python Environment

nscope

- Define a Python HPC component
  - Build components from existing codes
  - Build/define new components
- Spend time on high-level know-how instead of codes
  - Select technos, create patterns, tutorials
- Three circles
  - DSNA, DAAP, DADS, DTIM, DEFA, DMSM, DEMR, DOTA, DMPH
  - Industry, organisation & software vendors selected experts
  - WWW
- First year 2012 only opens to First circle
  - Trying to define a common practice alone is... non sense

Some NSCOPE Topics

- Software engineering
  - Component life-cycle
    - Distribution/maintenance
    - Binary/protected releases
  - Component definition
    - Interface, data model & algo formalisms
    - N-1 vs N+1 interfaces
- Component
  - Building strategies
    - Fortran/C/C++ HPC codes encapsulation
    - Pure Python to HPC Python
  - Actual technology implementation
    - Python/Cython/numpy, CGNS/HDF5, MPI/Multi-threading, DBMS/GUI
- Assembly
  - End-user services
    - Interfaces DB/proof
  - Patterns

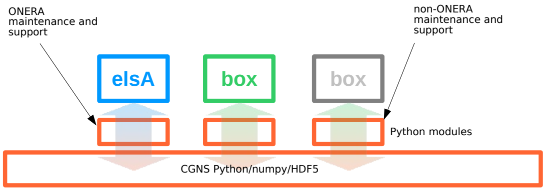
What is CGNS/Python ?

- A Python mapping of CGNS data model
  - Can be used for run-time interoperability as well as archival
  - Bet on the CGNS/SIDS data model
- Use the file at the right time
  - All workflow is in memory
    - Read the generated mesh, continue with per-processor sub-tree
    - Compute, post and view in memory
    - Archive at the end
  - Use HDF5
    - True de facto standard for files, large community
- Onera Open Source contribution
  - Large set of CGNS/Python modules and tools into pyCGNS
    - `http://pycgns.sourceforge.net`
  - CGNS/HDF5 light and thread safe implementation
    - `http://chlone.sourceforge.net`

NSCOPE-PRS-003/9/12 M.Poinot DSNA/CS2A

Non-intrusive approach

- Component life cycle vs workbench life cycle
- Maintenance cost is lower
  - Large common part
  - Specific part uses only non-proprietary standards
  - Requires a profile



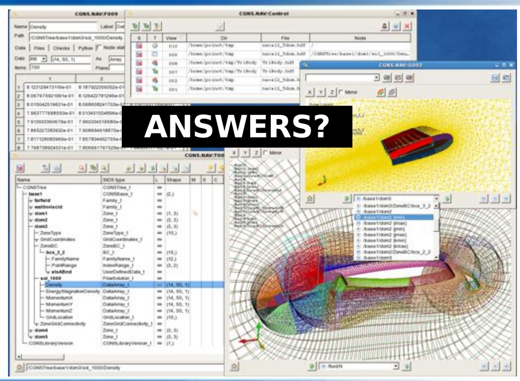
NSCOPE-PRS-003/10/12 M.Poinot DSNA/CS2A

NSCOPE deliverables

- 4Q12 - Topics identification
- 1Q13 - Second circle creation
- 2Q13 - Onera Scientific Day
- 4Q13 - Technology selection, Distribution
- 1Q14 - Component model definition with formalism
- ASAP - User guides, patterns, tutorials

NSCOPE-PRS-003/11/12 M.Poinot DSNA/CS2A

A common data model & implementation



NSCOPE-PRS-003/12/12 M.Poinot DSNA/CS2A

2.13 S. Deck, P.E. Weiss, R. Pain (ONERA)

Onera Scientific Day 2012.

Some Reflections on massive post-processing of large unsteady flow simulation datasets

S. Deck, P.E. Weiss, R. Pain
Onera-The French Aerospace Lab, F-92190, Meudon, France

The computational power has dramatically increased over the last decades. As a first consequence, the development of advanced modelling approaches (LES, RANS/LES) has received increasing attention among turbulence modelling specialists, CFD code developers and industrial CFD engineers. As an example, hybrid methods that couple the solution of the Reynolds Averaged Navier-Stokes (RANS) equations in equilibrium regions with Large Eddy Simulation (LES) in non-equilibrium regions of the flow are acquiring increasing prominence among the CFD community.

A second consequence of this upsurge in computational power is the rapid growth in the size of subsequent data sets, with unsteady 50-100 million point grids simulations now being conducted with increasing regularity. As the need for higher accuracy simulations has risen, the computational fluid dynamics (CFD) community has in turn put emphasis on assessing the quality of the results and now focuses a great deal of its effort on validation of advanced methods (see figure 1).

This presentation describes why improving the way the data from unsteady computations of turbulent flows are post-processed is needed. Some important issues like the spectral analysis of short duration data, the comparison of two data sets of different duration (e.g. simulations and experiment) will be discussed and illustrated on the basis of a high Reynolds axisymmetric separating/reattaching flow [2].

These issues lead to further remarks on the need for CFD research scientists to gather an improved knowledge of their available hardware. Indeed, when large data sets are involved, one has not only to assess the physical meaning of the chosen analysis but also its feasibility in term of IT equipment. The CPU cost of a post-processing technique on a large amount of data can have an order of magnitude equivalent to the computational resources required to simulate a configuration and generate the related unsteady data. Thus, the relationship between the performance of the hardware (e.g. storage media) and the manipulation of large scale matrices is also discussed.

	Level of validation
1	**Integral forces** (lift, drag, pitch)
2	**Mean aerodynamic field** (velocity or pressure profiles)
3	**Second order statistics** (rms quantities)
4	**One-point spectral analysis** (PSD)
5	**Two-point spectral analysis** (correlation, coherence and phase spectra)
6	**High-order and time-frequency analysis** (Time-Frequency, bi-coherence spectra)

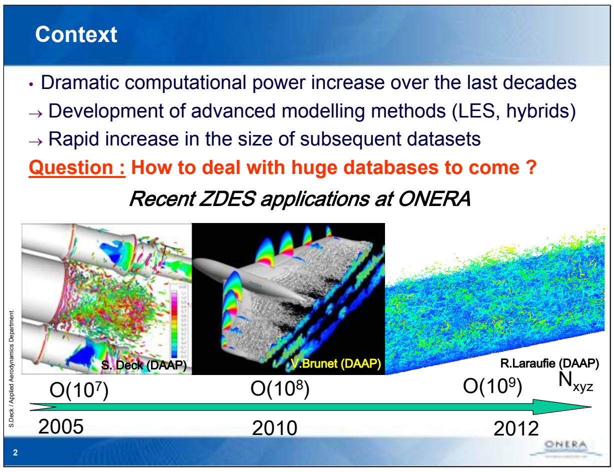
Figure 1: Levels of validation of unsteady simulation techniques (from [1])

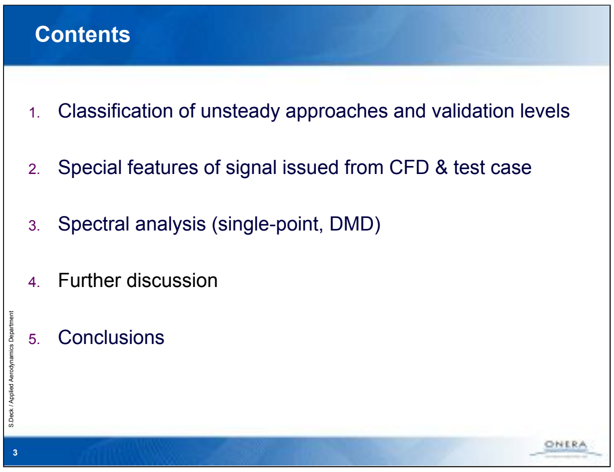
References

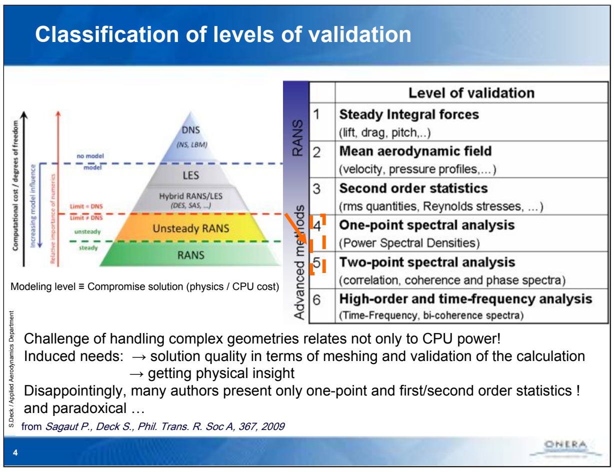
[1] Sagaut, P., Deck, S., "Large Eddy Simulation for Aerodynamics: status and perspectives", *Phil. Trans. R. A.,* Vol. 367, pp. 2849-2860, 2009.

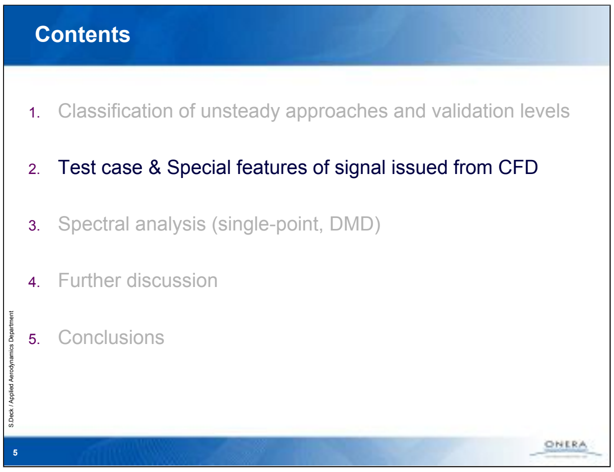
[2] Weiss, P.E., Deck, S., Robinet, J.C., Sagaut, P.. "On the dynamics of axisymmetric turbulent separating/reattaching flows", *Physics of Fluids, 21, 075103, 2009*.

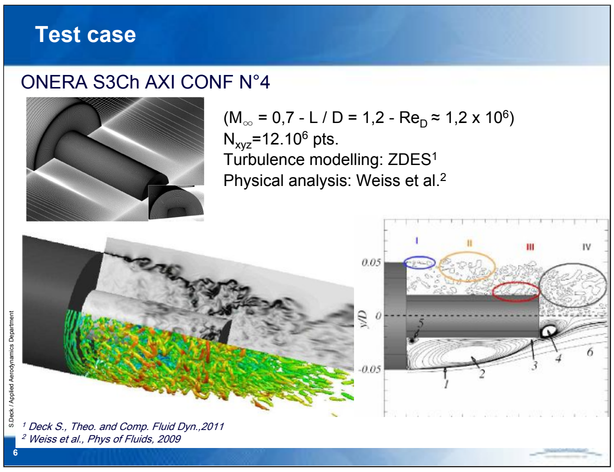

Some reflections on massive post-processing of large unsteady flow simulation data sets
Sébastien DECK, Pierre-Elie WEISS, Romain PAIN
Applied Aerodynamics Department
ONERA
THE FRENCH AEROSPACE LAB
return on innovation

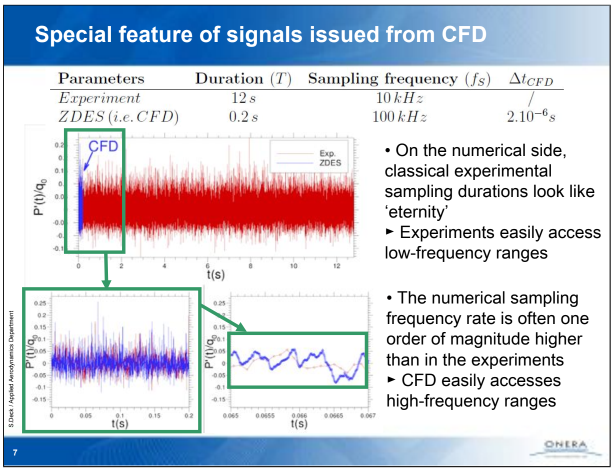

Context
• Dramatic computational power increase over the last decades
→ Development of advanced modelling methods (LES, hybrids)
→ Rapid increase in the size of subsequent datasets
Question : How to deal with huge databases to come ?
Recent ZDES applications at ONERA
S. Deck (DAAP)
V.Brunet (DAAP)
R.Laraufie (DAAP)
$O(10^7)$
$O(10^8)$
$O(10^9)$
N_{xyz}
2005
2010
2012
2

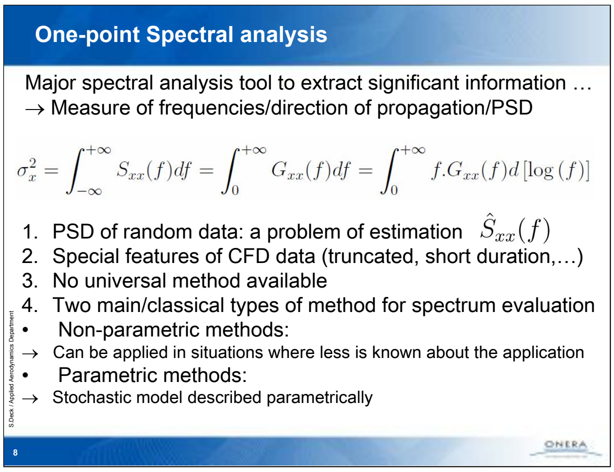

Contents
1. Classification of unsteady approaches and validation levels
2. Special features of signal issued from CFD & test case
3. Spectral analysis (single-point, DMD)
4. Further discussion
5. Conclusions
3


Classification of levels of validation
DNS
LES
Hybrid RANS/LES
Unsteady RANS
RANS
Modeling level ≡ Compromise solution (physics / CPU cost)
RANS
Advanced methods
Level of validation
1 Steady Integral forces (lift, drag, pitch,..)
2 Mean aerodynamic field (velocity, pressure profiles,…)
3 Second order statistics (rms quantities, Reynolds stresses, …)
4 One-point spectral analysis (Power Spectral Densities)
5 Two-point spectral analysis (correlation, coherence and phase spectra)
6 High-order and time-frequency analysis (Time-Frequency, bi-coherence spectra)
Challenge of handling complex geometries relates not only to CPU power!
Induced needs: → solution quality in terms of meshing and validation of the calculation
→ getting physical insight
Disappointingly, many authors present only one-point and first/second order statistics ! and paradoxical …
from Sagaut P., Deck S., Phil. Trans. R. Soc A, 367, 2009
4


Contents
1. Classification of unsteady approaches and validation levels
2. Test case & Special features of signal issued from CFD
3. Spectral analysis (single-point, DMD)
4. Further discussion
5. Conclusions
5


Test case
ONERA S3Ch AXI CONF N°4
(M_∞ = 0,7 - L / D = 1,2 - $Re_D \approx 1,2 \times 10^6$)
$N_{xyz}=12.10^6$ pts.
Turbulence modelling: ZDES[1]
Physical analysis: Weiss et al.[2]
I
II
III
IV
y/D
[1] Deck S., Theo. and Comp. Fluid Dyn.,2011
[2] Weiss et al., Phys of Fluids, 2009
6


Special feature of signals issued from CFD
Parameters | Duration (T) | Sampling frequency (f_S) | Δt_{CFD}
Experiment | 12 s | 10 kHz | /
ZDES (i.e. CFD) | 0.2 s | 100 kHz | 2.10^{-6} s
CFD
Exp.
ZDES
P'(t)/q₀
t(s)
• On the numerical side, classical experimental sampling durations look like 'eternity'
► Experiments easily access low-frequency ranges
• The numerical sampling frequency rate is often one order of magnitude higher than in the experiments
► CFD easily accesses high-frequency ranges
7


One-point Spectral analysis
Major spectral analysis tool to extract significant information …
→ Measure of frequencies/direction of propagation/PSD
$\sigma_x^2 = \int_{-\infty}^{+\infty} S_{xx}(f)df = \int_0^{+\infty} G_{xx}(f)df = \int_0^{+\infty} f.G_{xx}(f)d[\log(f)]$
1. PSD of random data: a problem of estimation $\hat{S}_{xx}(f)$
2. Special features of CFD data (truncated, short duration,…)
3. No universal method available
4. Two main/classical types of method for spectrum evaluation
• Non-parametric methods:
→ Can be applied in situations where less is known about the application
• Parametric methods:
→ Stochastic model described parametrically
8

One-point Spectral analysis (cont'd)

- Definition of PSD for a (WSS) random process

$$S_{xx}(f) = \lim_{T\to\infty} \frac{1}{T} E\left[|X(f,T)|^2\right] \quad X(f,T) = \int_0^T x(t)e^{-2j\pi f t}dt$$

- A first (simple but not consistant) approximation: the periodogram

$$\hat{S}^{per}_{xx}(f) = \frac{1}{T}|X(f,T)|^2$$

→ variance of the coefficient does not decrease when sampling time is increased
→ an averaged variant must be used

- Welch's averaged periodogram

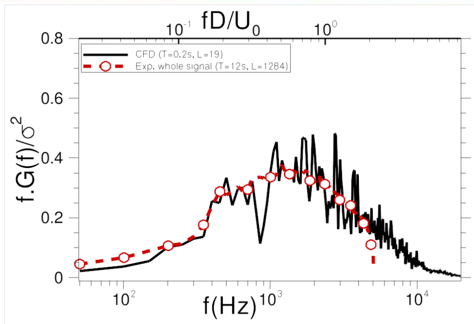
→initial sample set split into *L* non-overlapping blocks with length *M*: $\tilde{x}^{(i)}_M$
→additional window function w_M

$$\hat{S}^{welch}_{xx}(f) = \frac{\Delta t}{WL}\sum_{i=0}^{L-1}\frac{|\tilde{X}^{(i)}_M(f)|^2}{M} \quad W = (1/M)\sum_{n=0}^{M-1} w^2_M[n]$$

→compromise between frequency resolution (low *L*) and estimator variance (large *L*)

9

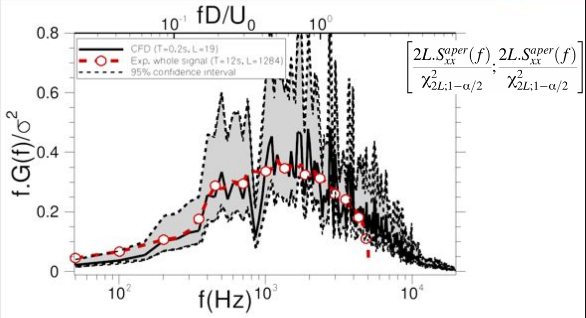
One-point Spectral analysis (cont'd)



→ inaccurate simulation?
→ fair comparison?

10

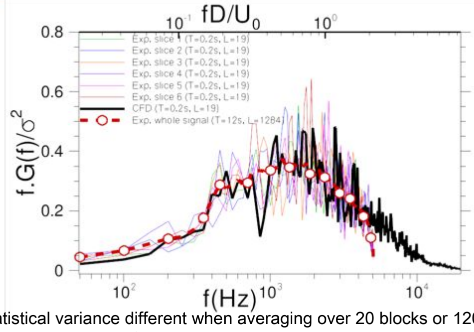
One-point Spectral analysis (cont'd)



→ confidence intervals should be always plotted on PSD !

11

One-point Spectral analysis (cont'd)



→ Statistical variance different when averaging over 20 blocks or 1200 blocks!
→ Splitting Exp. Data of time sequences into several shorter sample sets
→ CFD & Exp. Should be post-processed in the same way (Fair comparison)

12

A glimpse at Dynamic Mode Decomposition

- Proposed by Schmid(2010) & Rowley et al. (2009)
- Technique to process a sequence of snapshots (Exp/CFD) sampled in constant intervals $\Delta t_S = 1/f_S$
- Underlying mathematics based on Koopman modes which provides a linear representation of a nonlinear dynamical system

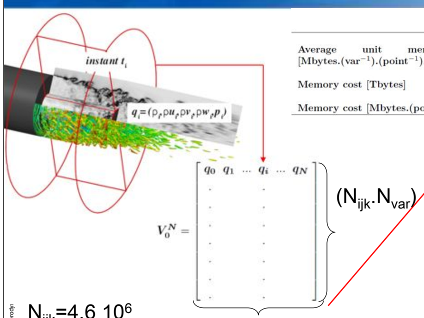
$$\mathcal{K}g(\boldsymbol{q_i}) = g(\mathcal{F}(\boldsymbol{q_i})) = g(\boldsymbol{q_{i+1}})$$

Koopman operator — observable of the flow field

- Capability to extract the spatial organisation and dynamics of one particular frequency (not trivial for broad banded spectra…)
- N snapshots: computation of an approximate Koopman operator (eigenvalues & eigen modes)
- Assumption: linear combination between snapshot *n* and the *n-1* previous
- coefficients stored in ***C*** *(companion matrix) to be 'eigen decomposed'*

13

A glimpse at Dynamic Mode Decomposition (cont'd)



	flow type file ascii	bin
Average unit memory cost [Mbytes.(var^{-1}).(point^{-1}).(snapshot^{-1})]	$1.64 \cdot 10^{-5}$	$3.92 \cdot 10^{-6}$
Memory cost [Tbytes]	10.1	2.4
Memory cost [Mbytes.(point^{-1})]	2.3	0.5

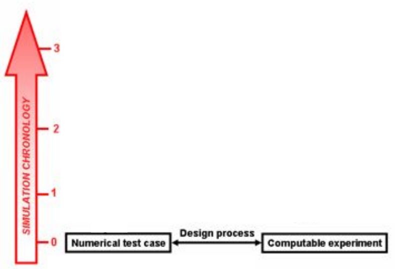
N_{ijk}=4.6 10^6
N=5 variables
T=0.2s →N+1=20000 snapshots

Post-processing has to be Pre-thought!

