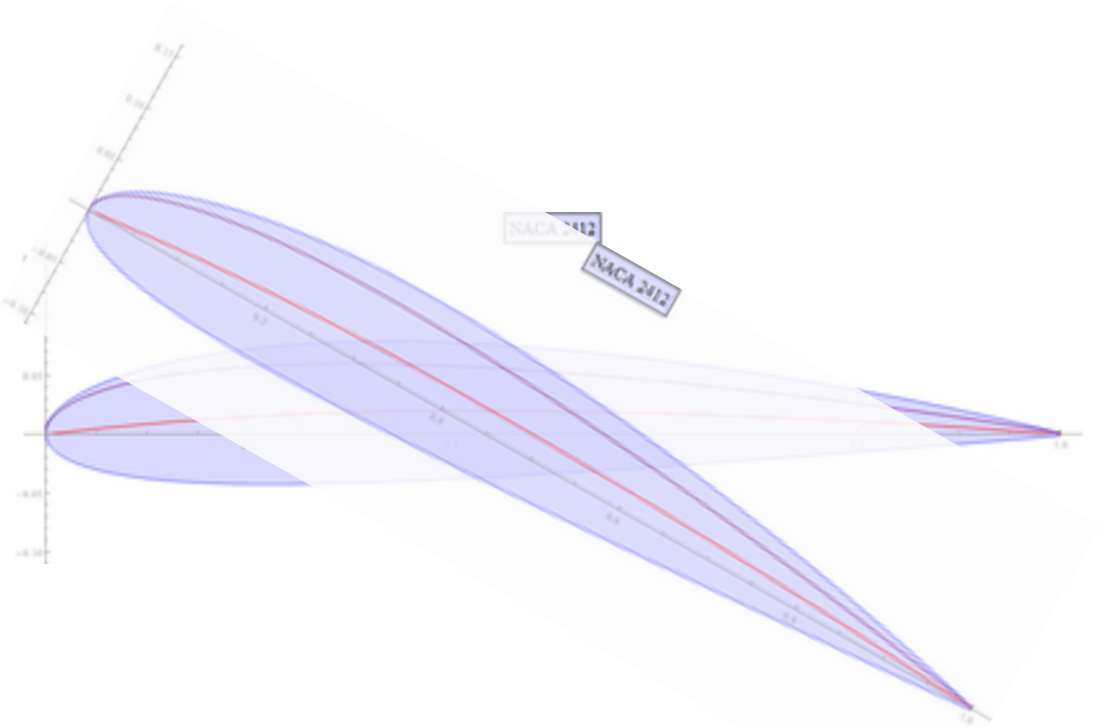
Introduction to Literature Seminar and Mini-Projects

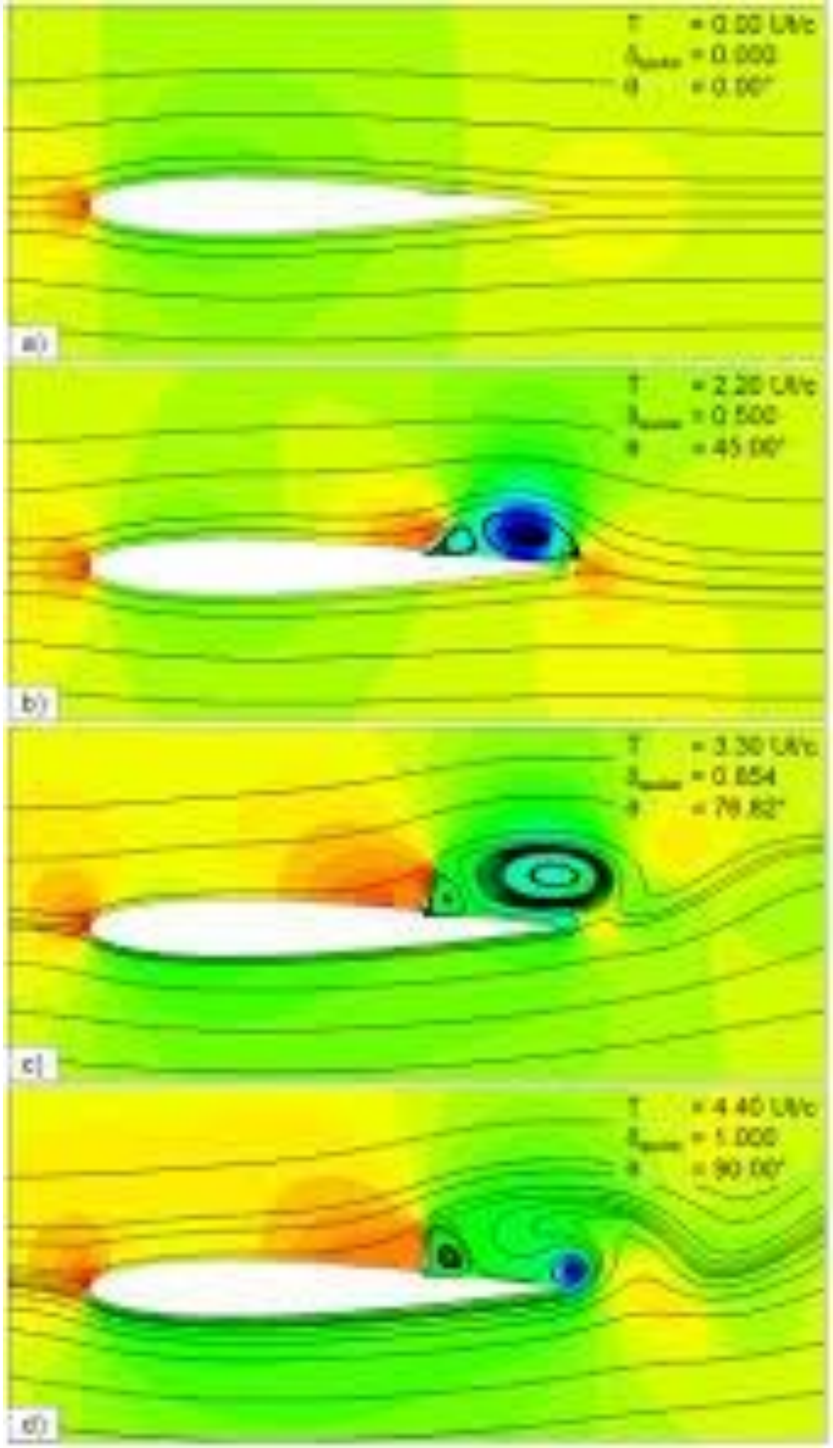
Project 1: Airfoil Optimization



https://en.wikipedia.org/wiki/NACA_airfoil

How does the lift-force depend on the angle of attack?

Navier-Stokes Equations



Calculate airflow around the wing profile.

https://www.youtube.com/watch?v=H-snLmMQK0Y

FEniCS:
Finite Element Simulations

The FEniCS Project is a collection of free software with an extensive list of features for automated, efficient solution of differential equations.

Through this web site, you can learn more about the project and learn how to obtain and how to use our software. We'd be delighted to offer support in case you need it, and encourage contributions from our users.

FEniCS'15 at Imperial College London

We are happy to announce that the 2015 FEniCS Workshop will be held on June 29 - July 1 2015 at Imperial College London, London, UK. More information can be found at the official FEniCS '15 webpage.

Periodic Table of the Finite Elements

Find your favorite FEniCS elements in the Periodic Table of Finite Elements. The periodic table displays and explains the properties of many of the most widely used elements, including the Lagrange, RT, BDM, Nedelec and DG elements.

FEniCS book released!

The FEniCS book titled *Automated Solution of Differential Equations by the Finite Element Method* has been published in 2012 as Volume 84 of the Springer Lecture Notes in Computational