14

Further discussion

Multi-skilled need for future CFD



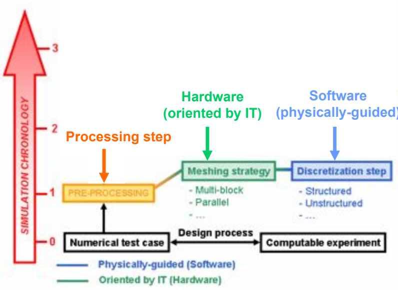
•**Chronology of a CFD simulation:**

0) **Definition of a numerical test case** from a well-documented computable experiment

15

Further discussion

Multi-skilled need for future CFD



•**Chronology of a CFD simulation:**

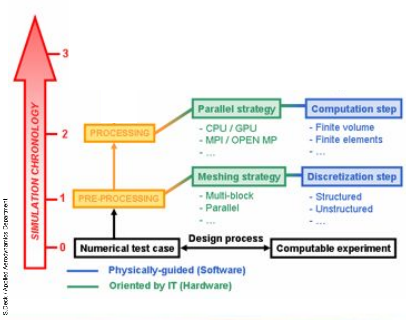
0) **Definition of a numerical test case** from a well-documented computable experiment

1) **Meshing strategy** : proper equilibrium on cores needs to be determined

16

Further discussion

Multi-skilled need for future CFD



•Chronology of a CFD simulation:

0) **Definition of a numerical test case** from a well-documented computable experiment

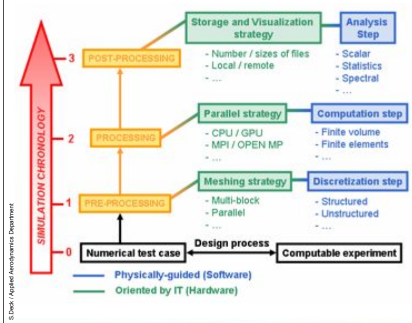
1) **Meshing strategy** : proper equilibrium on cores needs to be determined

2) **Parallel strategy :** number of cores, way of parallelism, types of processors

17

Further discussion

Multi-skilled need for future CFD



•Chronology of a CFD simulation:

0) **Definition of a numerical test case** from a well-documented computable experiment

1) **Meshing strategy** : proper equilibrium on cores needs to be determined

2) **Parallel strategy :** number of cores, way of parallelism, types of processors

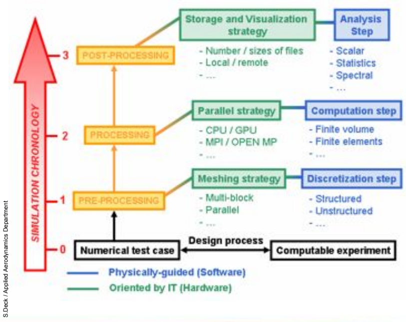
3) **Storage and visualization capabilities :**

Bad assessment of the post-processing feasibility can lead to the impossibility to analyze the results

18

Further discussion

Multi-skilled need for future CFD



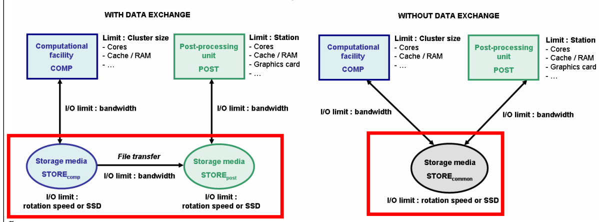
•Fluid mechanics theoretical background and discretization scheme knowledge are no longer sufficient to perform CFD (BUT OBVIOUSLY STILL MANDATORY !!!)

► **Growing need to better know the available hardware and its real performance**

19

Further discussion

Data transfer / load limit: Finding out the bottleneck



•From the IT point of view the limiting entities of the different processing steps have to be clearly identified

► **These entities correspond to the bottleneck of a network**

•The order of magnitude of the **data flow rate** and the **time to access memory** for each entity ('COMP', 'POST' and 'STORE') has to be assessed

► **The storage medium (HDD or flash disk/SSD) is almost always the bottleneck**

20

Further discussion

- **Characteristic phenomenom frequency :**
  - Vortex shedding : $f_{VS} \approx 450$ Hz => $T_{VS} \approx 2$ ms
  - To plot a spectral map at the extension wall :
    - Sampling 100 T_{VS} means :
      - storing 100 000 files (each timestep) => 300 Gb
      - storing 50 000 files (every two timesteps) => 150 Gb
- **Reminder:** the bottleneck is the storage medium
  - Every timestep a 'small' file gathering the values at the wall is written
  - With 3 Mb per map to store the data transfer falls from O(100) Mb/s to O(1) Mb/s
  - **Preconditioning is then mandatory** to reduce the number of files
  - → Loading time can be significantly reduced

$$t_{post\text{-}processing} = t_{data} + t_{algorithm}$$

with $t_{data} = t_{transfer}$ (42 h) + $t_{preconditioning}$ (45h) + t_{read} (3h) + t_{write} (3h)

- **In the present case : $t_{data} \sim 4$ days and $t_{algorithm}$ can strongly vary (up to 1 day)**

WARNING : An assumption is that all preconditioning and post-processing routines have already been developed and validated !

21

Conclusion

- A fair comparison of CFD and experiments is by far not trivial
- Simulation and experiments have to be post-processed the same way
- Advanced simulation = numerical experiment
  → Deep physical analysis is possible / mandatory
- Concerning unsteady simulations on HPC :
  - CPU resources : sampling duration Vs race the biggest number of points !
  - Large data manipulation is not a trivial issue
- Multi-skilled is needed in addition to the classical background of CFD : Signal processing, IT knowledge
- HPC is definitly no longer a one person job

22

2.14 K. Hillewaert

(Cenaero)

New challenges and opportunities created by high order discretization schemes for industrial flows

K. Hillewaert, C. Carton de Wiart and P. Geuzaine

Cenaero, Belgium

Onera Scientific Day – October 3, 2012

The simulation of turbulent flows by *Direct Numerical Simulation (DNS)* and *Large-Eddy Simulation (LES)* approaches requires extremely low numerical dispersion and dissipation errors. Recently *finite element (FEM)*-like high-order methods such as the *discontinuous Galerkin method (DGM)* [1, 2, 3], the *spectral difference method (SDM)* [4, 5] and the *spectral element method (SEM)* [6, 7] have been applied to such computations. The main motivation is that these methods bridge the gap between the high accuracy – deemed mandatory for adequate resolution of the turbulent structures – of academic solvers and the geometric flexibility of industrial solvers. Next to very interesting dispersion and dissipation properties, DGM offers a simple way of checking grid resolution. Finally excellent serial and parallel computational efficiencies are obtained.

The aforementioned advantages potentially make DGM a powerful tool for high fidelity simulation of turbulent flows in complex geometry. The talk will discuss recent developments performed at Cenaero and related to this new technology. Sample applications will be presented.

References

[1] Uranga, A., Persson, P.-O., Drela, M., and Peraire, J., 2009. “Implicit Large Eddy Simulation of Transitional Flows over Airfoils and Wings”. In Proceedings of the 19th AIAA Computational Fluid Dynamics, no. AIAA 2009-4131.

[2] van der Bos, F., and Geurts, B., 2010. “Computational error-analysis of a discontinuous Galerkin discretization applied to large-eddy simulation of homogeneous turbulence”. *Computer Methods in Applied Mechanics and Engineering.,* ***199***(13-16), pp. 903–915.

[3] L. Wei, L., and Pollard, A., 2011. “Direct numerical simulation of compressible turbulent channel flows using the discontinuous Galerkin method”. *Computers and Fluids,* ***47***, pp. 85–100.

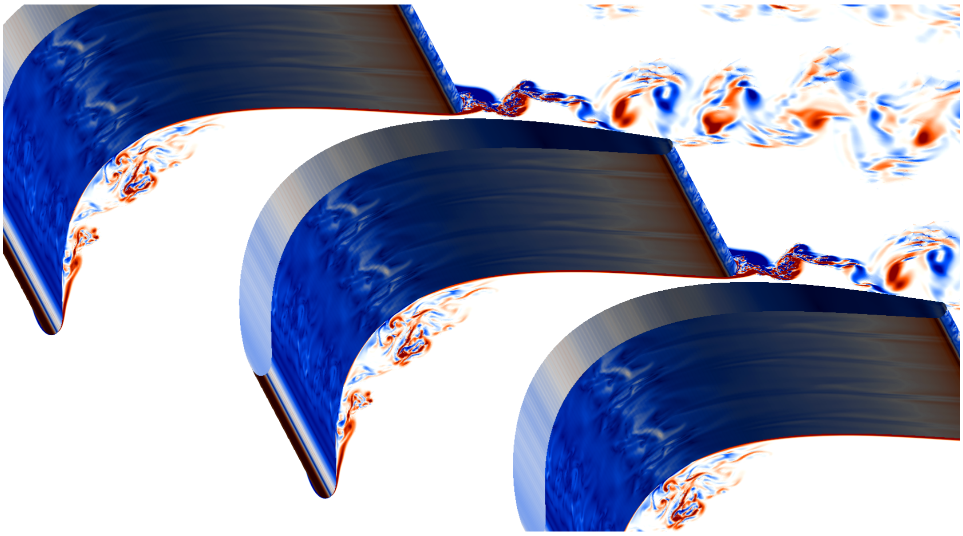


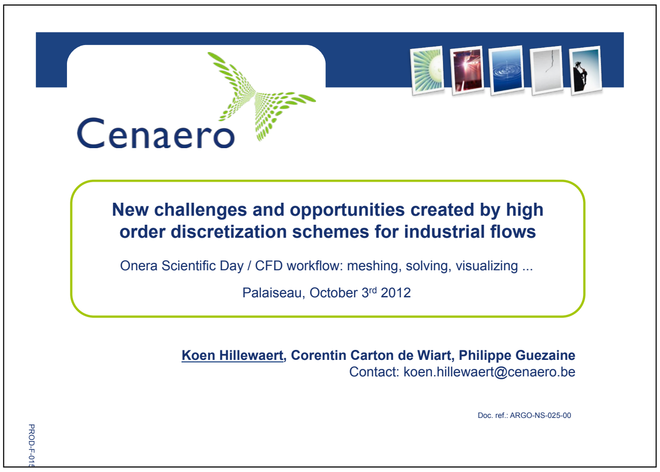
Figure 1: DNS of a low pressure turbine blade at $R_e = 85000$: spanwise component of the vorticity at the periodic boundary and skin friction on the blade surface.

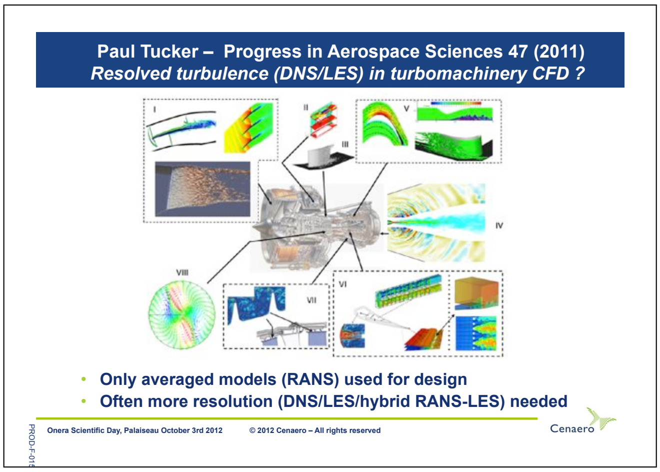
[4] Zhou, Y., and Wang, Z., 2011. “Effects of Surface Roughness on Laminar Separation Bubble over a Wing at a Low-Reynolds Number”. In Proceedings of the 49th AIAA Aerospace Sciences Meeting including the New Horizons Forum and Aerospace Exposition, no. AIAA 2011-736.

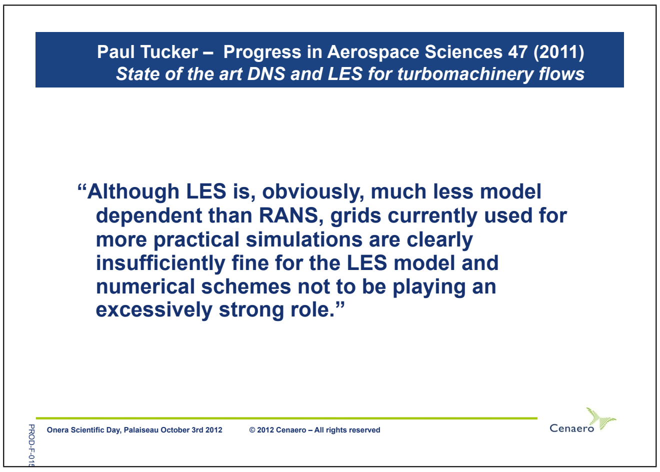
[5] Liang, C., Premasuthan, S., and Jameson, A., 2009. “Large Eddy Simulation of Compressible Turbulent Channel Flow with Spectral Difference method”. In 47th AIAA Aerospace Sciences Meeting, no. AIAA 2009-402.

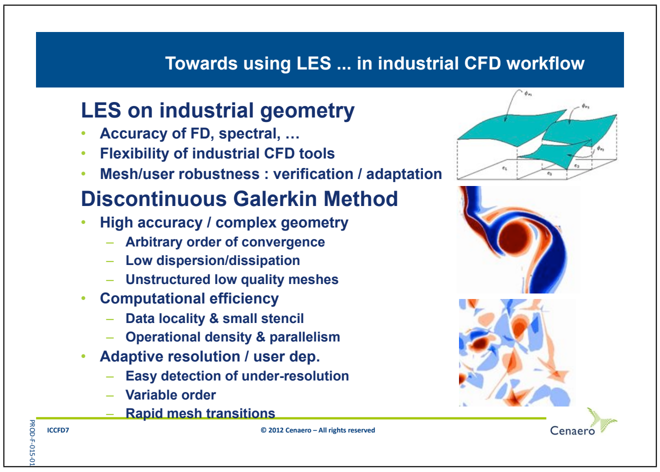
[6] Ohlson, J., 2009. “Spectral-element simulations of separated turbulent internal flows”. PhD thesis, Kungliga Tekniska Hogskolan Stockholm.

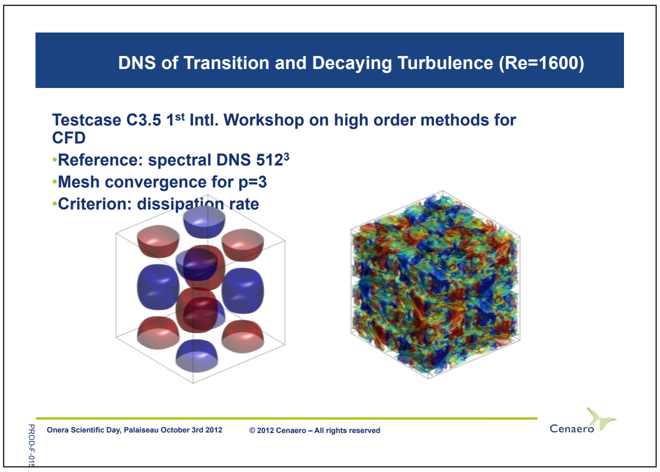
[7] Wasberg, C., Gjesdal, T., Pettersson Reif, B., and Andreassen, O., 2009. “Variational multiscale turbulence modelling in a high order spectral element method”. *Journal of Computational Physics*, ***228***, pp. 7333–7356.

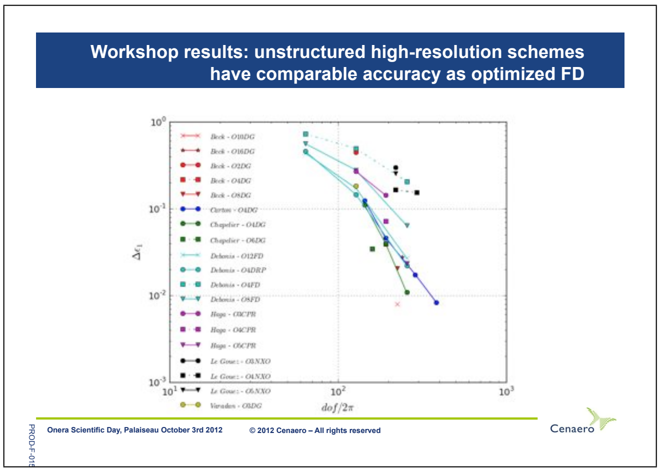

Cenaero
New challenges and opportunities created by high order discretization schemes for industrial flows
Onera Scientific Day / CFD workflow: meshing, solving, visualizing ...
Palaiseau, October 3rd 2012
Koen Hillewaert, Corentin Carton de Wiart, Philippe Guezaine
Contact: koen.hillewaert@cenaero.be
Doc. ref.: ARGO-NS-025-00

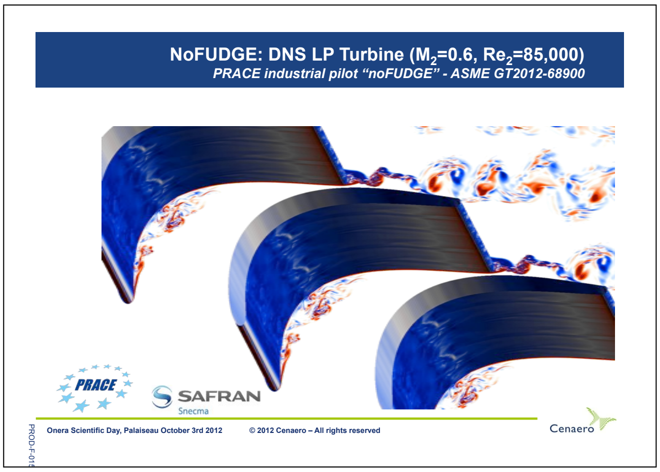

Paul Tucker – Progress in Aerospace Sciences 47 (2011)
Resolved turbulence (DNS/LES) in turbomachinery CFD ?
Only averaged models (RANS) used for design
Often more resolution (DNS/LES/hybrid RANS-LES) needed
Onera Scientific Day, Palaiseau October 3rd 2012
© 2012 Cenaero – All rights reserved
Cenaero

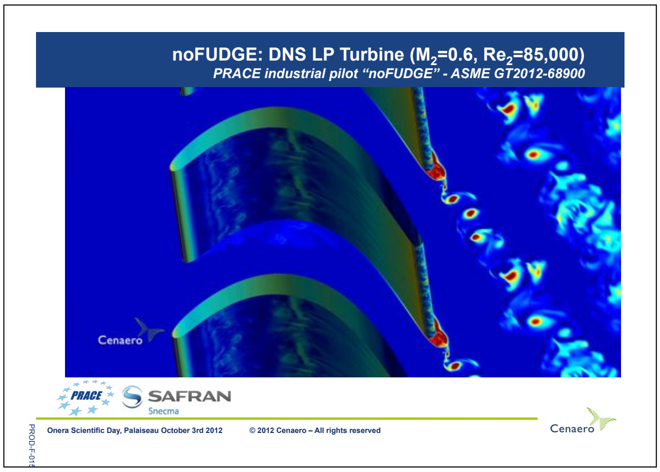

Paul Tucker – Progress in Aerospace Sciences 47 (2011)
State of the art DNS and LES for turbomachinery flows
"Although LES is, obviously, much less model dependent than RANS, grids currently used for more practical simulations are clearly insufficiently fine for the LES model and numerical schemes not to be playing an excessively strong role."
Onera Scientific Day, Palaiseau October 3rd 2012
© 2012 Cenaero – All rights reserved
Cenaero


Towards using LES ... in industrial CFD workflow
LES on industrial geometry
Accuracy of FD, spectral, ...
Flexibility of industrial CFD tools
Mesh/user robustness : verification / adaptation
Discontinuous Galerkin Method
High accuracy / complex geometry
Arbitrary order of convergence
Low dispersion/dissipation
Unstructured low quality meshes
Computational efficiency
Data locality & small stencil
Operational density & parallelism
Adaptive resolution / user dep.
Easy detection of under-resolution
Variable order
Rapid mesh transitions
ICCFD7
© 2012 Cenaero – All rights reserved
Cenaero

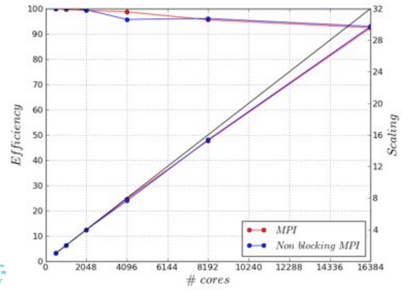

DNS of Transition and Decaying Turbulence (Re=1600)
Testcase C3.5 1st Intl. Workshop on high order methods for CFD
Reference: spectral DNS 512³
Mesh convergence for p=3
Criterion: dissipation rate
Onera Scientific Day, Palaiseau October 3rd 2012
© 2012 Cenaero – All rights reserved
Cenaero


Workshop results: unstructured high-resolution schemes have comparable accuracy as optimized FD
10⁰
10⁻¹
10⁻²
10⁻³
10¹
10²
10³
dof/2π
Onera Scientific Day, Palaiseau October 3rd 2012
© 2012 Cenaero – All rights reserved
Cenaero

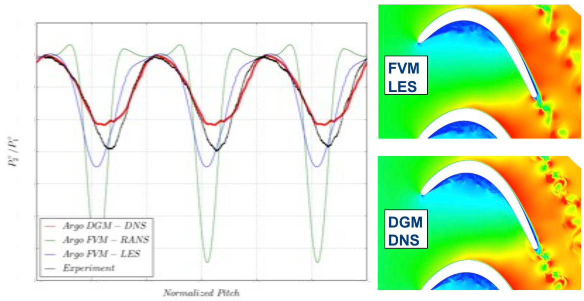

NoFUDGE: DNS LP Turbine (M2=0.6, Re2=85,000)
PRACE industrial pilot "noFUDGE" - ASME GT2012-68900
PRACE
SAFRAN
Snecma
Onera Scientific Day, Palaiseau October 3rd 2012
© 2012 Cenaero – All rights reserved
Cenaero


noFUDGE: DNS LP Turbine (M2=0.6, Re2=85,000)
PRACE industrial pilot "noFUDGE" - ASME GT2012-68900
Cenaero
PRACE
SAFRAN
Snecma
Onera Scientific Day, Palaiseau October 3rd 2012
© 2012 Cenaero – All rights reserved
Cenaero

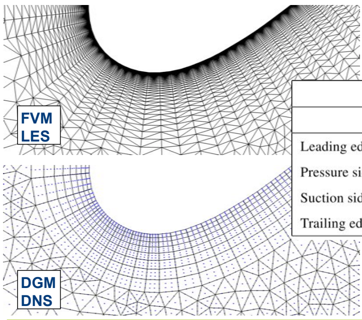
Weak Scaling on Juelicher BlueGene/P (OpenMP+MPI)



Total Pressure in the Wake



FVM (KE-scheme) Mesh vs DGM (Cubic Polynomial) Mesh



	FVM		DGM		
	$x^+ = z^+$	y^+	x^+	z^+	y^+
Leading edge	35	0.5	15	30	3
Pressure side	5	0.1	3	3	0.3
Suction side	50	0.3	15	15	1.5
Trailing edge	10	0.3	2	10	1

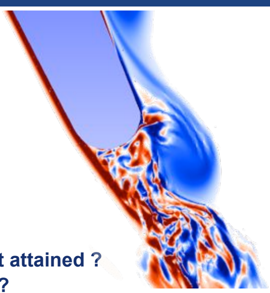
8.5M dof/variable FVM LES
15M dof/variable DGM DNS

(Rough) Cost Estimate (FVM on Intel; DGM on BlueGene/P)

	FVM LES	DGM DNS
Order of accuracy	2	4
# dof / variable (M dof)	8.5	15
CPU time for one t_c (kCPUh)	11	112
Memory per core (MB)	700	500
Number of cores	256	4096
CPU time / # dof	0.0013	0.0075
Memory / # dof	0.02	0.036

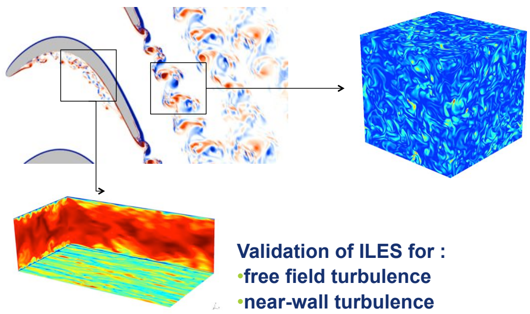
- **Intel ~ 4 times faster than BlueGene/P**
- **DGM 40% more expensive in equivalent CPU / dof**
- **But, DGM more accurate, more stable and less dissipative**

Where can we still improve ?