Recent project news

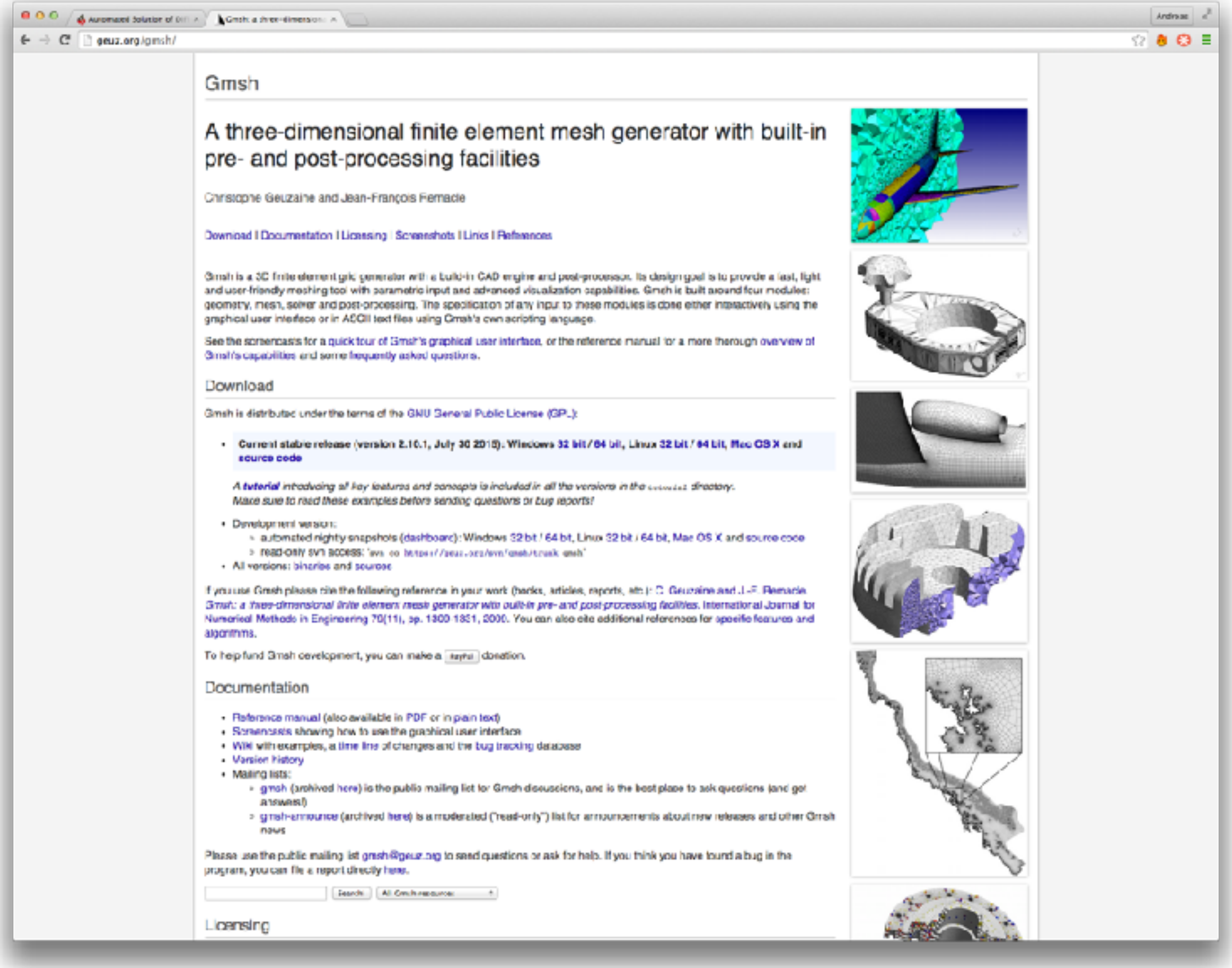
Anders Logg @anderslogg 14 Sep
#fenicsnews twitter.com/jack_s_hale/s…

Anders Logg @anderslogg 7 Sep
Checkout FEniCS on #dockerhub #fenicsnews twitter.com/jack_s_hale/st…
Expand

Johannes Ring @johannesring 20 Aug
Binary packages for the 1.6.0 release of FEniCS are now available for Ubuntu and OS X 10.10. #fenicsnews
Expand

Anders Logg @anderslogg 13 Aug
Found this Russian candidate thesis (?) doing poroelasticity with FEniCS. Nice! #fenicsnews s-vfu.ru/upload/iblock/… pic.twitter.com/TEBubvMaNv

Gmsh



Gmsh

A three-dimensional finite element mesh generator with built-in pre- and post-processing facilities

Christophe Geuzaine and Jean-François Remacle

Download | Documentation | Licensing | Screenshots | Links | References

Gmsh is a 3D finite element grid generator with a build-in CAD engine and post-processor. Its design goal is to provide a fast, light and user-friendly meshing tool with parametric input and advanced visualization capabilities. Gmsh is built around four modules: geometry, mesh, solver and post-processing. The specification of any input to these modules is done either interactively using the graphical user interface or in ASCII text files using Gmsh's own scripting language.

See the screencasts for a quick tour of Gmsh's graphical user interface, or the reference manual for a more thorough overview of Gmsh's capabilities and some frequently asked questions.

Download

Gmsh is distributed under the terms of the GNU General Public License (GPL):

- **Current stable release (version 2.10.1, July 30 2015): Windows 32 bit / 64 bit, Linux 32 bit / 64 bit, Mac OS X and source code**

  *A **tutorial** introducing all key features and concepts is included in all the versions in the* `tutorial` *directory.*
  Make sure to read these examples before sending questions or bug reports!