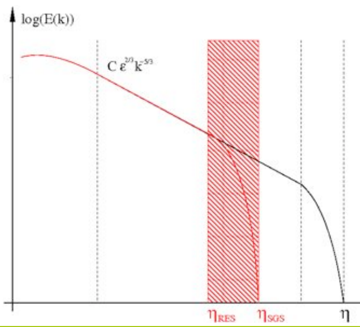
Reasons ?
- **Statistical convergence not attained ?**
- **Inlet turbulence neglected ?**
- **Slight under-resolution in TE region ? → adaptivity**

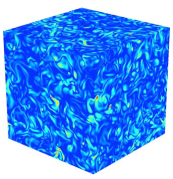
Validation on canonical testcases



Validation of ILES for :
- **free field turbulence**
- **near-wall turbulence**

Large Eddy Simulation





HIT at infinite Reynolds number

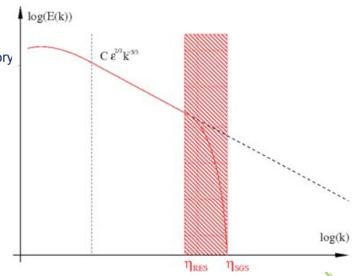
- After transient phase:
  - Temporal: kinetic energy

$$E(t) = E(0)t^{-p}$$

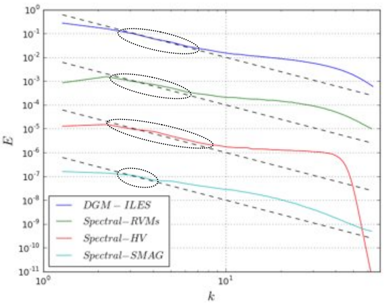
  - Spectral: Kolmogorov's theory

$$E(k) = C\epsilon^{\frac{2}{3}}k^{-\frac{5}{3}}$$

- Computational set up
  - DGM (p=4), N=643, 1283
  - Explicit time (RK4)
  - Initial solution from spectral computation
- Comparison
  - state-of-the-art SGS
  - spectral code
  - equivalent resolution

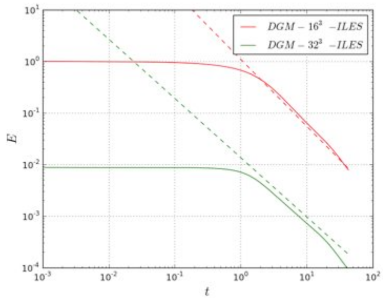


Spectral Distribution of Energy



- Smagorinsky model is not able to capture the formation of the inertial range
- DGM/ILES has same behavior as RVMs and HV model
- Is it the same for wall-bounded flows?

Temporal evolution of kinetic energy



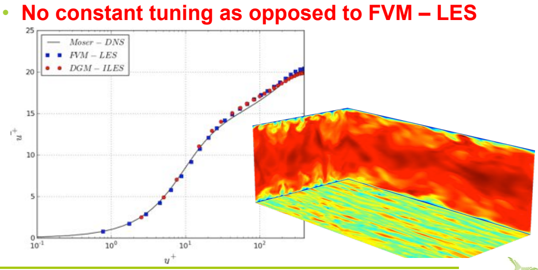
Slopes in agreement with theoretical value

$$E(t) = E(0)t^{-p}$$

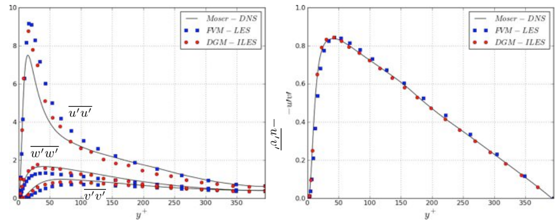
with p = 1.3±0.1

Implicit LES (or Truncated DNS) of Channel Flow (Re_τ=395)

- Quartic polynomials
- 12 x 12 x 16 mesh ~ 48 x 48 x 64 dof
- No constant tuning as opposed to FVM – LES

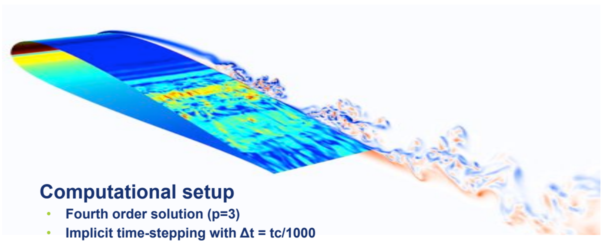


Velocity Fluctuations



DNS and ILES of transitional flow SD7003 airfoil at Re=60k with α=4° AoA

Case C3.3 1st Intl Workshop on High Order Methods in CFD



Computational setup

- Fourth order solution (p=3)
- Implicit time-stepping with Δt = tc/1000
- Newton-Krylov with Jacobi preconditioner
- Span 20% of the chord

DNS and ILES of transitional flow SD7003 airfoil at Re=60k with α=4° AoA

Case C3.3 1st Intl Workshop on High Order Methods in CFD

Computational setup

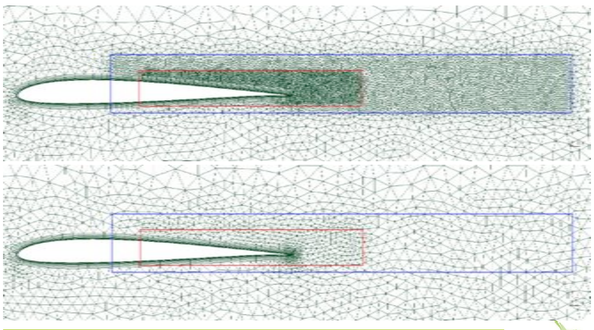
- Fourth order solution (p=3)
- Implicit time-stepping with Δt = tc/1000
- Newton-Krylov with Jacobi preconditioner
- Span 20% of the chord

Mesh characteristics

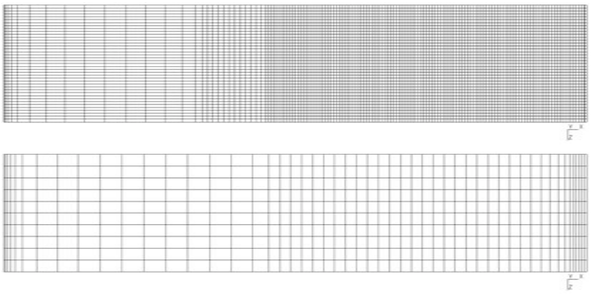
- Extrusion of a 2D mesh (hex, wedges)
- Uniform mesh in turbulent region (Δx=Δz)
- LES mesh is 4 times coarser in turbulent region

	DNS	LES
$\Delta y_0/c$ (wall-normal)	$3.33\ 10^{-4}$	$3.33\ 10^{-4}$
$\Delta x/c$ (box 1)	$1.67\ 10^{-3}$	$6.67\ 10^{-3}$
$\Delta x/c$ (box 2)	$3.33\ 10^{-3}$	$1.33\ 10^{-2}$
$\Delta z/c$ (spanwise)	$1.67\ 10^{-3}$	$6.67\ 10^{-3}$
y^+ at $x/c = 0.8$	1.2	1.2
$x^+ = z^+$ at $x/c = 0.8$	6	24
Number of hexahedra (/1000)	84.7	8.7
Number of wedges (/1000)	646.1	47.9
Total number of dof per variable (at continuity) [k]	10934.3	874.5

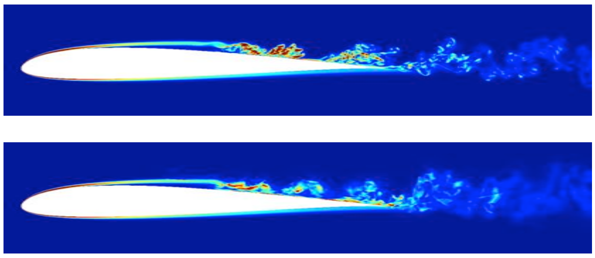
2D mesh on the periodic plane



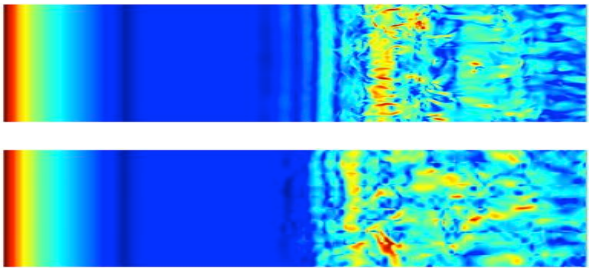
Span- and chordwise resolution



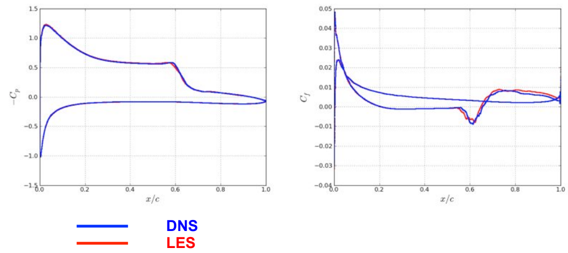
vorticity



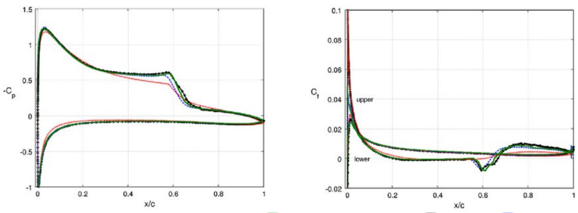
Skin friction



Comparison LES and DNS



Comparison with Literature



	DNS/DGM	ILES/DGM	Uranga [11]	Visbal [12]
$\overline{C_L}$	0.196	0.201	0.22	-
$\overline{C_D}$	0.602	0.607	0.603	-
Separation	0.209	0.207	0.21	0.23
Reattachment	0.654	0.647	0.65	0.67
Cost to compute one t_c [CPUh]	11001	415	-	-

Conclusions and perspectives

- DGM good candidate for industrial resolved turbulence
  - high order obtained on unstructured meshes
  - viable method for DNS of transitional regimes
  - implicit way of checking for resolution → robustness
  - ILES would remove need for model tuning ?
  - efficient use of large scale HPC resources
- Important gains in efficiency can still be achieved
  - hp-adaptive strategy
  - iterative strategies → multigrid algorithms
- Interaction turbulence and high-order communities needed
  - Implementation and assessment of modeling approaches
  - Reduced dependence on numerical methods
  - shock capturing and interaction with LES modeling
- Peripheral technology needs to be developed (Gmsh?)
  - creation of the curvilinear meshes
  - visualisation of large high-order data sets

2nd workshop HOM for CFD, Cologne, May 27-28 2013

Acknowledgments

- Contributors and colleagues
  - Bastien Gorissen - mesh generation
  - Guillaume Verheylewegen (UCL) - shock capturing
  - Marcus Drosson (ULg/Umicore) - RANS models
  - Pierre Schrooyen (UCL) - interaction turbulence and ablation
- Collaborations
  - Jean-François Remacle (UCL) - Gmsh and DGM
  - Grégoire Winckelmans (UCL) - fundamental turbulence
  - Laurent Bricteux (UMons) - fundamental turbulence
  - Christophe Geuzaine (ULg) - Gmsh
- Computational grants
  - DEISA DECI project CoBaULD - transition on E387 airfoil
  - PRACE industrial pilot noFUDGE - LP turbine
- Funding projects
  - FP6 research project ADIGMA
  - FP7 project IDIHOM
  - ERDF and ESF funding (contract N○ EP1A122030000102)

2.15 Y. Fournier (EDF)

Evolving the *Code_Saturne* and NEPTUNE_CFD solver toolchains for billion-cell calculations

Y. Fournier

EDF R&D, Fluid Dynamics, Power Generation, and Environment Department,
6 quai Watier –BP 49 – 78401 Chatou Cedex

EDF has been developing the *Code_Saturne* CFD solver since 1997, and co-developing the NEPTUNE_CFD solver, which shares much of the same architecture, since the early 2000's. These codes, though not ancient, have seen changes in the dominant computer architectures, from vector machines to clusters to a mix of clusters and supercomputers from the IBM Blue Gene or Cray XT/XE series.

As these codes are intended for often complex industrial studies, they must meet the competing goals of being relatively easy to set up, to reduce the risk of user error, while allowing very fine-grained control, for complex situations. This is ensured by providing both a graphical user interface for the base setup and a wide array of user subroutines for finer grained setup. Also, some mesh preprocessing tools are provided, such as mesh joining, and their use must also be as simple as possible. For better evolution, we allow for major revisions (every 2 years or so) to break the user data setup, but functionality must not be lost.

As users are not expected to be computer science or parallelism specialists, we have strived to make the toolchain as transparent as possible relative to HPC aspects, using the same scripts from a laptop to a supercomputer. This has required parallelizing a growing portion of the toolchain, especially pre and post-processing aspects.

Both codes are routinely used for meshes approaching 200 million cells, and *Code_Saturne* has already been tested up to 3.2 billion cells.

To reduce the volume of data produced, users may already extract end postprocess partial data, and in the near future, we expect to add co-visualization or in-situ visualization possibilities, so as to reduce the volume of data that must be both output and archived.

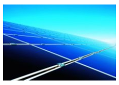
In this talk, we will briefly explain how our tools have evolved, and focus on current and future evolutions to ease the usage of larger and larger datasets.

Evolving the *Code_Saturne* and NEPTUNE_CFD toolchains for billion-cell calculations

Yvan Fournier, EDF R&D
October 3, 2012

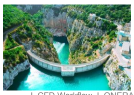
SUMMARY

Simulate and decide



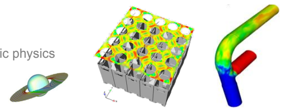

- Power generation
  - Improve the **efficiency** and **safety** of our facilities
  - Optimization of maintenance and life span
  - Response to specific events (flood, heat wave, incidents,…)
- Preparation of the future
  - New technologies for power generation
  - Innovation in renewable energies and storage
  - Anticipation of climate constraints on shared resources


- Promotion of sustainable development
  - Help customers optimize their energy consumption
- Global Challenges
  - Evaluation and reduction of environmental impact, waste management
- Development of future simulation tools

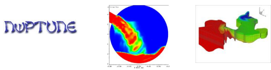



Code development at EDF R&D (1/2)

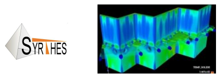
- *Code_Saturne*
  - general usage single phase CFD, plus specific physics
  - property of EDF, open source (GPL)
  - http://www.code-saturne.org



- NEPTUNE_CFD
  - multiphase CFD, esp. water/steam
  - property of EDF/CEA/AREVA/IRSN



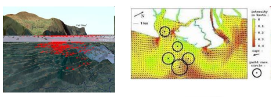
- SYRTHES
  - thermal diffusion in solid and radiative transfer
  - property of EDF, open source (GPL)
  - http://rd.edf.com/syrthes



- *Code_Aster*
  - general usage structure mechanics
  - property of EDF, open source (GPL)
  - http://www.code-aster.org

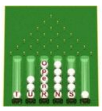
Code development at EDF R&D (2/2)

- TELEMAC system
  - free surface flows
  - Many partners, mostly open source (GPL, LGPL)
  - http://www.opentelemac.org



- SALOME platform
  - integration platform (CAD, meshing, post-processing, code coupling)
  - property of EDF/CEA/OpenCascade, open source (LGPL)
  - http://www.salome-platform.org

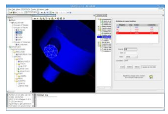
- Open TURNS
  - tool for uncertainty treatment and reliability analysis
  - property of EDF/CEA/Phimeca, open source (LGPL)
  - http://trac.openturns.org



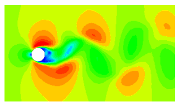
- and many others
  - Neutronics, electromagnetism
  - component codes, system codes
  - …

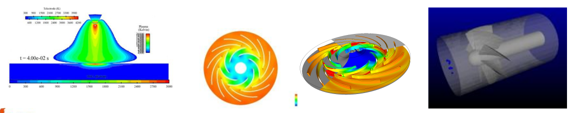
Code_Saturne EDF's general purpose CFD tool

- Co-located finite volume
  - Arbitrary unstructured (polyhedral meshes)



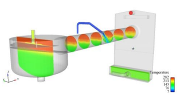
- Physical modeling
  - Laminar and turbulent flows: k-ε, k-ω SST, v2f, RSM, LES
  - Radiative transfer (DOM, P-1)
  - Coal, heavy-fuel and gas combustion
  - Electric arcs and Joule effect
  - Lagrangian module for particles tracking
  - Atmospheric modeling
  - ALE method for deformable meshes
  - Rotor / stator interaction for pumps modeling
  - Conjugate heat transfer (SYRTHES & 1D)





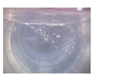
NEPTUNE_CFD EDF/CEA/AREVA/IRSN multiphase CFD code

- Development team: EDF and CEA
  - Development, validation, maintenance, installation, training, hot-line
  - Pioneering applications
- Users
  - EDF, CEA, IRSN, AREVA-NP
  - European project partners (NURISP)
  - Academic collaborations (IMFT)



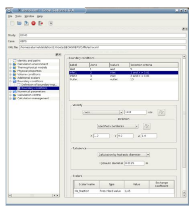
- Main features
  - 3D and local two-phase flow analysis
  - Generalized multi-field model
  - Physical models
    - Turbulence (k-ε and RSM)
    - Interfacial area and polydispersion models
    - Set of models for boiling bubbly flows
    - Set of models for stratified steam-water flows
    - Conjugate heat transfer

- Based on *Code_Saturne* infrastructure and toolchain
  - Different numerical scheme (similar solvers and gradient reconstruction)
  - not open-source (EDF/CEA/AREVA/IRSN co-development)

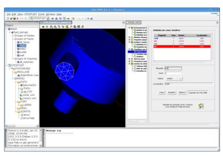


Required Environment

- Runs on all Unix-type systems (Linux, Unix, Mac OS-X)
  - Windows port underway
- Pre-requisites
  - compilers
    - C 99 (gcc, xlc, icc, …)
    - Fortran 95 (fgfortran, xlf, ifort, …)
  - for parallel computing
    - MPI library : Open MPI, MPICH2, …
    - METIS or SCOTCH partitioner (optional)
  - for GUI
    - Python, Qt4, SIP, libxml2
  - Optional data exchange libraries
    - MED, CGNS, libCCMIO

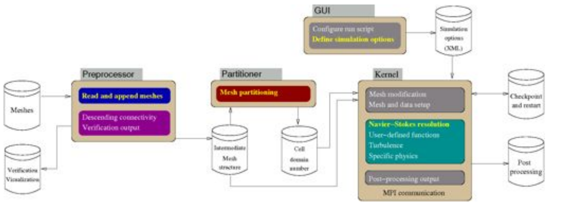


- Optional integration under the SALOME platform
  - GUI extensions
    - mouse selection of boundary zones
    - advanced user files management
  - from CAD to post-processing in one tool



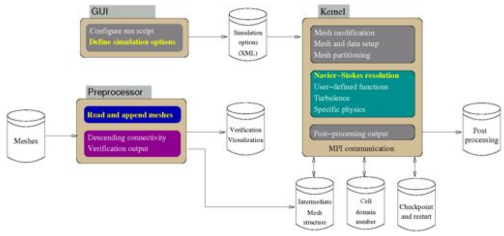
Code_Saturne / NEPTUNE_CFD 2010 toolchain

- Reduced number of tools
  - Each with rich functionality
  - Natural separation between interactive and potentially long-running parts
  - Some overlap between old/new algorithms
    - serial mesh joining in Preprocessor, parallel version in Kernel
  - In-line (pdf) documentation

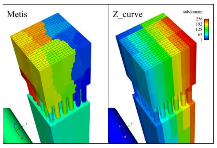


Code_Saturne / NEPTUNE_CFD 2012 toolchain

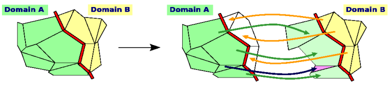
- Partitioner merged with code
  - Encourages parallel partitioning
  - Potentially, only 1 of every r ranks may participate
    - to avoid possible decreasing quality issues
  - Partitioning using PT-SCOTCH, ParMETIS, or Hilbert or Morton space-filling curve



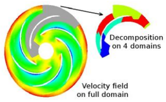
Parallelism and periodicity (1)



- Classical domain partitioning using MPI
  - Partitioning using METIS, SCOTCH or internal Morton or Hilbert space-filling curve
  - Classical « ghost cell » method for both parallelism and periodicity
    - Most operations require only ghost cells sharing faces
    - Extended neighborhoods for gradients also require ghost cells sharing vertices



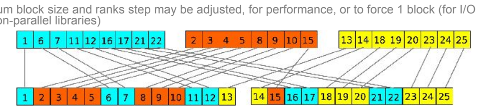
- Periodicity uses same mechanism
  - True geometric periodicity (not a BC)
  - Vector and tensor rotation may be required (semi-explicit component coupling in rotation)
- Input output is partition independent
- Currently adding OpenMP



Parallelism and periodicity (2)

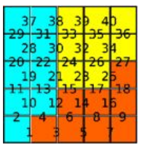
21	22	23	24	25
16	17	18	19	20
11	12	13	14	15
6	7	8	9	10
1	2	3	4	5

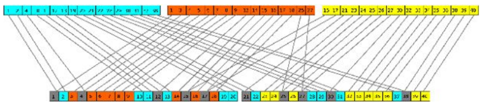
- Use of global numbering
  - We associate a global number to each mesh entity
    - A specific C type (cs_gnum_t) is used for this. An unsigned long integer (64-bit) is necessary for larger meshes
    - Currently equal to the initial (pre-partitioning) number
- Allows for partition-independent single-image files
  - Essential for restart files, also used for postprocessing output
  - Shared file MPI-IO possible does not require indexed datatypes
- Allows automatic identification of "Interfaces"
  - Matching between vertices on parallel boundaries
  - Allow summing contributions from multiple processes in a robust manner and in linear time
- Redistribution on n blocks
  - n blocks ≤ n cores
  - Minimum block size and ranks step may be adjusted, for performance, or to force 1 block (for I/O with non-parallel libraries)



Parallelism and periodicity (3)

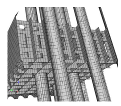
- Conversely, simply using global numbers allows reconstructing neighbor partition entity equivalents mapping
  - Used for parallel ghost cell construction from initially partitioned mesh with no ghost data
- Arbitrary distribution, inefficient for halo exchange, but allow for simpler data structure related algorithms with deterministic performance bounds
  - Owning processor determined simply by global number, messages are aggregated
- Switch from one representation to the other currently uses `MPI_Alltoall` and `MPI_Alltoallv`, but we may switch to a more "sparse" algorithm such as Crystalrouter if performance issues reuquire it
  - Not an issue under 16000 cores, not critical at 64000

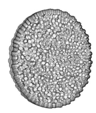




Supported Meshes

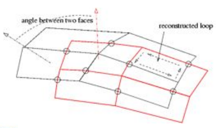
- Mesh generators
  - Simail: easy-to-use, with command file, but no CAD
  - I-deas NX
  - GAMBIT (Fluent), ICEM-CFD, ANSYS meshing
  - Star-CCM+
  - SALOME SMESH (http://www.salome-platform.org)
  - Gmsh,
  - Harpoon, ...
- Formats
  - MED, CGNS, Star-CCM+, Simail, I-deas universal,
  - GAMBIT neutral, EnSight Gold…
- Cells: arbitrary arrangement of polyhedra
  - For example: tetrahedra, hexahedra, prisms, pyramids, general n-faced polyhedra, ...

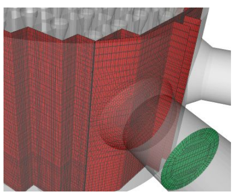




Mesh Joining (1)

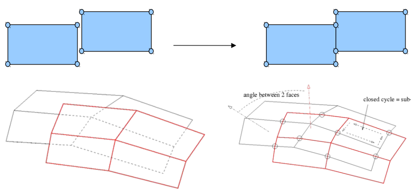
- Arbitrary interfaces: « any type of mesh / format » + « any type of mesh / format »
  - Meshes may be contained in one single file or in several separate files
  - Based on intersecting faces and reconstructing-sub-faces
  - Expertise may be required if arbitrary interfaces are used:
    - In critical regions
    - With LES
    - With very different mesh refinements
    - On curved CAD interfaces
  - Periodic matching builds on this





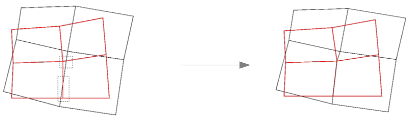
Mesh Joining (2)

- Parallelizing this algorithm requires the same main steps as the serial algorithm:
  - Detect intersections (within a given tolerance) between edges of overlapping faces
    - Uses parallel octree for face bounding boxes, built in a bottom-up fashion (no balance condition required)
    - Preprocessor version used a lexicographical binary search, whose best case was O(n.log(n)), and worst case was O(n2).
  - Subdivide edges according to inserted intersection vertices
  - Merge coincident or nearly-coincident vertices/intersections
    - Requires synchronized merge accept/refuse
  - Re-build sub-faces

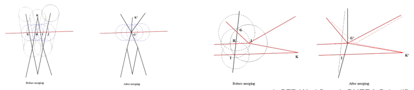


Mesh Joining (3)

- Tolerance merging of vertices allows for avoidal of small faces, though it may lead to higher warping
  - Warped faces may be split into triangles, though this has other disadvantages

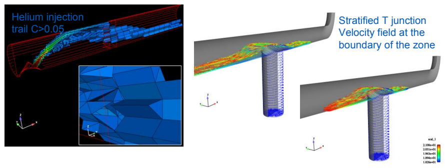


- The merging of vertices not in each other's tolerance neighborhood through merge combinations (transitivity) is normally detected and avoided
  - Merge chains may be broken above a selected threshold to avoid this
  - This may limit joinings when tolerance is set too high



Postprocessing output

- Users may define an arbitrary number of post-processing:
  - Writers
    - Choice of an output format (EnSight, MED, CGNS, CCMIO), format options, and output frequency
  - Meshes
    - Subsets of the computational mesh
    - Each associated to a list of writers
- This allows fine control of user output
  - Frequently output fields may be associated to partial data, on a possibly time-dependent mesh subset
    - Useful for example to output partial data at higher frequency so as to generate movies, with "reasonable" output volumes



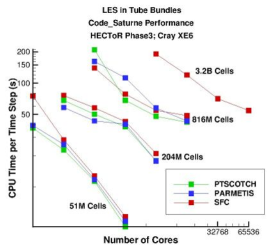
High Performance Computing with *Code_Saturne*

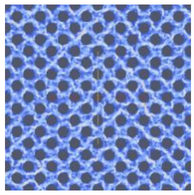
- *Code_Saturne* and NEPTUNE_CFD used extensively on HPC machines
  - EDF clusters (IBM Idataplex, Blue Gene/P; Blue Gene/Q)
  - CCRT calculation center (CEA based)
  - PRACE machines
    - (HECTor (EPCC), Jugene (FZJ), Curie (GENCI)
  - DOE machines
    - Jaguar (ORNL), Intrepid (Argonne)

- Tests run by STFC Daresbury up to 3.2 billion cells
  - intensive work on parallel optimization and debugging loop
- *Code_Saturne* used as reference in PRACE European project
  - reference code for CFD benchmarks on 6 large European HPC centers

Scalability of *Code_Saturne*

- Scalability as a function of mesh size
  - At 65 000 cores and 3,2 Billion cells, about 50 000 cells / core

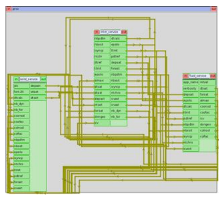




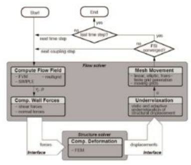
Parallel Code coupling

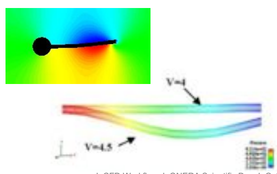
- Parallel n to p coupling using "Parallel Location and Exchange" sub-library
  - Uses MPI
  - Fully distributed (no master node requiring global data)
  - Sucessor to /refactoring of FVM (also used in Cwipi)
    - Core communication in PLE, rest moved to code
- SYRTHES (conjugate heat transfer)
  - Coupling with (parallel) SYRTHES 4 run on industrial study using 128+ processors
  - Coupling with (serial) SYRTHES 3 used older mechanism
- *Code_Saturne*
  - RANS/LES
  - Different turbulence models and time steps in fixed overlapping domains
  - Turbomachinery
    - Same turbulence model and time step, moving subdomain

Coupling using SALOME



- Fluid-Structure interaction through coupling with Code_Aster
  - Using the YACS module
  - Not in production yet





Next steps

- Add mesh multiplication (homogeneous mesh refinement)
  - Will reduce mesh memory pressure
    - May allow us to wait a bit longer for parallel meshers
  - Will allow computing on a fine mesh, visualization or postprocessing on a coarser mesh
    - May reduce I/O and archival pressure
    - Will need user acceptance/education...
  - Current developments under way, in the framework of the PRACE project
- Add parallel mesh boundary layer reconstruction
  - Will allow rebuilding a boundary layer directly in the code
    - No need to go back to the meshing tool
    - A same base mesh will be usable with different (high/low Reynolds) boundary layer modes
  - Currently being developed as part of a PhD at EDF
- Add in-situ visualization or co-visualization
  - Future writers need not write to file, but may output to other representations
    - MedCoupling or VTK (for co-visualization ?)
    - ADIOS ?
  - Targeting VTK, for use with ParaView, or the SALOME platform's ParaVis (based on the latter)
  - Preliminary low-level work for co-visualization done
    - Initial testing planned in 2013
  - Intended to reduce I/O and archival pressure
    - Possible computational steering benefits

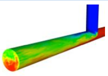
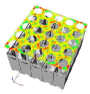
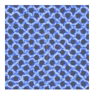
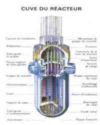
Future steps

- Pursue integration with the SALOME platform
  - Only integrate directly with components which are HPC compatible
    - Or in a manner compatible with HPC
    - Currently, Visualization (Paraview/ParaVis) is HPC compatible
    - Mesh is making progress
  - Coarser Integration (using files) for parts of the platform which are less HPC oriented
  - For future ensemble calculations, may benefit from OpenTurns integration for driving of uncertainty determination
- Optimize for future ensemble calculations
  - Using in memory data staging (avoiding files) with ADIOS, HDF5, or similar technologies may mitigate IO volumes
  - Pseudo code coupling (actually postprocessing coupling) may allow determine key statistics with less I/O and archival
    - This needs to be done in a relatively fault-tolerant manner
      - one run crashing must not cause the loss of the whole ensemble
      - Nesting ?

THANK YOU FOR YOUR ATTENTION

EXTRA SLIDES

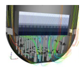
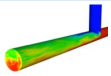
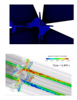
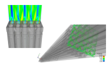
HPC timeline – Capability (peak) examples

2003	2006	2007	2012	2015
Consecutive to the Civaux thermal fatigue event Computations enable to better understand the wall thermal loading in an injection. Knowing the root causes of the event ⇒ define a new design to avoid this problem.	 Computation with an L.E.S. approach for turbulent modelling Refined mesh near the wall.	 Part of a fuel assembly 5x5 grid experimental mock-up	 LES of tube bundle	Complete reactor vessel 

Computations with smaller and smaller scales in larger and larger geometries
⇒ a better understanding of physical phenomena ⇒ a more effective help for decision making

10^6 cells	10^7 cells	10^8 cells	3.10^9 cells	5.10^{10} cells
Fujistu VPP 5000 1 of 4 vector processors 2 month length computation	Cluster, IBM Power5 400 processors 9 days	IBM Blue Gene/L 8 000 processors # 1 month	Cray XE6 4 000 cores # short test (< 1day)	? > 60 000 cores # 1 month
# 1 Gb of storage 2 Gb of memory	# 15 Gb of storage 25 Gb of memory	# 200 Gb of storage 250 Gb of memory	# 5 Tb of storage 3 Tb of memory	# 100 Tb of storage 60 Tb of memory
Power of the computer	Pre-processing not parallelized Power of the computer	Pre-processing not parallelized Mesh generation	*Mostly resolved* … ibid. … Scalability / Solver	*resolved* … ibid. … … ibid. … Visualisation

HPC timeline – Capacity (production) examples

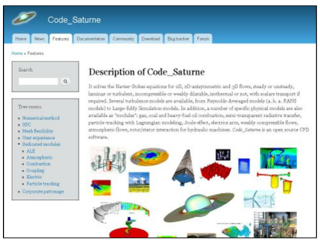
2003	2006	2009	2012	2015
 Lower core plenum flows, determining core inlet conditions Several applications, including dilution	 Computation with an L.E.S. approach for turbulent modelling Refined mesh near the wall.	 Part of a fuel assembly 5x5 grid experimental mock-up	 DNB study (NEPTUNE_CFD) 5x5 experimental grid, full mock-up height	**Ensemble calculations for uncertainty determination**

Computations on a wider range of geometry and flow conditions
⇒ a better understanding of physical phenomena ⇒ a more effective help for decision making

5.10^5 cells	10^7 cells	10^7 cells	2.10^8 cells	10^9 cells
Fujistu VPP 5000 1 of 4 vector processors 2 month length computation	Cluster, IBM Power5 400 processors 9 days	IBM Blue Gene/L 1 024 processors # 1 week/100 000 time steps	IBM Idataplex 2 000 cores # 2 months	? > 60 000 cores # 2 months
	meshing data transfer (for mesh and visualization)	… ibid. … … ibid. …	CAD … ibid. … fault tolerance (requires restarts)	*… ibid. …* *avoided (in situ)* … ibid. … ensemble data manipulation

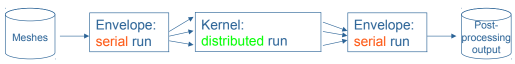
Code_Saturne open source practical info

- Distribution of Code_Saturne
  - GPL license, auxiliary library (PLE) under LGPL license
- Code_Saturne EDF website
  - http://code-saturne.org
  - source download
  - Code presentation and documentation
  - Contact with EDF development and support team
  - Code_Saturne news
  - Forum and bug-tracker



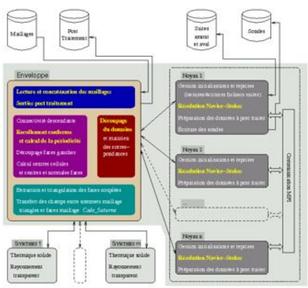
Parallelism history and roadmap (1)

- In recent years, HPC related work focused on parallelizing *Code_Saturne*, also benefiting NEPTUNE_CFD.
  - Initial prototype in 1997 and *Code_Saturne* V1.0 in 2000 were not parallel, but designed to be "parallel-compatible".
  - V1.1 released Nov 2004: parallel using domain splitting and MPI
    - 2-3 man-months Pre/post processor work starting mid 2001;
      - Includes METIS for domain partitioning;
      - Preprocessor limited to 1022 processor output (on Linux) due to limit on simultaneous number of open files
    - Parallelism operational after 2-3 mm Kernel work late 2001 / early 2002;
      - Restarts possible only on the same number of processors;
      - Single-image parallel restart system implemented spring 2003 (3 mm), allowing restarts on any number of processors;
  - Port to NEPTUNE_CFD summer 2003.



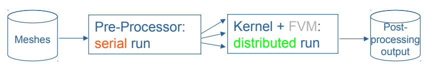
Parallelism history and roadmap (2)

- MPI (Message Passing Interface) calls not done directly in Fortran code, in which wrappers with simplified arguments are used.
- Messages between Kernel, Envelope, and Syrthes use a serial "portable" (big-endian) binary format, either using files, pipes, or MPI.
- SYRTHES coupling is also serialized through the Envelope module (requiring exchanges to be done using pipes or MPI, and not files).



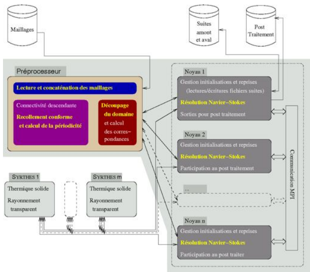
Parallelism history and roadmap (3)

- V1.2 released Nov 2006 introduced single-image parallel post-processor output as an option
  - Parallel ⌣ serial I/O aspects handled by preliminary FVM (Finite Volume Mesh) library, (parallel) successor to "Enveloppe" (Pre-Post adaptor);
    - Legacy post-processor mode using "Enveloppe" tool still used, as FVM was not fully featured;
  - Added "extended neighborhood" gradient reconstruction option, which introduced a second optional ghost cell set;
- V1.3 released Dec 2008 adds many HPC-oriented improvements
  - Post-processor output fully handled by FVM library;
  - Post-processor output functionality removed from "Enveloppe" module
    - renamed to "Pre-Processor";
    - allows removal of 1022 processor run limit of versions 1.1 and 1.2;
  - Ghost cell construction migrated from pre-processor to FVM / Kernel
    - parallelized, removed 1 memory bottleneck;
  - Pre-processor simplifications and optimizations
    - up to 40% gain in max memory use over version 1.2 in example cases



Parallelism history and roadmap (4)

- Coupling with SYRTHES is now handled directly by Kernel, removing Enveloppe bottleneck
  - Only coupled boundary faces by Code_Saturne and the required interpolation data are added to its memory footprint



Parallelism history and roadmap (5)

- By 2012, pre-processing was fully parallelized
  - 2007: Separated basic Pre-Processing from Partitioning (V1.4)
    - Allows running both steps on different machines
    - Partitionning requires less memory than pre-processing

- 2008-2010: Migration of advanced preprocessing from preprocessor to parallel kernel
  - Concatenation, joining of conforming/non-conforming meshes, periodicity

- July 2011status
  - A reduced subset of the current serial Pre-Processor survives as a mesh import tool
  - Optional graph-based partitioning done as a separate step, using PT-SCOTCH or ParMETIS

Parallelism history and roadmap (6)

- In 2013, the complete toolchain will be parallelized
  - Parallel mesh readers for major formats (at least MED 3.0, possibly CGNS 3.1)
  - A reduced subset of the current serial Pre-Processor may survive as a legacy mesh import tool

- Coupling status (May 2012)
  - SYRTHES 3.4: semi-distributed coupling (n / 1) using internal library
    - Each rank of Code_Saturne exchanges only local data
  - SYRTHES 4.0: fully distributed coupling (n / p) using PLE library
    - FVM, the precursor of PLE, is also available as an option in ParaMEDMEM (not in default builds), and by cwipi.
  - Code_Saturne: fully distributed coupling (n / p) using PLE library
    - Coupling of multiple Code_Saturne domains used for rotor/stator or RANS/LES coupling
  - Code_Aster: serialized coupling (n / 1 /1) using YACS/CALCIUM API
    - Only rank 0 comunicates using YACS, and gathers/scatters coupling data for other *Code_Saturne* ranks.
    - A switch to using ParaMEDMEM could be recommended, but is not urgent.

Code_Saturne general features

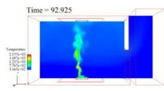
- Technology
  - Co-located finite volume, arbitrary unstructured meshes, predictor-corrector method
  - 350 000 lines of code, 35% Fortran, 45% C, 13% Python
- From version 2.0 on, different versions "x.y.z" are released
  - Production version every two years (increasing x)
    - With the release of a Verification & Validation summary report
  - Intermediate version every six months (increasing y)
    - With non-regression tests to ensure the code quality
  - Corrective versions when needed (increasing z)
    - To make sure the users are provided with the fixes in time

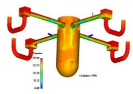
Code evolution

- Initially, turbulent Navier-Stokes kernel in Fortran 77, with a second adapter executable written in C for format conversions and I/O (and possible couplings.
- Progressively, add mesh handling structures in C in the kernel (not visible for the physics and numerics developer)
  - All handling of parallelism using MPI goes through a C-coded layer
  - Corresponding Fortran calls simplified as much as possible
  - More complex structures handled using C (to benefit from structures and dynamic memory allocation), and only a simple API is callable from Fortran.
- Fortran 90/95 was not a good option in 1997:
  - Compilers quite buggy based on feedback from other EDF software at the time
  - No free compiler (such as g77) for Fortran 90, a potential problem for external users
- Use of external libraries for some code aspects
  - Most often written in C

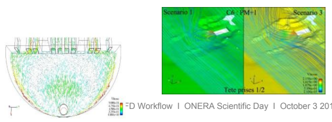
Applications

- Nuclear security and production optimization
  - PWR plant life time
  - Accident of dilution
  - Behavior of the assembly
  - Slagging and fouling in Pulverized Coal Furnace
  - Interaction fluid / structure in Pulverized Coal Furnace
  - Circulation and deposit of radionuclide into the reactor
  - Fire safety
  - Severe Accident on the Reactor Coolant System

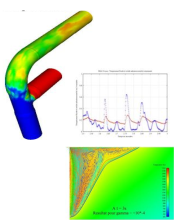




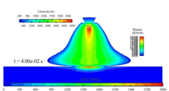
- Availability
  - cold source
  - Fouling of filtration organs by external bodies
  - (seaweed, sea currents, fingerlings, plant debris)
  - Availability of oil processing
  - Passage of particles and
  - cold shock in pumps

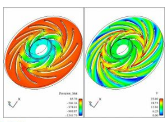


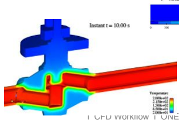
Understanding of phenomena



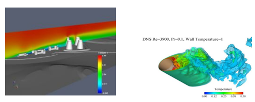
- Thermo hydraulics heading
  - Thermal fatigue in mixing area
  - Thermo hydraulics heading of valves
  - Calculus of hydraulic characteristics of safety valve
  - Thermal head for the primary pumps
- Various
  - Control the swirl flow
  - Prediction of tonal noise phenomena and cavitation in pipes
  - Multi-physics modeling of arc welding



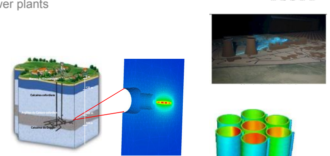


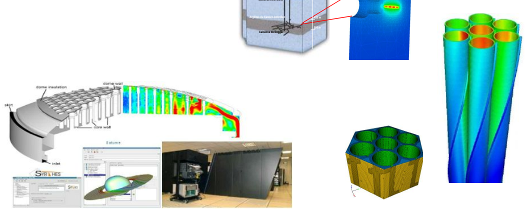


Other applications



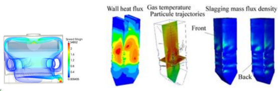
- Atmospheric modelization
  - Atmospheric dispersal around nuclear power plants
- Various
  - Dome temperature of EDF Energy's AGR
  - Prototype of generation IV reactors
  - Storage of nuclear waste

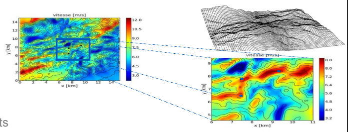


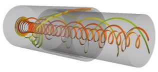


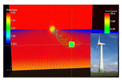
Other means of production

- Thermal production
  - Wood combustion modelization
  - Configurations of advanced low-NOx of the Q6(
  - Optimization of the combustion of the fuel/ coal
  - Biomass combustion
  - Oxy-combustion
  - Co-combustion (coal-biomass)
- New means of production
  - Wind potential estimates
  - Influence of the turbulence and the waves on the marine current turbine
  - Waste heat recovery in iron and steel production
  - Prototype of energy storage CAES
  - Towards Zero Emission Coal Power Plants



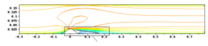


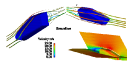


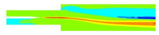


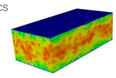
Quality assurance

- *Code_Saturne* widely used at EDF and outside
  - 150 EDF users (R&D and engineering)
  - 400 estimated users contacts outside EDF
  - *Code_Saturne* used for all general CFD calculation
  - *Code_Saturne* especially used in calculations for Nuclear Authorities
    - Code developed under Quality Assurance
- Verification
  - Before each long-term release is declared "fit for industrial use"
  - Around 30 test cases, covering all capabilities of the code
  - Academic test cases to industrial configurations
  - 1 to 15 calculations per test case
    - Around 1 man.year of work
- Validation
  - Further validation for specific industrial domains single phase thermal-hydraulics
- Auto-validation tool
  - Intended to automate regression testing
  - Decouple release cycle from validation cycle







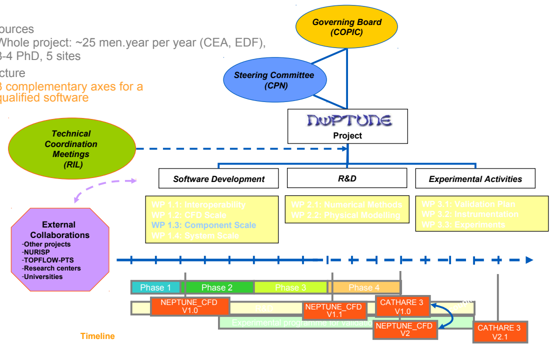


Code_Saturne version history

- Development
  - 1998: Prototype (long time EDF in-house experience, ESTET-ASTRID, N3S, ...)
  - 2000: version 1.0 (basic modelling, wide range of meshes)
  - 2001: Qualification for single phase nuclear thermal-hydraulic applications
  - 2004: Version 1.1 (complex physics, LES, parallel computing)
  - 2006: Version 1.2 (state of the art turbulence models, GUI)
  - 2008: Version 1.3 (massively parallel, ALE, code coupling, ...)
  - Released as open source (GPL licence)
  - 2008: Intermediate version 1.4 (parallel I/O, multigrid, atmospheric, cooling towers, ...)
  - 2010: Version 2.0 (massively parallel, code coupling, easy install & packaging, extended GUI)
  - 2011: Intermediate version 2.1 (parallel mesh joining and partitioning, improved scripts and setup)
  - 2012: Intermediate versions 2.2, 2.3 (many physical model additions, including GGDH thermal wall)
  - 2013: Version 3.0 (AFM, DFM thermal wall laws)

The NEPTUNE project

- Resources
  - Whole project: ~25 men.year per year (CEA, EDF),
  - 3-4 PhD, 5 sites
- Structure
  - 3 complementary axes for a qualified software



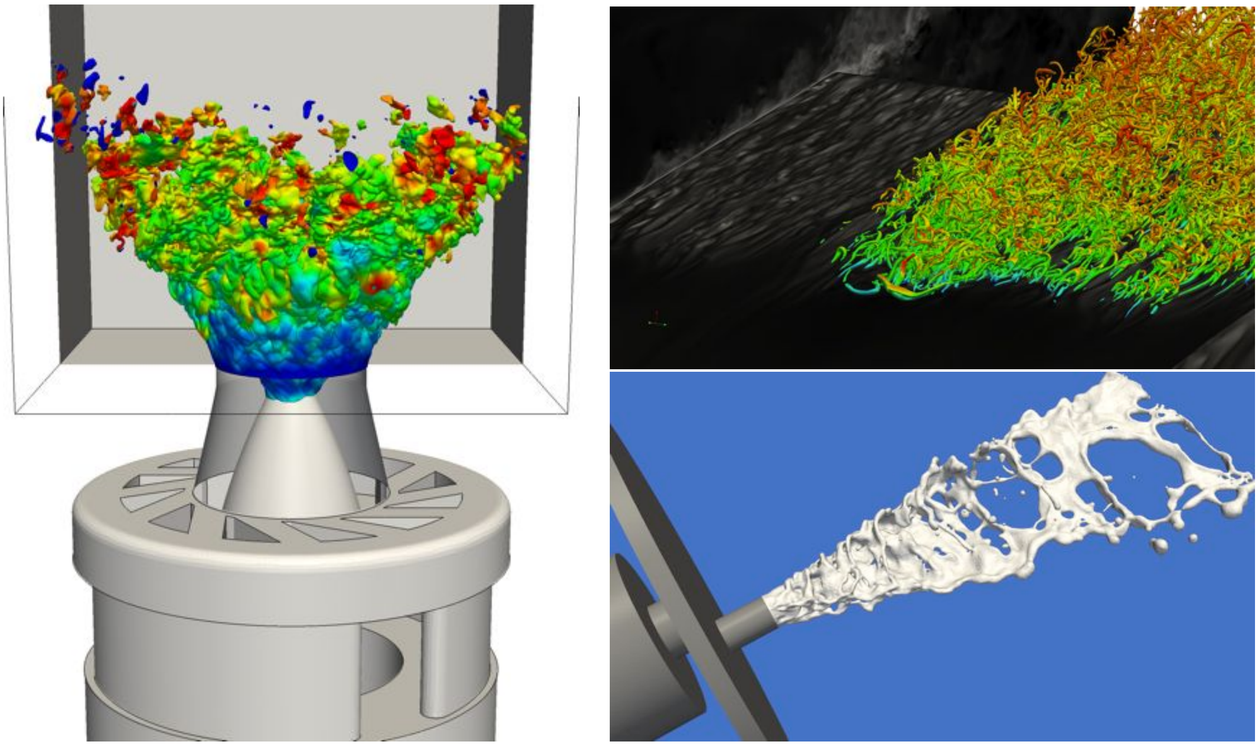
2.16 V. Moureau (Coria)

ONERA Scientific Day 2012

Strategies for the massively parallel solving of reacting and two-phase flows with billion-cell meshes. A few case studies with the YALES2 solver

Vincent MOUREAU
CORIA - CNRS UMR6614, INSA et Université de Rouen
http://www.coria-cfd.fr/index.php/User:Moureauv

The steady increase of computational resources in super-computing centers is a strong driving mechanism for Large-Eddy Simulation (LES) and Direct-Numerical Simulation (DNS) of turbulent flows. In these approaches, that are based on the solving of the 3D unsteady Navier-Stokes equations, the range of resolved scales and the CPU cost are directly related to the mesh resolution and the accuracy of the numerical schemes. However, the solving of the Navier-Stokes equations with several thousand CPUs and billion-cell meshes is challenging, especially when dealing with flows at low-Mach number. In this case, an elliptic linear system with several billion degrees of freedom needs to be inverted.



This presentation will focus on numerical strategies for the solving of turbulent reactive and two-phase flows on massively parallel machines. These strategies are implemented in a low-Mach number code named YALES2, which is collaboratively developed by CORIA and several other labs. In the recent years, this solver has been ported on various French and European platforms including the Curie machine at CEA in the framework of the 2nd call of the PRACE project. Thanks to its high-performance linear solvers and its fully distributed mesh management, YALES2 has been applied in a large range of turbulent flows with billion-cell meshes ranging from premixed turbulent flames in complex aeronautical burners to primary atomization of liquid fuel. During the summer, the code was used to perform highly-resolved LES of turbulent heat transfers on a turbine blade with tetrahedral-based meshes counting up to 143 billion elements.

YALES2

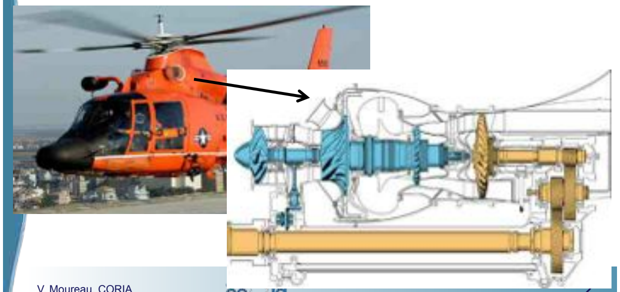
Strategies for the massively parallel solving of reacting and two-phase flows with billion-cell meshes.

A few case studies with the YALES2 solver

V. Moureau, G. Lartigue, P. Domingo,
L. Vervisch, G. Ribert, Y. D'Angelo
CNRS-CORIA, UMR 6614, Rouen
http://www.coria-cfd.fr

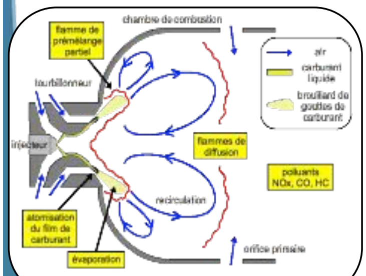
Why trying to model turbulent combustion ?

- 86% of the usable energy on earth is obtained through combustion
- Combustion occurs in many applications
  - Aeronautical engines, automotive industry, furnaces, …



The challenge

- Many phenomena at very different scales



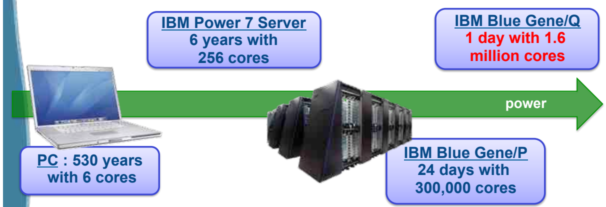
Aeronautical burner

Turbulence	10 mm 0.01mm
Atomization	0.1 mm
Spray transport	10 μm
Evaporation	10 μm
Combustion	0.1 mm

3 to 4 orders of magnitude

Estimate of resources required for future large-scale computations

- Simulation time for an aeronautical combustor with a mesh of 10 billion tets (2020 target)



- Many issues still have to be addressed
  - Mesh management, solving of large linear systems, post-processing, …

Outline

- Motivation
- The YALES2 code
  - Description
  - Mesh management
  - High performance linear solvers
  - Embedded post-processing
- A few case studies
- Conclusions

The YALES2 code

YALES2 www.coria.cfd.fr

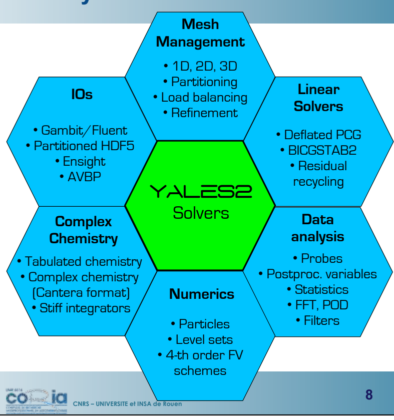
- YALES2 is an unstructured low-Mach number code for the DNS and LES of reacting two-phase flows in complex geometries. It solves the unsteady 3D Navier-Stokes equations
- It is used by more than 60 people in labs and in the industry
  - Labs: CORIA, I3M, LEGI, EM2C, IMFT, CERFACS, IFP-EN, ULB, …
  - Industry:

SAFRAN, Rhodia, GDF SUEZ, AIR LIQUIDE, AIRBUS, RENAULT, AREVA

- Awards
  - 3rd of the Bull-Joseph Fourier prize in 2009
  - 2011 IBM faculty award

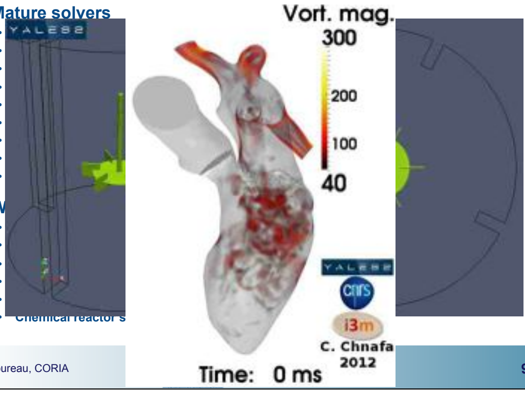
The YALES2 library

- 2 main maintainers
  - V. Moureau
  - G. Lartigue
- 250 000 lines of object-oriented f90
- Git and svn version management
- Portable on all the major platforms (even ARM proc.)



The YALES2 solvers

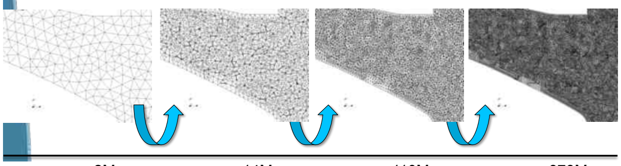
- Mature solvers
- W…



Mesh management

Strategy: refinement and partitioning

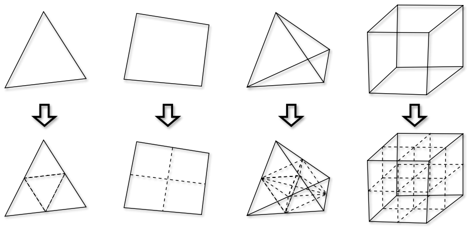
Automatic mesh refinement + partitioning



Size	2M	14M	110M	878M
Number of blocks	1	16	128	1024
Number of files	1	16	128	256

Mesh generation

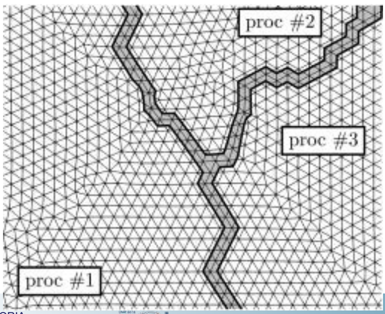
- Homogeneous mesh refinement allows to reach massive mesh sizes. The only constraint is that the geometry has to be well described by the first mesh.



- For tets, mesh refinement is not obvious (Rivara 1984)

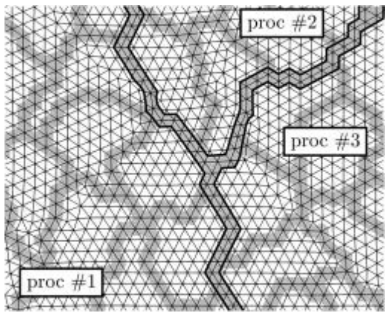
Mesh management on the processors

- 1st solution: single-level domain decomposition
- Several available libraries: Metis, Scotch, …



Mesh management on the processors

- 2nd solution: two-level domain decomposition (Moureau et al 2011)



High-performance linear solvers

Governing equations

- For DNS of iso-thermal flows at low-Mach number
- Velocity equation

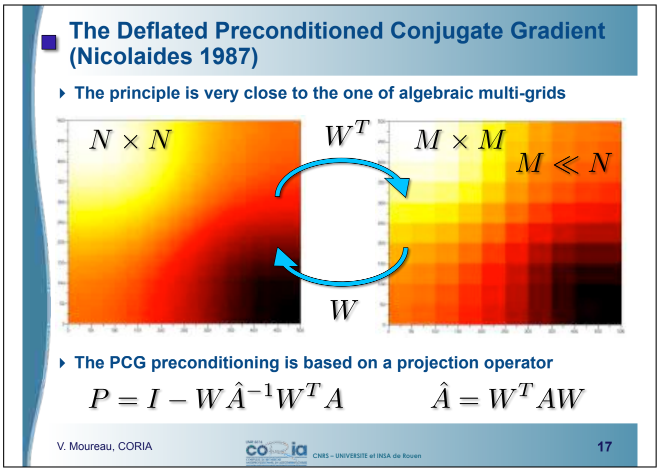
$$\frac{\partial \mathbf{u}}{\partial t} + \nabla \cdot (\mathbf{u}\mathbf{u}) = -\frac{1}{\rho}\nabla P + \frac{1}{\rho}\nabla \cdot \tau$$

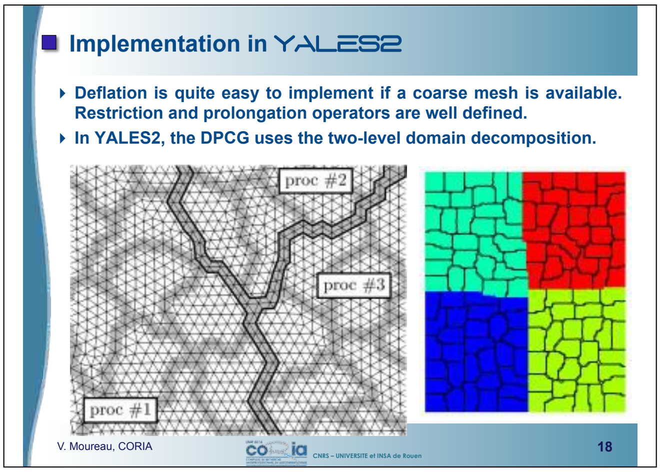
- Divergence-free constraint

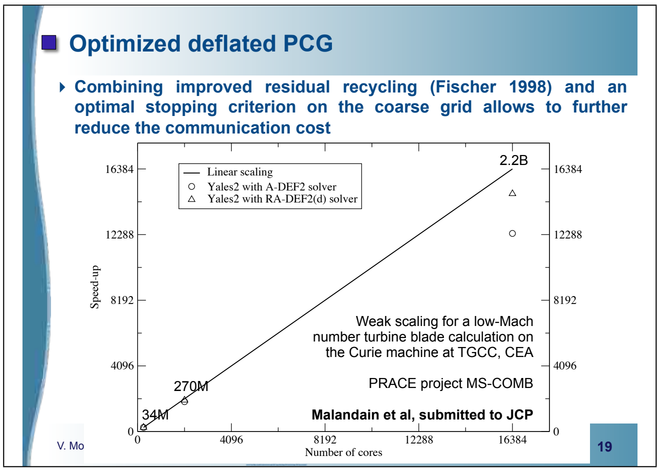
$$\nabla \cdot \mathbf{u} = 0$$

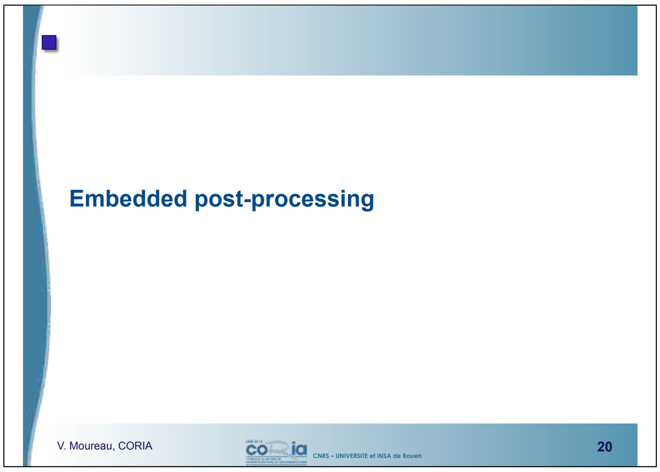
- Often solved with projection methods (Chorin 1968)

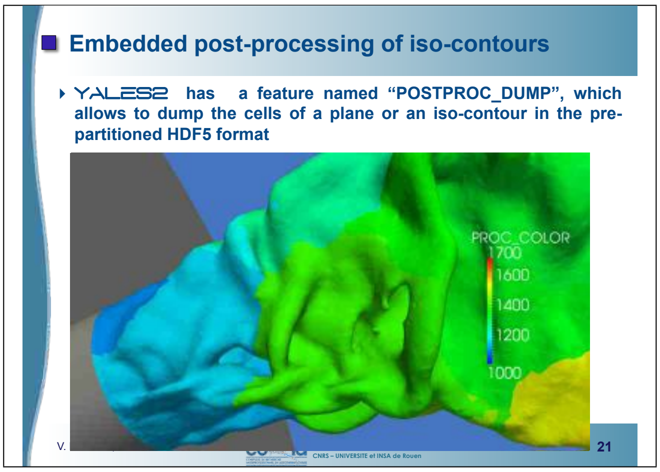
$$\nabla \cdot \left(\frac{1}{\rho}\nabla P\right) = \frac{\nabla \cdot \mathbf{u}^\star}{\Delta t} \iff Ax = b$$

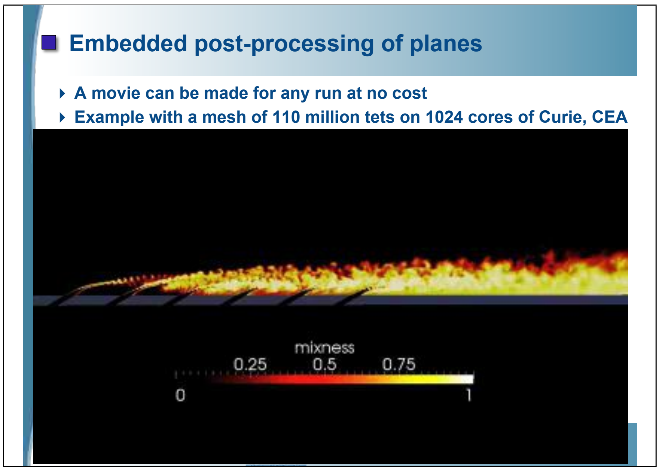

The Deflated Preconditioned Conjugate Gradient (Nicolaides 1987)
The principle is very close to the one of algebraic multi-grids
$N \times N$
W^T
$M \times M$
$M \ll N$
W
The PCG preconditioning is based on a projection operator
$P = I - W\hat{A}^{-1}W^T A$
$\hat{A} = W^T A W$
V. Moureau, CORIA
CNRS – UNIVERSITE et INSA de Rouen
17

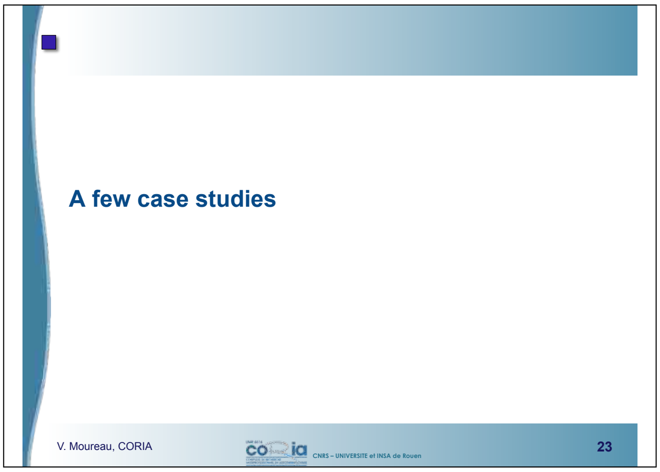

Implementation in YALES2
Deflation is quite easy to implement if a coarse mesh is available. Restriction and prolongation operators are well defined.
In YALES2, the DPCG uses the two-level domain decomposition.
proc #2
proc #3
proc #1
V. Moureau, CORIA
CNRS – UNIVERSITE et INSA de Rouen
18

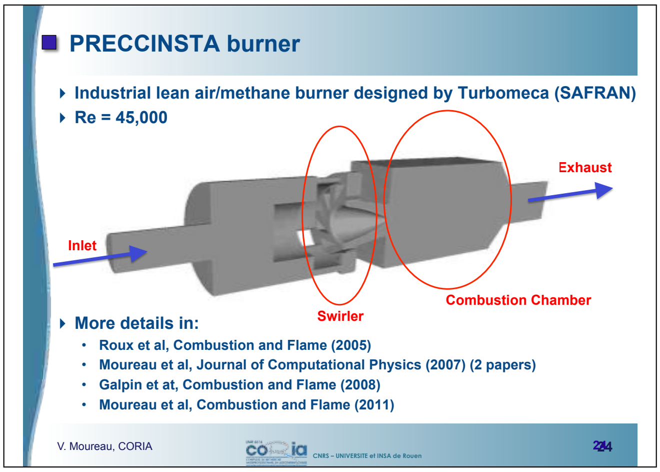

Optimized deflated PCG
Combining improved residual recycling (Fischer 1998) and an optimal stopping criterion on the coarse grid allows to further reduce the communication cost
Linear scaling
Yales2 with A-DEF2 solver
Yales2 with RA-DEF2(d) solver
2.2B
Speed-up
16384
12288
8192
4096
0
Weak scaling for a low-Mach number turbine blade calculation on the Curie machine at TGCC, CEA
PRACE project MS-COMB
Malandain et al, submitted to JCP
270M
34M
Number of cores
V. Mo
19

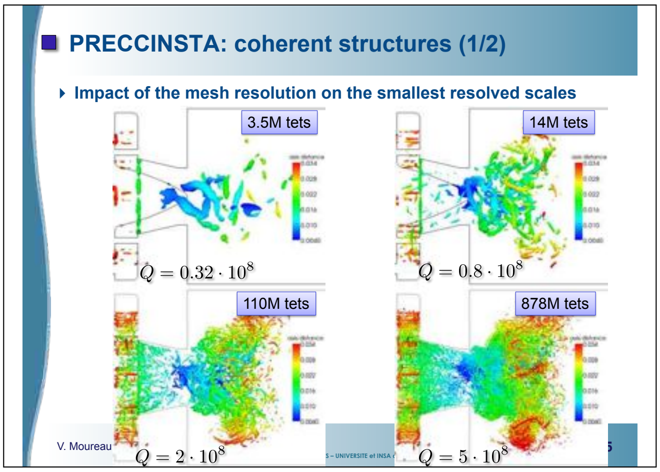

Embedded post-processing
V. Moureau, CORIA
CNRS – UNIVERSITE et INSA de Rouen
20

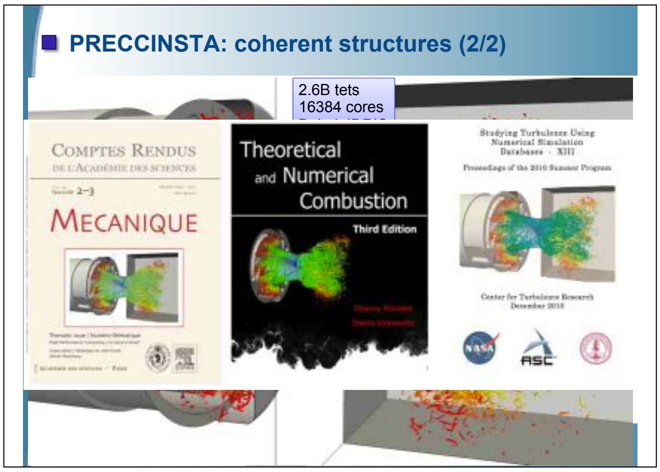

Embedded post-processing of iso-contours
YALES2 has a feature named "POSTPROC_DUMP", which allows to dump the cells of a plane or an iso-contour in the pre-partitioned HDF5 format
PROC_COLOR
1700
1600
1400
1200
1000
21

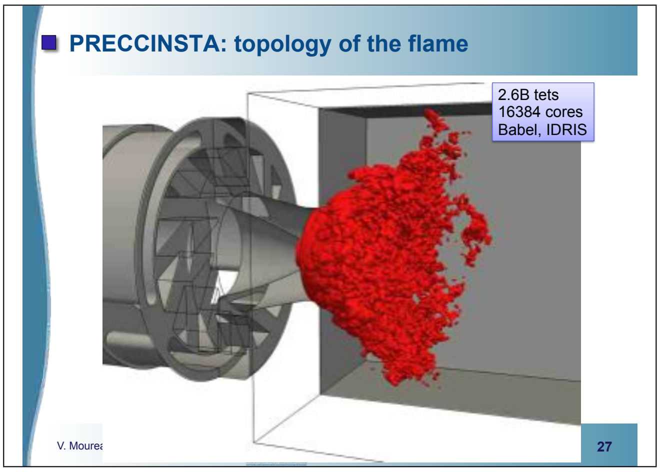

Embedded post-processing of planes
A movie can be made for any run at no cost
Example with a mesh of 110 million tets on 1024 cores of Curie, CEA
mixness
0.25
0.5
0.75
0
1

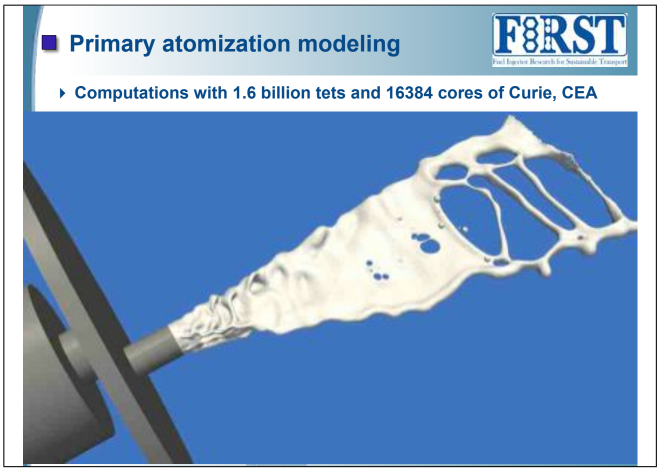

A few case studies
V. Moureau, CORIA
CNRS – UNIVERSITE et INSA de Rouen
23

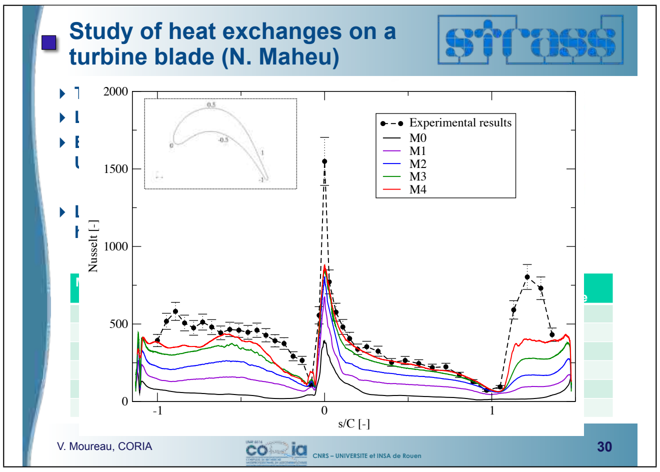

PRECCINSTA burner
Industrial lean air/methane burner designed by Turbomeca (SAFRAN)
Re = 45,000
Exhaust
Inlet
Swirler
Combustion Chamber
More details in:
Roux et al, Combustion and Flame (2005)
Moureau et al, Journal of Computational Physics (2007) (2 papers)
Galpin et at, Combustion and Flame (2008)
Moureau et al, Combustion and Flame (2011)
V. Moureau, CORIA
CNRS – UNIVERSITE et INSA de Rouen
24

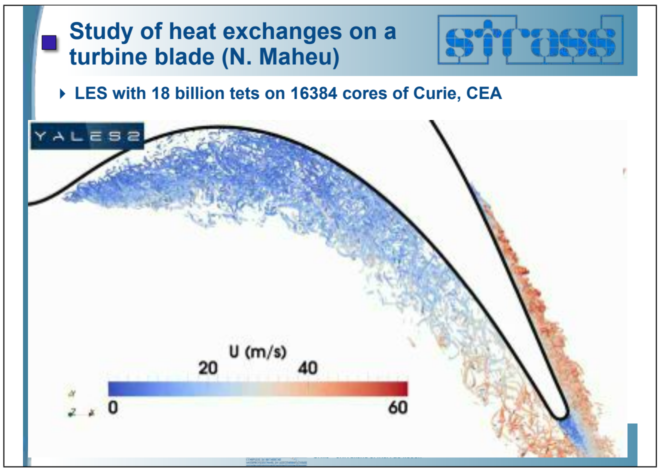

PRECCINSTA: coherent structures (1/2)
Impact of the mesh resolution on the smallest resolved scales
3.5M tets
14M tets
$Q = 0.32 \cdot 10^8$
$Q = 0.8 \cdot 10^8$
110M tets
878M tets
$Q = 2 \cdot 10^8$
$Q = 5 \cdot 10^8$
V. Moureau

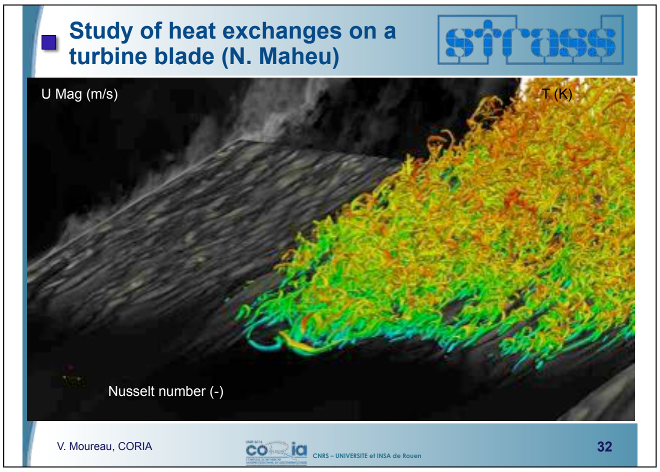

PRECCINSTA: coherent structures (2/2)
2.6B tets
16384 cores
Comptes Rendus de l'Académie des Sciences
Mécanique
Theoretical and Numerical Combustion
Third Edition
Studying Turbulence Using Numerical Simulation Databases - XIII
Proceedings of the 2010 Summer Program
Center for Turbulence Research
December 2010

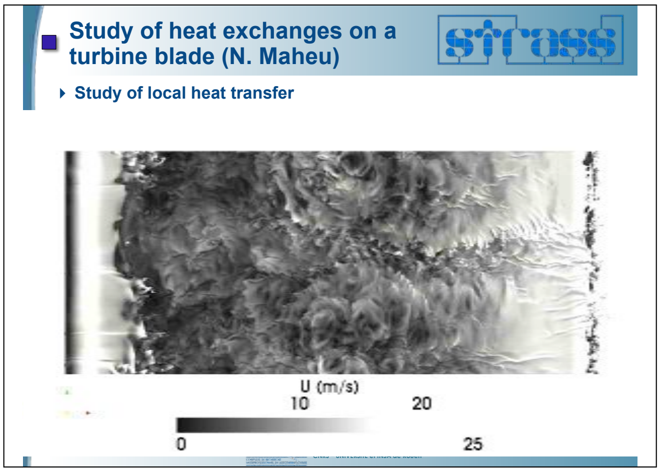

PRECCINSTA: topology of the flame
2.6B tets
16384 cores
Babel, IDRIS
27


Primary atomization modeling
FIRST
Computations with 1.6 billion tets and 16384 cores of Curie, CEA


Study of heat exchanges on a turbine blade (N. Maheu)
Experimental results
M0
M1
M2
M3
M4
Nusselt [-]
s/C [-]
V. Moureau, CORIA
30


Study of heat exchanges on a turbine blade (N. Maheu)
LES with 18 billion tets on 16384 cores of Curie, CEA
YALES2
U (m/s)

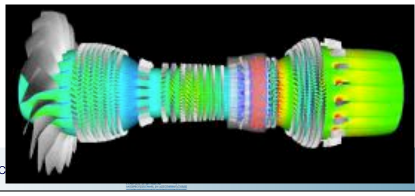

Study of heat exchanges on a turbine blade (N. Maheu)
U Mag (m/s)
T (K)
Nusselt number (-)
V. Moureau, CORIA
32


Study of heat exchanges on a turbine blade (N. Maheu)
Study of local heat transfer
U (m/s)

Conclusions & perspectives

Conclusions & Perspectives

- Billion-cell calculations are feasible on the current machines
- Their pre- and post-processing are still difficult
- Some remaining challenges and some potential solutions
  - Large-scale feature extraction: high-order implicit filters
  - Mesh generation: local mesh refinement, mesh skewness smoothing
  - Efficiency of multi-physics simulations: dynamic load balancing
- A 2020 target ?



References & Acknowledgements

- References

  - Malandain, M., Maheu, N., and Moureau, V., "Optimization of the deflated Conjugate Gradient algorithm for the solving of elliptic equations on massively parallel machines", submitted to J. Comp. Physics
  - Maheu, N., Moureau, V., Domingo, P., Duchaine, F. & Balarac, G., « Large-eddy simulations of flow and heat transfer around a low-mach turbine blade ». CTR Summer Program. Center for Turbulence Research, NASA Ames/Stanford Univ, 2012.
  - Moureau, V., Domingo, P., and Vervisch, L., "From Large-Eddy Simulation to Direct Numerical Simulation of a lean premixed swirl flame: Filtered Laminar Flame-PDF modelling", Comb. and Flame, 2011, 158, 1340–1357
  - Moureau, V., Domingo, P., and Vervisch, L., "Design of a massively parallel CFD code for complex geometries", Comptes Rendus Mécanique, 2011, 339 (2-3), 141-148

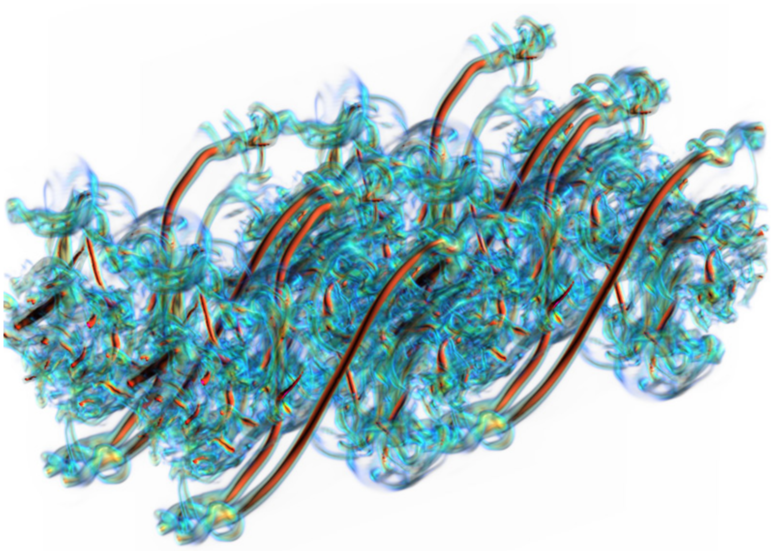
- Acknowledgements
  - D. Taieb, G. Lartigue, P. Domingo, L. Vervisch, Y. D'Angelo, O. Desjardins
  - PhD students: N. Maheu, F. Pecquery, M. Malandain, …
  - Computing centers: IDRIS, TGCC, CINES, CRIHAN, JULICH

2.17 Y.M. Lefebvre (Intelligent Light)

CFD Workflow Improvements for Today and Tomorrow

Yves-Marie Lefebvre[1], Earl P. N. Duque[2], Matthew N. Godo[3] and Steve M. Legensky[4]

Intelligent Light, Rutherford, New Jersey, 07070, USA



Volume rendering of vorticity magnitude for a temporal mixing layer.

This presentation describes various efforts conducted in recent years by Intelligent Light in the field of CFD workflows. These projects took two forms:

- Workflow improvement and data management capabilities in our commercial post-processor **FieldView**. We will describe in this section recent successes achieved by Intelligent Light's Services Department in implementing and combining these methods for high-end industrial users, which led to significant improvements in average CFD results processing time and overall throughput, with the same hardware infrastructure.

- Research projects, conducted by Intelligent Light's Applied Research Group (ARG), together with some of the most advanced CFD research teams in the world. This part will focus on the "Intelligent In-Situ Feature Detection, Tracking and Visualization for Turbulent Flow Simulations" (IFDT) project [1] resulting in a new prototype visualization and CFD data analysis software system for flow feature data tracking and extraction. This prototype system can explore, detect,

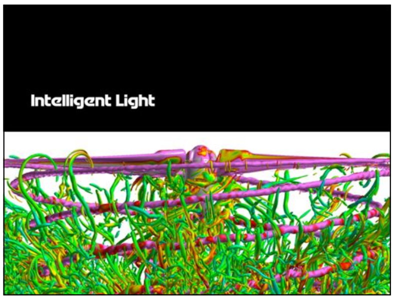
[1] Sales and Support Engineer, Intelligent Light, yml@ilight.com
[2] Manager, Applied Research Group, Intelligent Light
[3] Product Manager, Intelligent Light
[4] General Manager and Founder, Intelligent Light

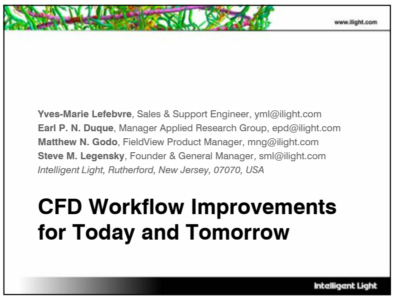
track and analyze flow features predicted by large scale unsteady CFD simulations. The feature extractor method executes In-Situ with a flow solver via a Python Interface Framework, to avoid the overhead of saving data to file. The Volume Rendering capability (see image above) was developed in a prototype version of **FieldView,** and will soon be available in an upcoming production release.

References:

[1] Earl P.N. Duque, Daniel Hiepler, Christopher P. Stone and Steve M Legensky, Kwan-Liu Ma, Christopher Muelder and Jishang Wei, “IFDT – Intelligent In-Situ Feature Detection, Extraction, Tracking and Visualization for Turbulent Flow Simulations”, Paper Number ICCFD7-1703, Seventh International Conference on Computational Fluid Dynamics (ICCFD7), Big Island, Hawaii, July 9-13, 2012.

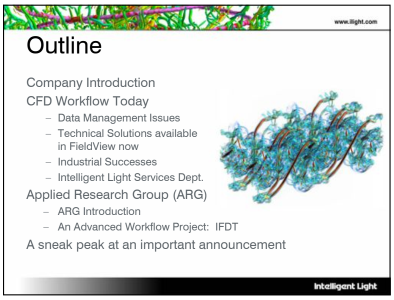
ONERA Scientific Days - CFD Workflow

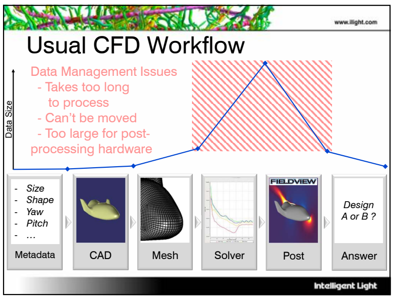




ONERA Scientific Days - CFD Workflow

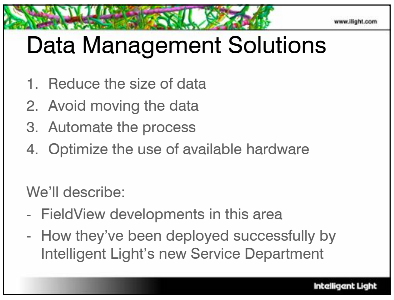
© 2012 Intelligent Light Proprietary

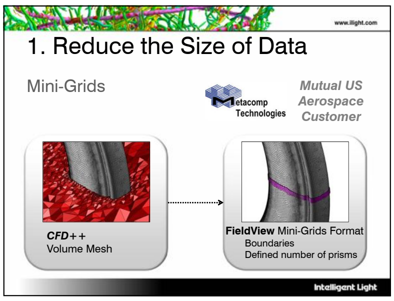




ONERA Scientific Days - CFD Workflow

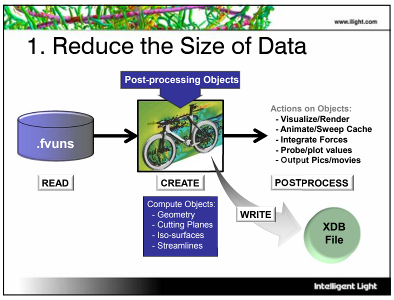
© 2012 Intelligent Light Proprietary

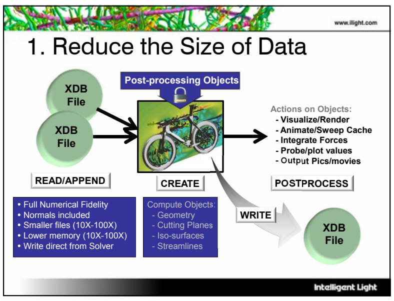




ONERA Scientific Days - CFD Workflow

© 2012 Intelligent Light Proprietary

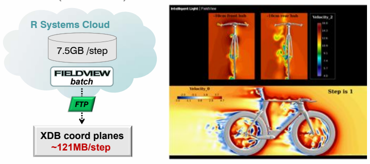




ONERA Scientific Days - CFD Workflow

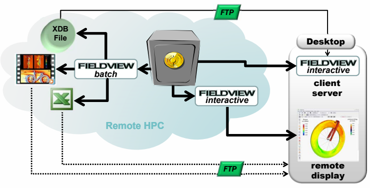
1. Reduce the Size of Data

XDB (Extracts)



Matthew N Godo, *Aerodynamic Performance of Racing Bicycles & Practical Considerations for CFD Workflows*, STAR Global Conference 2012

2. Avoid Moving the Data



ONERA Scientific Days - CFD Workflow

3. Automate the Process

FVX / python Automation

- 80% of the design is driven by CFD
- 90% of post-processing is automated via batch

"We can automatically generate full car reports, videos and graphics and share those around the company. Everyone … then has the knowledge needed to make better decisions."
Nathan Sykes, CFD Team Leader

4. Optimize Use of Hardware

Typical Post-Processing Node

- **8 to 12 cores**
- 32 to 384 GB of RAM
- High-end GPU (ex: nVIDIA Quadro 4000 to 6000 or AMD FirePro W5000 to 9000)

CFD has greatly benefited from Parallelization

- Solvers often show very good scalability
- What about Parallel Post-Processing?

ONERA Scientific Days - CFD Workflow

A Word On Parallel Post-Processing

Parallel Post-Processing is a requirement

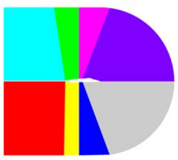
- For reading multi-file partitioned data
- For reading very large datasets
  - May exceed the capacity of a single node

Is it sufficient for reaching high performance?

1. Read data	2. Create Post-Processing Objects	3. Render Objects
Good Scalability	- Boundaries - Cuts - Iso-surfaces - Streamlines ?	New Graphic Engine

A Word On Parallel Post-Processing

Scalability of post-processing objects creation



Example

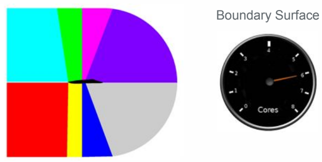
- Typical external aero mesh
- Metis partitioning algorithm
- 8 partitions

ONERA Scientific Days - CFD Workflow

A Word On Parallel Post-Processing

Scalability of post-processing objects creation

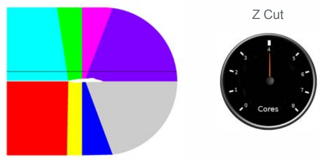
Boundary Surface



A Word On Parallel Post-Processing

Scalability of post-processing objects creation

Z Cut

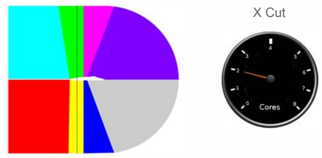


ONERA Scientific Days - CFD Workflow

www.ilight.com

A Word On Parallel Post-Processing

Scalability of post-processing objects creation

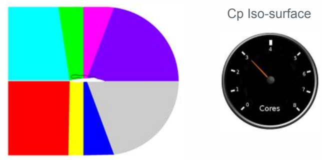


Intelligent Light

www.ilight.com

A Word On Parallel Post-Processing

Scalability of post-processing objects creation



Intelligent Light

9

ONERA Scientific Days - CFD Workflow

www.ilight.com

IL Services – Design a new Workflow

Keys to Increased Performance
- Use more cores
  - Parallel post-processing
  - Concurrent Processes
- Lower memory usage (Lets you use even more cores)
  - XDBs
- Increase graphic card usage
  - Hardware rendered batch

Solution
Concurrent Hardware Rendered XDB Workflow

Intelligent Light

www.ilight.com

Intelligent Light Services Dept.

- Today's CFD environment is highly complex
- There is no such thing as a Workflow that will work for every organization
  Different **constraints** (Hardware, network, data size…)
  Different **goals**
- Intelligent Light Services Department has been created to help our customers with their CFD workflow

Intelligent Light

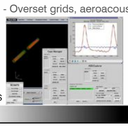
10

ONERA Scientific Days - CFD Workflow

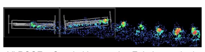
www.ilight.com

Applied Research Group (ARG)

- Manager: Earl P. N. Duque (epd@ilight.com)
- Full time + Distributed Team + Partners
  University Collaborators: MIT, Penn State, Georgia Tech, University of California
- Industrial and Government Projects
  - NASA, Air Force, Dept. of Energy
  - Overset grids, aeroacoustics, ultra-scale data management…



RCAAPS



VLRCOE – Coupled Lagrangian Eulerian method

Intelligent Light

www.ilight.com

Applied Research Group (ARG)

Bring the most advanced technologies into the FieldView commercial product

Former ARG projects now in FieldView
XDB
MPI based Parallel

python ARG actively using NumPy, SciPy, PyQt, Matplotlib, Python Frameworks.

Intelligent Light

11

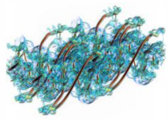
ONERA Scientific Days - CFD Workflow

www.ilight.com

The IFDT Project

IFDT – Intelligent In-Situ Feature Detection, Extraction, Tracking and Visualization for Flow Simulations

1. In-situ processing
2. Volume Rendering
   - Detection and tracking of flow features
   - Statistics
3. python based user interface



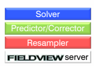
Reference:
- Earl P.N. Duque, Daniel Hiepler, Christopher P. Stone and Steve M Legensky, Kwan-Liu Ma, Christopher Muelder and Jishang Wei, "IFDT – Intelligent In-Situ Feature Detection, Extraction, Tracking and Visualization for Turbulent Flow Simulations", Paper Number ICCFD7-1703, Seventh International Conference on Computational Fluid Dynamics (ICCFD7), Big Island, Hawaii, July 9-13, 2012.
- To be presented at SuperComputing 2012, Visualization Showcase, November 2012

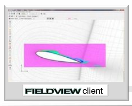
Intelligent Light

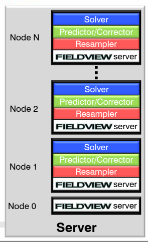
www.ilight.com

In-Situ Processing

Python Software Interface Framework
- Each program has been converted into a shared python library
- Executed under a python high level control







Intelligent Light

12

ONERA Scientific Days - CFD Workflow

In-Situ Processing Benefits

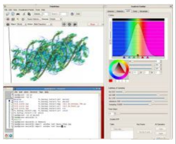
Programs share the same address space

- No File I/O
- Minimum memory footprint on the server
  - + 7% for LESLIE3D
  - + 14% for OVERFLOW-2

Volume Rendering

Volume Rendering used with various Transfer Functions for feature discrimination

- Four types of transfer function shapes
  Gaussian, Step-Function, "S-Curve", Delta Function
- The Transfer Function UI
  - Enables Interactive highlighting of flow
  - Speeds evaluation of flow features
  - Interactive Feature Discrimination



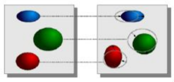
13

ONERA Scientific Days - CFD Workflow

Detection and Tracking

Predictor-Corrector Feature Extraction and Tracking

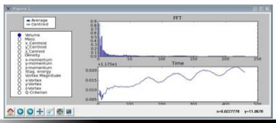
- Based upon the prediction-correction method by Muelder and Ma
- A prediction step to make the best guess of the feature region in the subsequent time step
- Followed by growing and shrinking the border of the predicted region to coherently extract the actual feature of interest



Reference:
C. Muelder and K.-L. Ma, "Interactive feature extraction and tracking by utilizing region coherency," in *Proceedings of 2009 IEEE Pacific Visualization Symposium*, 2009.

Statistics Viewer

- Transient information of a tracked feature is collected by the server processes and sent to the client for display
- Time history of all the voxels within a given tracked feature.
  Average value or Centroid of the volume of voxels

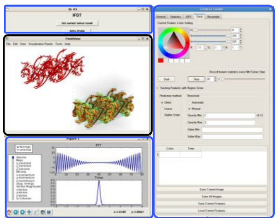


14

ONERA Scientific Days - CFD Workflow

Python Based User Interface

FieldView Prototype



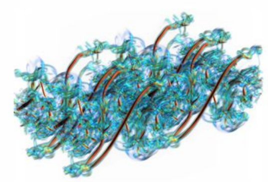
python based interface

Conclusion

- Post-processing has became strategic in high-end CFD workflows
  Too much data to process
- Many data management solutions have been added to FieldView over the years
  - Best solution is a different combination for every workflow
  - Services Dept. created to help customers progress through this maze
- More capabilities will be added in the future, thanks to the advanced technologies developed by ARG

15

ONERA Scientific Days - CFD Workflow



16

2.18 P. Sadlo (Univ. Stuttgart)

Advanced Techniques in Computational Flow Visualization

Filip Sadlo

Visualization Research Center of the University of Stuttgart

|Germany

Abstract

Today's CFD workflows typically focus on the meshing and solving stages. As a consequence, the visualization stage cannot catch up with the complexity and variety of the simulation results. On the one hand this slows down research and development in the application domains and the CFD workflow itself, on the other hand it can prevent important discoveries and insights.

Over the last two decades, research in computational visualization has emerged into a competitive discipline at the interface between scientific computing and computer graphics. Its ultimate goal "seeing the unseen" has been pursued in many application domains and in particular in CFD where straightforward depiction is often insufficient. We will exemplify the potential of advanced flow visualization techniques to analyze vortical flow, to reveal the structure of transport in time-dependent vector fields, and to accurately visualize CFD results given in higher-order representation. However, while computational flow visualization is a success story in research, the application domain stays behind. A possible way to a more holistic CFD workflow could be through commercial simulation codes: most of them feature a post-processing stage and it could be in the interest of these companies and their customers to ease the overall CFD procedure by including more advanced flow visualization techniques.

Advanced Techniques in Computational Flow Visualization

Onera Scientific Day 2012 – CFD Workflow – 2012-10-03

Filip Sadlo – Visualization Research Center, University of Stuttgart, Germany

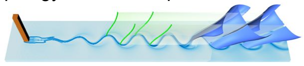
VISUS

VISUALIZATION RESEARCH CENTER, UNIVERSITY OF STUTTGART

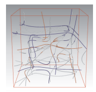
Overview

- Vortical Flow

- Topology for Time-Dependent Flow



- Visualization of Higher-Order Data



2 VISUS

VISUALIZATION RESEARCH CENTER, UNIVERSITY OF STUTTGART

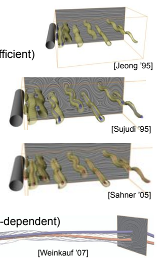
Vortex Visualization

Vortex criteria (Eulerian, local)

- Vorticity: $\omega = \nabla \times \mathbf{u}$, $|\omega| > 0$ (Galilean-invariant (GI), insufficient)
- Helicity: $h = \omega \cdot \mathbf{u}$, $|h| > 0$ (not so bad, but not GI)
- $\lambda_2 < 0$ [Jeong '95] (good, GI)
  (λ_2: medium eigenvalue of ($S^2 + \Omega^2$))
  $S = (\nabla u + \nabla u^T)/2$, $\Omega = (\nabla u - \nabla u^T)/2$

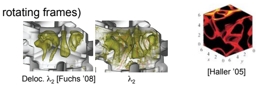
Vortex core lines

- Sujudi-Haimes: $\mathbf{u} \parallel \varepsilon_r$ [Sujudi '95] (not GI)
  (ε_r: real eigenvector of $\nabla\mathbf{u}$, others have to be complex)
- Valley lines of λ_2 [Sahner '05] (GI)
- Space-time Sujudi-Haimes [Weinkauf '07] (GI, time-dependent)



Vortex criteria (Lagrangian)

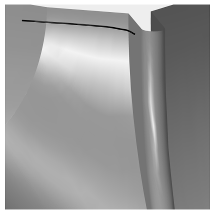
- Mz [Haller '05] (GI, even invariant to rotating frames)
- Delocalized criteria [Fuchs '08]
  (Lagrangian averaging of any quantity)

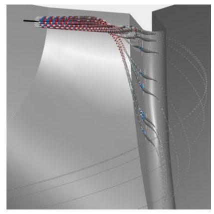


3 VISUS

VISUALIZATION RESEARCH CENTER, UNIVERSITY OF STUTTGART

Vorticity Transport Analysis: Motivation





- Vortex core lines
  - Local feature – local analysis
- Vorticity transport on path lines
  - Visualize vortex generation

4 Sadlo and Peikert, Visualization Tools for Vorticity Transport Analysis in Incompressible Flow, IEEE Trans. on Vis. and Comp. Graph., vol. 12, no. 5, pp. 949 – 956, 2006. VISUS

VISUALIZATION RESEARCH CENTER, UNIVERSITY OF STUTTGART

Vorticity Transport Analysis

Incompressible Navier-Stokes equation

$$\frac{D\mathbf{u}}{Dt} = \frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} = -\frac{\nabla p}{\rho} + \nu \nabla^2 \mathbf{u}$$

p: pressure
ρ: uniform density
ν: uniform (kinematic) viscosity

Vorticity equation

$$\frac{D\omega}{Dt} = \frac{\partial \omega}{\partial t} + \mathbf{u} \cdot \nabla \omega = \omega \cdot \nabla \mathbf{u} + \nu \nabla^2 \omega$$

5 VISUS

VISUALIZATION RESEARCH CENTER, UNIVERSITY OF STUTTGART

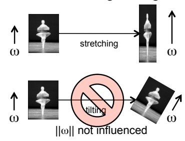
Vorticity Transport Analysis

Vorticity equation

$$\frac{D\omega}{Dt} = \frac{\partial \omega}{\partial t} + \mathbf{u} \cdot \nabla \omega = \underbrace{\omega \cdot \nabla \mathbf{u}}_{\text{stretching / tilting}} + \underbrace{\nu \nabla^2 \omega}_{\text{diffusion}}$$

$(\omega \cdot \nabla \mathbf{u})_{\parallel\omega}$ $(\omega \cdot \nabla \mathbf{u})_{\perp\omega}$

Vortex stretching / tilting:



Goal: quantitative analysis
→ Restrict analysis to ||ω|| by projection on ω

$$\left(\frac{\partial \omega}{\partial t} + \mathbf{u} \cdot \nabla \omega\right)_{\parallel\omega} = (\omega \cdot \nabla \mathbf{u})_{\parallel\omega} + (\nu \nabla^2 \omega)_{\parallel\omega} = \sigma + \delta$$

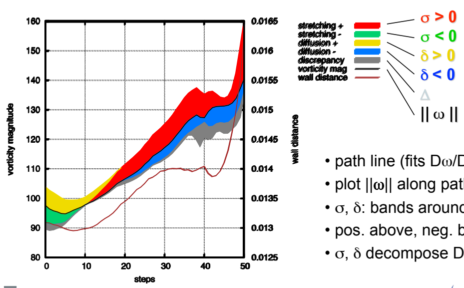
Visualize σ, δ, Δ

$\Delta = |\sigma + \delta - \text{LHS}|$ (discrepancy due to numerics)

6 VISUS

VISUALIZATION RESEARCH CENTER, UNIVERSITY OF STUTTGART

Vorticity Transport Analysis: 2D Plot

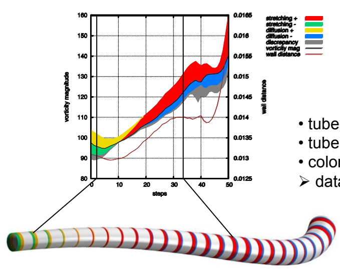


- path line (fits Dω/Dt)
- plot ||ω|| along pathline
- σ, δ: bands around ||ω||
- pos. above, neg. below
- σ, δ decompose D||ω||/Dt

7 VISUS

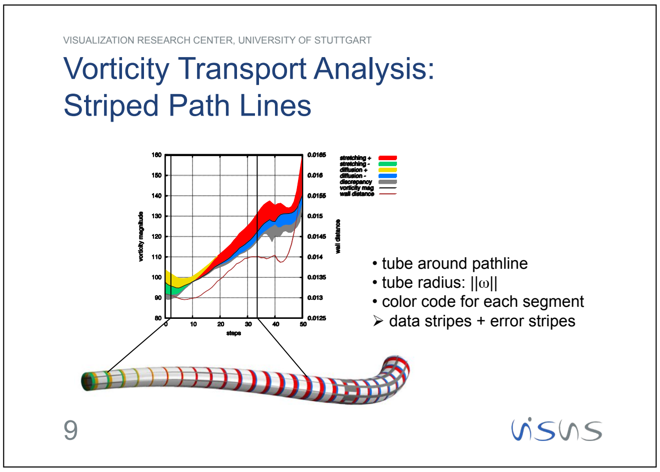
VISUALIZATION RESEARCH CENTER, UNIVERSITY OF STUTTGART

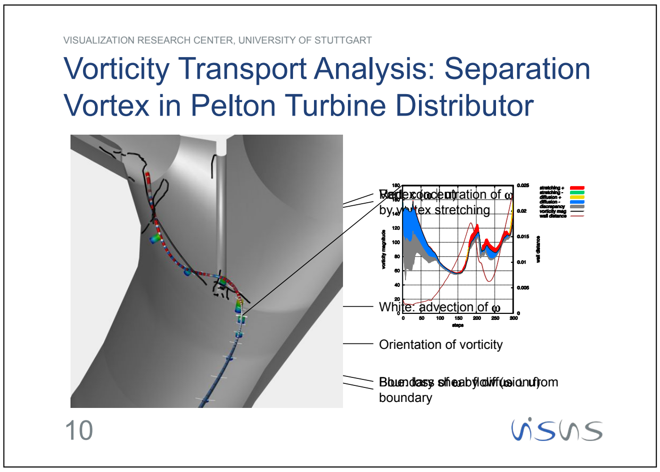
Vorticity Transport Analysis: Striped Path Lines

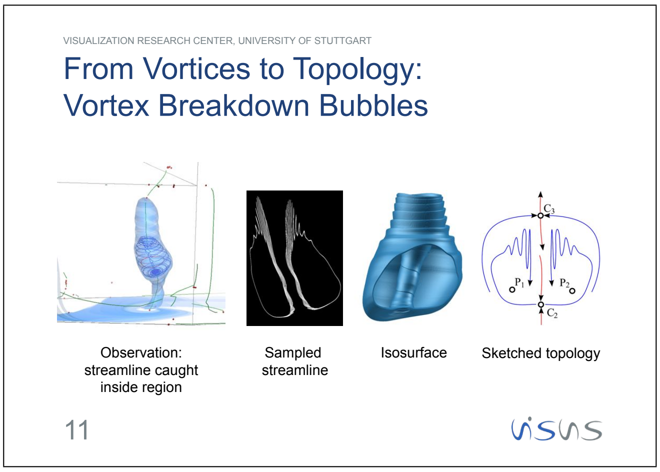


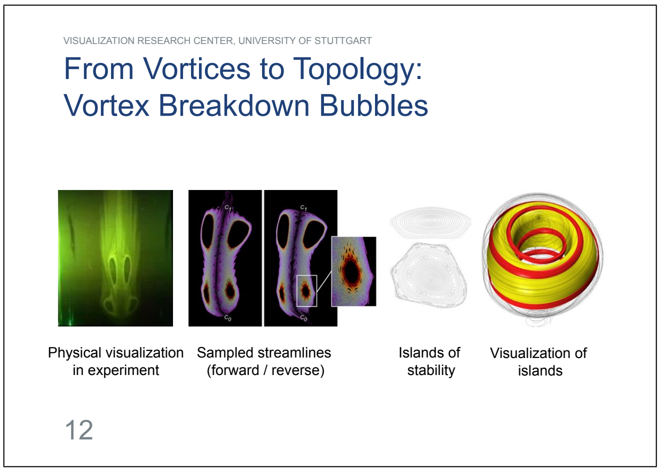
- tube around pathline
- tube radius: ||ω||
- color code for each segment
  ➢ data stripes

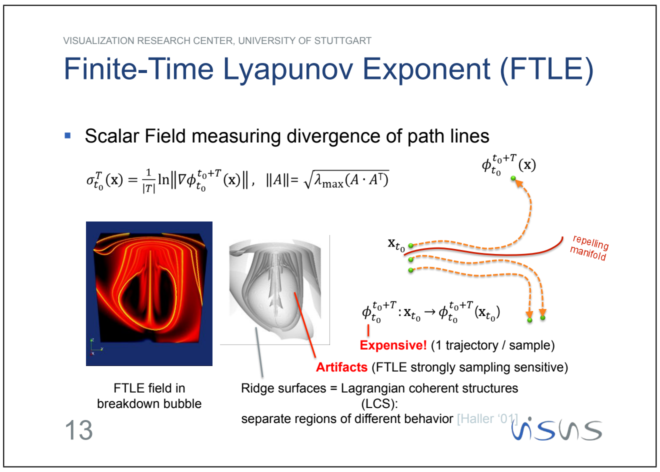
8 VISUS

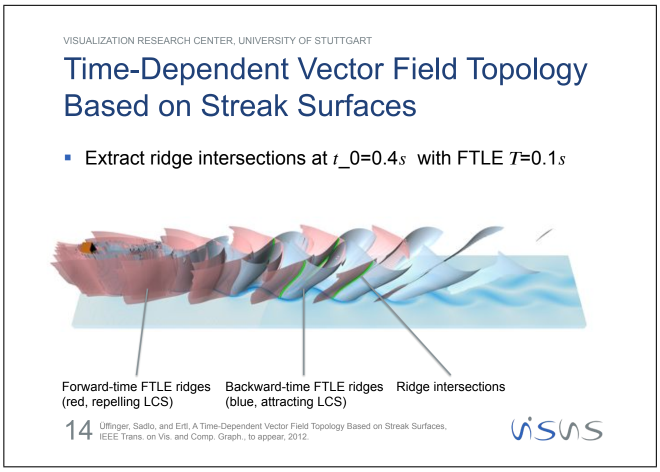

VISUALIZATION RESEARCH CENTER, UNIVERSITY OF STUTTGART
Vorticity Transport Analysis: Striped Path Lines
• tube around pathline
• tube radius: ||ω||
• color code for each segment
➢ data stripes + error stripes
9

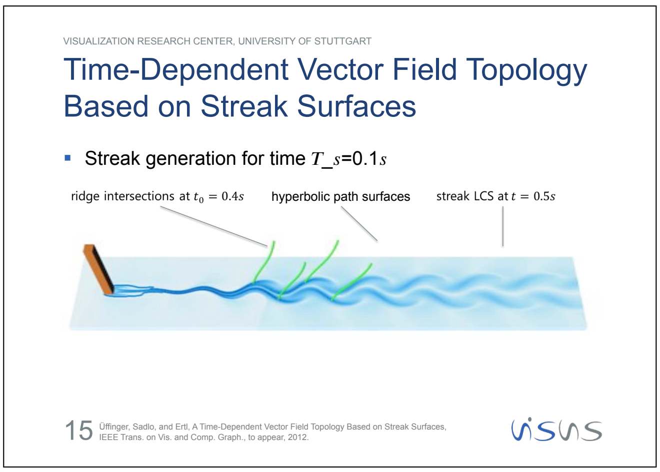

VISUALIZATION RESEARCH CENTER, UNIVERSITY OF STUTTGART
Vorticity Transport Analysis: Separation Vortex in Pelton Turbine Distributor
Red: concentration of ω by vortex stretching
White: advection of ω
Orientation of vorticity
Blue: loss of ω by diffusion from boundary
10

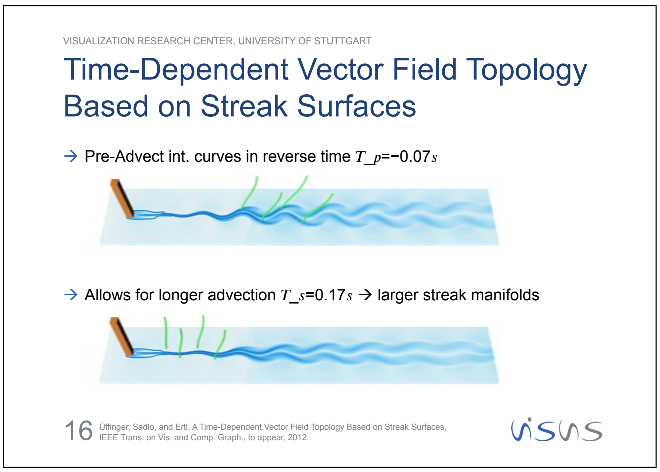

VISUALIZATION RESEARCH CENTER, UNIVERSITY OF STUTTGART
From Vortices to Topology: Vortex Breakdown Bubbles
Observation: streamline caught inside region
Sampled streamline
Isosurface
Sketched topology
11


VISUALIZATION RESEARCH CENTER, UNIVERSITY OF STUTTGART
From Vortices to Topology: Vortex Breakdown Bubbles
Physical visualization in experiment
Sampled streamlines (forward / reverse)
Islands of stability
Visualization of islands
12


VISUALIZATION RESEARCH CENTER, UNIVERSITY OF STUTTGART
Finite-Time Lyapunov Exponent (FTLE)
▪ Scalar Field measuring divergence of path lines
$\sigma_{t_0}^{T}(\mathrm{x}) = \frac{1}{|T|}\ln\left\|\nabla\phi_{t_0}^{t_0+T}(\mathrm{x})\right\|, \quad \|A\| = \sqrt{\lambda_{\max}(A \cdot A^{T})}$
$\phi_{t_0}^{t_0+T}(\mathrm{x})$
repelling manifold
$\phi_{t_0}^{t_0+T} : \mathrm{x}_{t_0} \to \phi_{t_0}^{t_0+T}(\mathrm{x}_{t_0})$
Expensive! (1 trajectory / sample)
Artifacts (FTLE strongly sampling sensitive)
FTLE field in breakdown bubble
Ridge surfaces = Lagrangian coherent structures (LCS): separate regions of different behavior [Haller '01]
13


VISUALIZATION RESEARCH CENTER, UNIVERSITY OF STUTTGART
Time-Dependent Vector Field Topology Based on Streak Surfaces
▪ Extract ridge intersections at t_0=0.4s with FTLE T=0.1s
Forward-time FTLE ridges (red, repelling LCS)
Backward-time FTLE ridges (blue, attracting LCS)
Ridge intersections
14 Üffinger, Sadlo, and Ertl, A Time-Dependent Vector Field Topology Based on Streak Surfaces, IEEE Trans. on Vis. and Comp. Graph., to appear, 2012.

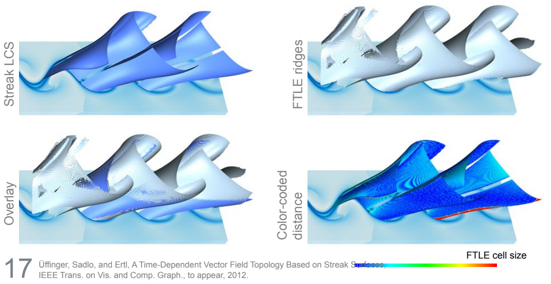

VISUALIZATION RESEARCH CENTER, UNIVERSITY OF STUTTGART
Time-Dependent Vector Field Topology Based on Streak Surfaces
▪ Streak generation for time T_s=0.1s
ridge intersections at $t_0 = 0.4s$
hyperbolic path surfaces
streak LCS at $t = 0.5s$
15 Üffinger, Sadlo, and Ertl, A Time-Dependent Vector Field Topology Based on Streak Surfaces, IEEE Trans. on Vis. and Comp. Graph., to appear, 2012.


VISUALIZATION RESEARCH CENTER, UNIVERSITY OF STUTTGART
Time-Dependent Vector Field Topology Based on Streak Surfaces
→ Pre-Advect int. curves in reverse time T_p=−0.07s
→ Allows for longer advection T_s=0.17s → larger streak manifolds
16 Üffinger, Sadlo, and Ertl, A Time-Dependent Vector Field Topology Based on Streak Surfaces, IEEE Trans. on Vis. and Comp. Graph., to appear, 2012.

VISUALIZATION RESEARCH CENTER, UNIVERSITY OF STUTTGART

Time-Dependent Vector Field Topology Based on Streak Surfaces

- Reduced Aliasing (and faster: $O(N^2)$ instead $O(N^4)$ for animations)

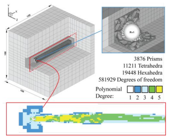


17 Üffinger, Sadlo, and Ertl, A Time-Dependent Vector Field Topology Based on Streak Surfaces, IEEE Trans. on Vis. and Comp. Graph., to appear, 2012.

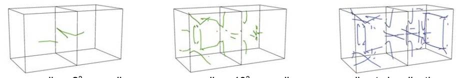
VISUALIZATION RESEARCH CENTER, UNIVERSITY OF STUTTGART

Visualization of Higher-Order CFD Data

- Discontinuous Galerkin (DG) Simulation
  - Higher-order (polynomial) representation
  - → Rich structure per cell
- Most visualization techniques based on trilinear interpolation
  - → Common approach: resampling
  - → Overhead in memory and computation
  - → Artifacts because fine detail hard to resolve



→ Direct visualization of ridges and vortex core lines in DG data



18 C. Pagot, D. Osmari, F. Sadlo, D.Weiskopf, T. Ertl, J. Comba: Efficient Parallel Vectors Feature Extraction from Higher-Order Data, Computer Graphics Forum, 30(3), pp. 751–760, 2011.

VISUALIZATION RESEARCH CENTER, UNIVERSITY OF STUTTGART

Visualization of Higher-Order CFD Data

- Based on reduced affine arithmetic (RAA) to avoid missed features
- Overall procedure (all in parallel on GPU)
  - A. Obtain seeds for line features
  - B. Line features by integration from seeds in feature flow field (FFF)

1. Octree subdivision based on RAA
2. Quadtree subdivision of octree faces based on RAA
3. Newton iterations for final refinement of seeds
4. Integration of line features in FFF

19 C. Pagot, D. Osmari, F. Sadlo, D.Weiskopf, T. Ertl, J. Comba: Efficient Parallel Vectors Feature Extraction from Higher-Order Data, Computer Graphics Forum, 30(3), pp. 751–760, 2011.

VISUALIZATION RESEARCH CENTER, UNIVERSITY OF STUTTGART

Selected References

[Fuchs '08] R. Fuchs, R. Peikert, F. Sadlo, B. Alsallakh, E. Gröller. Delocalized Unsteady Vortex Region Detectors. *Proceedings VMV 2008*, pp. 81–90, 2008.

[Haller '01] G. Haller. Distinguished material surfaces and coherent structures in three-dimensional fluid flows. *Physica D 149, pp.* 248–277, 2001.

[Haller '05] G. Haller. An objective definition of a vortex. *Journal of Fluid Mechanics,* 525:1–26, 2005.

[Jeong '95] J. Jeong, F. Hussain. On the identification of a vortex. *Journal of Fluid Mechanics,* 285:69–84, 1995.

[Pagot '11] C. Pagot, D. Osmari, F. Sadlo, D.Weiskopf, T. Ertl, J. Comba: Efficient Parallel Vectors Feature Extraction from Higher-Order Data, *Computer Graphics Forum*, 30(3), pp. 751–760, 2011.

[Sadlo '06] F. Sadlo, R. Peikert, M. Sick. Visualization Tools for Vorticity Transport Analysis in Incompressible Flow. *IEEE Trans. on Vis. and Comp. Graph.*, 12(5), pp. 949–956 2006.

[Sahner '05] J. Sahner, T. Weinkauf, H. C. Hege. Galilean Invariant Extraction and Iconic Representation of Vortex Core Lines. In Proc. *EuroVis 2005*, pp. 151–160, 2005.

[Sujudi '95] D. Sujudi, R. Haimes. Identification of swirling flow in 3D vector fields. *Technical Report AIAA-95-1715, American Institute of Aeronautics and Astronautics*, 1995.

[Üffinger '12] M. Üffinger, F. Sadlo, Ertl, A Time-Dependent Vector Field Topology Based on Streak Surfaces, *IEEE Trans. on Vis. and Comp. Graph.*, to appear, 2012.

[Weinkauf '07] T. Weinkauf, J. Sahner, H. Theisel, H.-C. Hege. Cores of Swirling Particle Motion in Unsteady Flows. *IEEE Trans. on Vis. and Comp. Graph.*, 13(6), 2007.

20 VISUS

VISUALIZATION RESEARCH CENTER, UNIVERSITY OF STUTTGART

Conclusion

- Advanced visualization techniques exist, but
  - have to be identified in visualization literature,
  - need to be integrated into CFD workflows,
  - interpretation and R&D processes need to be adapted
- Get in contact with visualization groups
  - They need data and new problems → win-win situation
  - Don't expect production-level software, but you will likely get prototypes and support to guide a new implementation

21 VISUS

VISUALIZATION RESEARCH CENTER, UNIVERSITY OF STUTTGART

Thank you for your attention!

Acknowledgments

SimTech Cluster of Excellence

SFB-TRR 75 Tropfendynamische Prozesse unter extremen Umgebungsbedingungen

22 VISUS

2.19 P.F. Berte (ONERA)

Deploying and managing a visualization farm at Onera

P.F. Berte

Onera - The French Aerospace Lab, Network and Computing Department(DRI)
BP72, 92322 Châtillon Cedex, France

CFD numerical simulation workflow involves many steps:

- Pre-processing: meshing, domain splitting...
- Solving (using for example Onera CFD solvers such as elsA, Cedre)
- Post-processing: Data mining, results visualization using tools like Paraview or commercial post-processing software (Tecplot, Ensight, ...)

Onera's IT department provides a mutualized infrastructure capable of fulfilling each step of this workflow, including storage, network and scientific clusters.

Nowadays, as CFD physical models' complexity increases, the main challenge of a computing center such as Onera's is to find the best way to optimize the use of its resources.

How to deliver the highest performance of the HPC infrastructure at the lowest cost, and share it fairly among end-users?

Since many years an answer has been found for the solving process: a job scheduling software allows clusters to increase resource utilization and reduce costs: works are embedded in a job envelope, and then submitted on the cluster. The scheduler manages priority (with mechanisms such as "fairshare"), resources (CPU cores, memory...) and do it - most of the time - efficiently.

But what about pre/post processing tasks such as mesh generation or visualization? Most of these tasks are "interactive" and/or "display intensive". So they were traditionally run on local workstations; it is clearly not the best way to optimize the cost/performance ratio. In order to improve this ratio, visualization farms are appearing in datacenters. They allow the end-user to launch and access remote interactive applications.

In this talk, we will see an example of a vizualisation farm deployment at Onera. We will also present an home-made reservation system whose aim is to deal with this optimization paradigm.

Deploying and managing a Visualization Farm @ Onera

Onera Scientific Day - October, 3 2012

Network and computing department (DRI), Onera

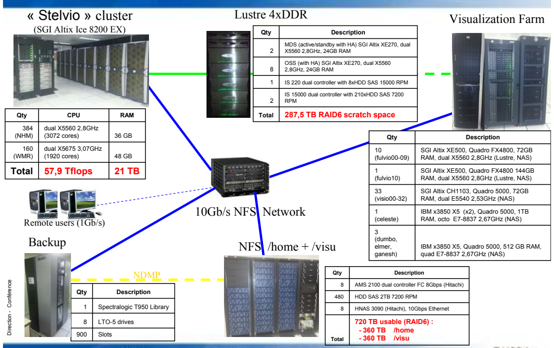
P.F. Berte
pierre-frederic.berte@onera.fr

Plan

- Onera global HPC Infrastructure
- Remote Visualization Introduction
  - What is it?
  - Why?
  - How?
- Remote visualization at Onera
  - in the past…
  - Remote display software selection
  - Towards a reservation system…
  - Home made Reservation system development and deployment : « visionera »
- What next?

Onera HPC Infrastructure

Onera HPC Infrastructure (1)



Qty	CPU	RAM
384 (NHM)	dual X5560 2,8GHz (3072 cores)	36 GB
160 (WMR)	dual X5675 3,07GHz (1920 cores)	48 GB
Total	57,9 Tflops	21 TB

Qty	Description
2	MDS (active/standby with HA) SGI Altix XE270, dual X5560 2,8GHz, 24GB RAM
8	OSS (with HA) SGI Altix XE270, dual X5560 2,8GHz, 24GB RAM
1	IS 220 dual controller with 8xHDD SAS 15000 RPM
2	IS 15000 dual controller with 210xHDD SAS 7200 RPM
Total	287,5 TB RAID6 scratch space

Qty	Description
10 (fulvio00-09)	SGI Altix XE500, Quadro FX4800, 72GB RAM, dual X5560 2,8GHz (Lustre, NAS)
1 (fulvio10)	SGI Altix XE500, Quadro FX4800 144GB RAM, dual X5560 2,8GHz (Lustre, NAS)
33 (visio00-32)	SGI Altix CH1103, Quadro 5000, 72GB RAM, dual E5540 2,53GHz (NAS)
1 (celeste)	IBM x3850 X5 (x2), Quadro 5000, 1TB RAM, octo E7-8837 2,67GHz (NAS)
3 (dumbo, elmer, ganesh)	IBM x3850 X5, Quadro 5000, 512 GB RAM, quad E7-8837 2,67GHz (NAS)

Qty	Description
1	Spectralogic T950 Library
8	LTO-5 drives
900	Slots

Qty	Description
8	AMS 2100 dual controller FC 8Gbps (Hitachi)
480	HDD SAS 2TB 7200 RPM
8	HNAS 3090 (Hitachi), 10Gbps Ethernet
Total	720 TB usable (RAID6) : - 360 TB /home - 360 TB /visu

Onera HPC Infrastructure (2)

- HPC Usage at Onera :
  - Mostly CFD
  - Electromagnetism
  - Structure Computation
- Solving (on « stelvio » cluster)
- Pre/Post processing & visualization :
  - Meshing (Icem CFD, Centaur…)
  - Visualization (Tecplot, Ensight, Paraview, VisIt…)
  - Requirements :
    - CPU & Memory
    - 3D intensive (OpenGL)
    - I/O throughput
    - interactive

Remote Visualization Introduction

Remote Visualization

- What is it?
  - Using a remote infrastructure to interact with data (pre, post processing, visualizing)
  - Remote Visualization allows to perform visualization on a remote compute system (with dedicated capabilities, such as 3D hardware acceleration), and have the display efficiently sent to the local client workstation/PC. [RZG]
- Why?
  - Flexibility
  - Cost
  - Efficient
  - Easy administration
  - Improved security

Remote display solutions

2 kinds of solutions :

- « Desktop class » remote display
  - Not graphically intensive, mostly 2D
  - Office tasks, development, remote administration…
  - Software : RDP (ICA…), VNC, X Server running on client (Unix/Linux)…, VDI
  - Not efficient for CFD tasks
- « High performance class » remote display
  - Typically for CAD, CFD, R&D Labs
  - Advanced 3D capabilities
  - High quality throughput
  - Software : Turbo VNC/Virtual GL, HP Remote Graphics (RGS), Nice / DCV…

Remote visualization at Onera

Onera's visualization history

Early :
- 1999 :
  - SGI Octane workstations (Irix)

The beginning of Remote visualization…

1999 :
- SGI Origin 2000 (Remote X Server) (Irix)

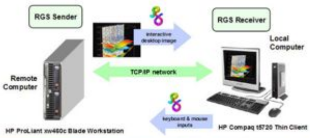
2004-2005 :
- SGI Prism SMP Server with VizServer (Irix)
- 32 x HP xw8200/xw8600 workstations (rack) with HP RGS (RHEL)

2010 :
- 10 x SGI Altix XE500, Quadro FX4800, 72GB RAM, dual X5560 2,8GHz (Lustre, NAS)
- 1 x SGI Altix XE500, Quadro FX4800 144GB RAM, dual X5560 2,8GHz (Lustre, NAS)

Remote display software @ Onera : RGS

Onera chose HP RGS in 2005 :

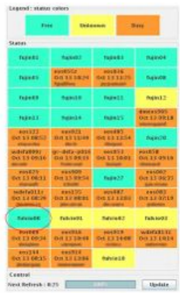
- 1:1 Remote display (1:many possible)
- OpenGL Support (Linux)
- Good ratio quality/speed, with easy quality adjustment / Network bandwidth
- PAM Authentication
- Easy integration into X server (Addon module)
- Windows & Linux Client



Towards a Reservation System… (1)

Early visualization farm management : « Cerbere »
- Home-made tool
- Java application
- Real time availability status of the remote workstations

⇨ not a reservation system
⇨ the first connected, the first served
⇨ Workstations used all the time !



Towards a Reservation System… (2)

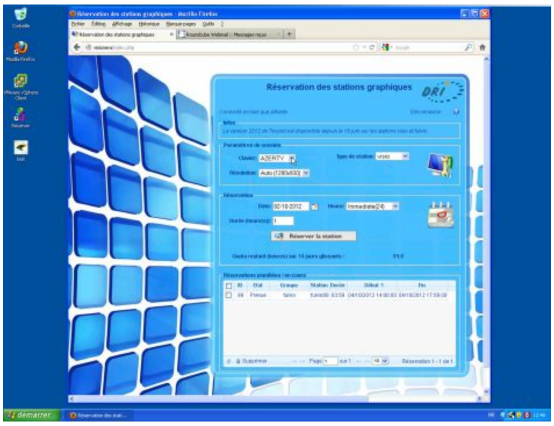
2011 :
Next-gen workstations replacing HP xw8200 :
- "Visio" systems (2011) :
  - 33 x SGI Altix CH1103, Quadro 5000, 72GB RAM, dual E5540 2,53GHz (NAS)
- Followed by big memory systems (2012) :
  - 1 x IBM x3850 X5 (x2), Quadro 5000, 1TB RAM, octo E7-8837 2,67GHz (NAS)
  - 3 x IBM x3850 X5, Quadro 5000, 512 GB RAM, quad E7-8837 2,67GHz (NAS)

⇨ Need to improve access to workstations so that every user can connect !

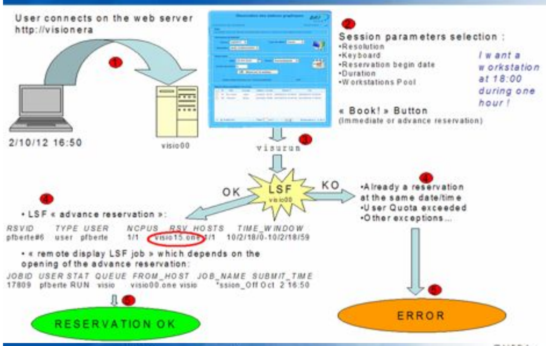
- Commercial ?
  - Oxalya Visuportal
  - Nice DCV
  - Vizstack
  - No machine NX
- Home made ?

Onera reservation system specifications

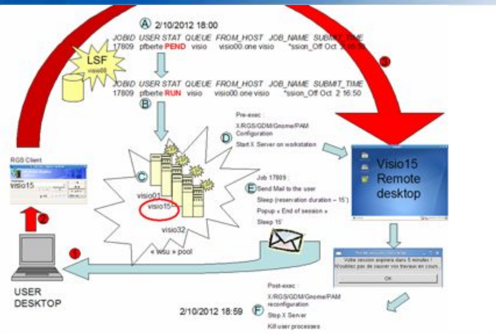
- Engine :
  - Platform LSF batch engine
  - Perl scripts (visurun/visudel/visushow)
  - C
  - Extended with a web frontend (PHP/Ajax/JQuery)
- 2 reservation modes :
  - Instant reservation
  - Advance reservation
- User must be able to customize his session :
  - Graphic resolution
  - Keyboard settings
  - « Pool » of stations
  - Duration
- Limits :
  - Maximum duration of a graphic session
  - User quota (sliding window)
  - User can only book 1 station at a time
  - « Day-off / Day-on » policies



Behind the Onera reservation system (1)



Behind the Onera reservation system (2)



ONERA

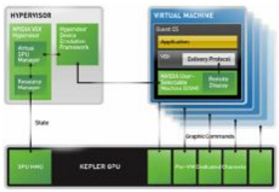
Further developments

- Collaborative work?
- Application selection?
- Parallel rendering?
- License management?
- …

ONERA

What's next in remote visualization ? VDI ?

- VDI = Virtual Desktop Infrastructure
  - GPU Virtualization on existing hypervisors :
    - Citrix, VMWare, Microsoft
      - API Interceptor or GPU Passthrough
- NVidia's Monterey Project
  ⇨ VGX platform



ONERA

ARISTOTE Association Loi de 1901. Siège social : CEA-DSI CEN Saclay Bât. 474, 91191 Gif-sur-Yvette Cedex.
Secrétariat : Aristote, École Polytechnique, 91128 Palaiseau Cedex.
Tél. : +33(0)1 69 33 99 66 Fax : +33(0)1 69 33 99 67 Courriel : Marie.Tetard@polytechnique.edu
Site internet http://www.association-aristote.fr