- Development version:
  - automated nightly snapshots (dashboard): Windows 32 bit / 64 bit, Linux 32 bit / 64 bit, Mac OS X and source code
  - read-only svn access: `svn co https://geuz.org/svn/gmsh/trunk gmsh`
- All versions: binaries and sources

If you use Gmsh please cite the following reference in your work (books, articles, reports, etc.): 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), pp. 1309-1331, 2009. You can also cite additional references for specific features and algorithms.

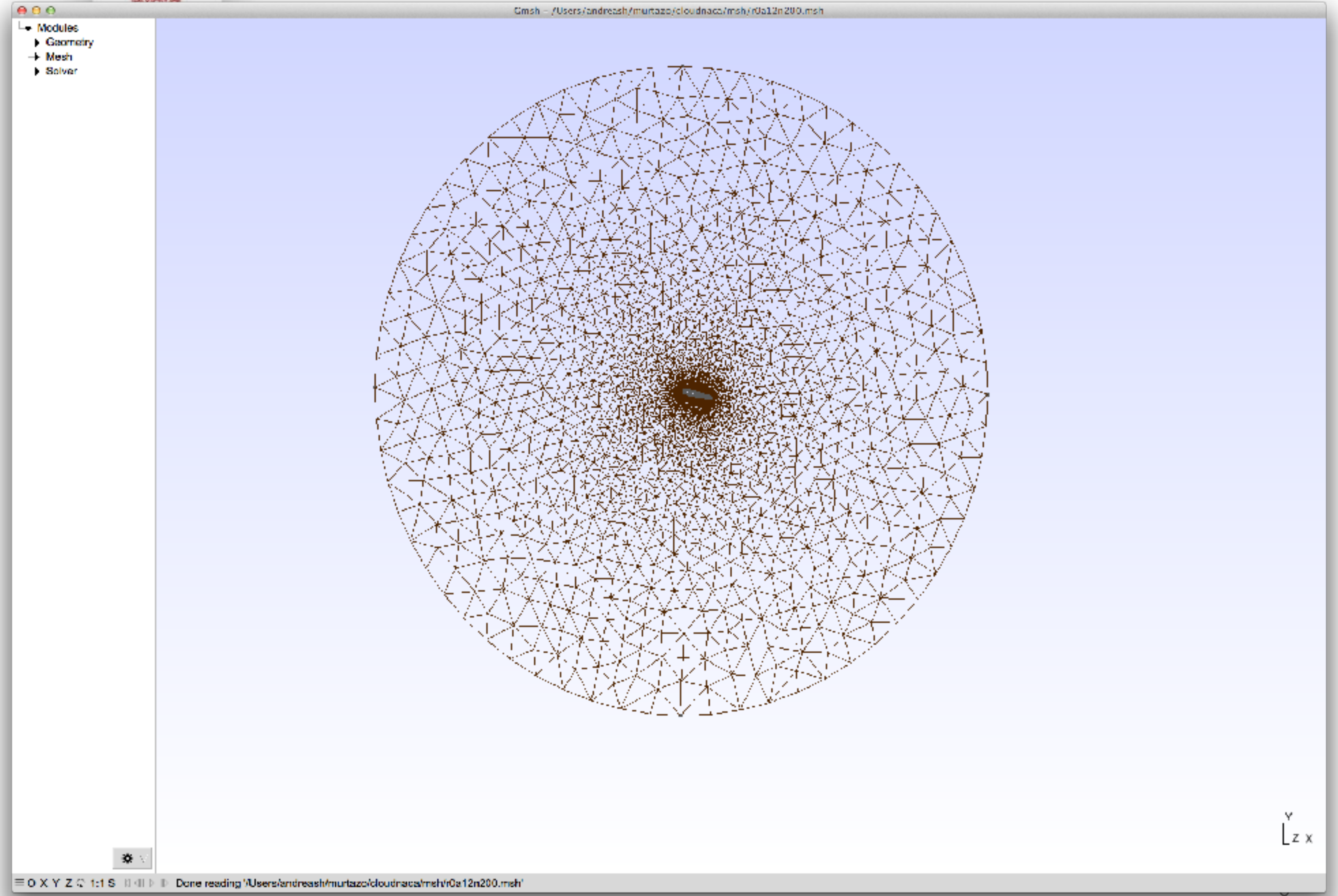
To help fund Gmsh development, you can make a PayPal donation.

Documentation

- Reference manual (also available in PDF or in plain text)
- Screencasts showing how to use the graphical user interface
- Wiki with examples, a time line of changes and the bug tracking database
- Version history
- Mailing lists:
  - gmsh (archived here) is the public mailing list for Gmsh discussions, and is the best place to ask questions (and get answers!)
  - gmsh-announce (archived here) is a moderated ("read-only") list for announcements about new releases and other Gmsh news

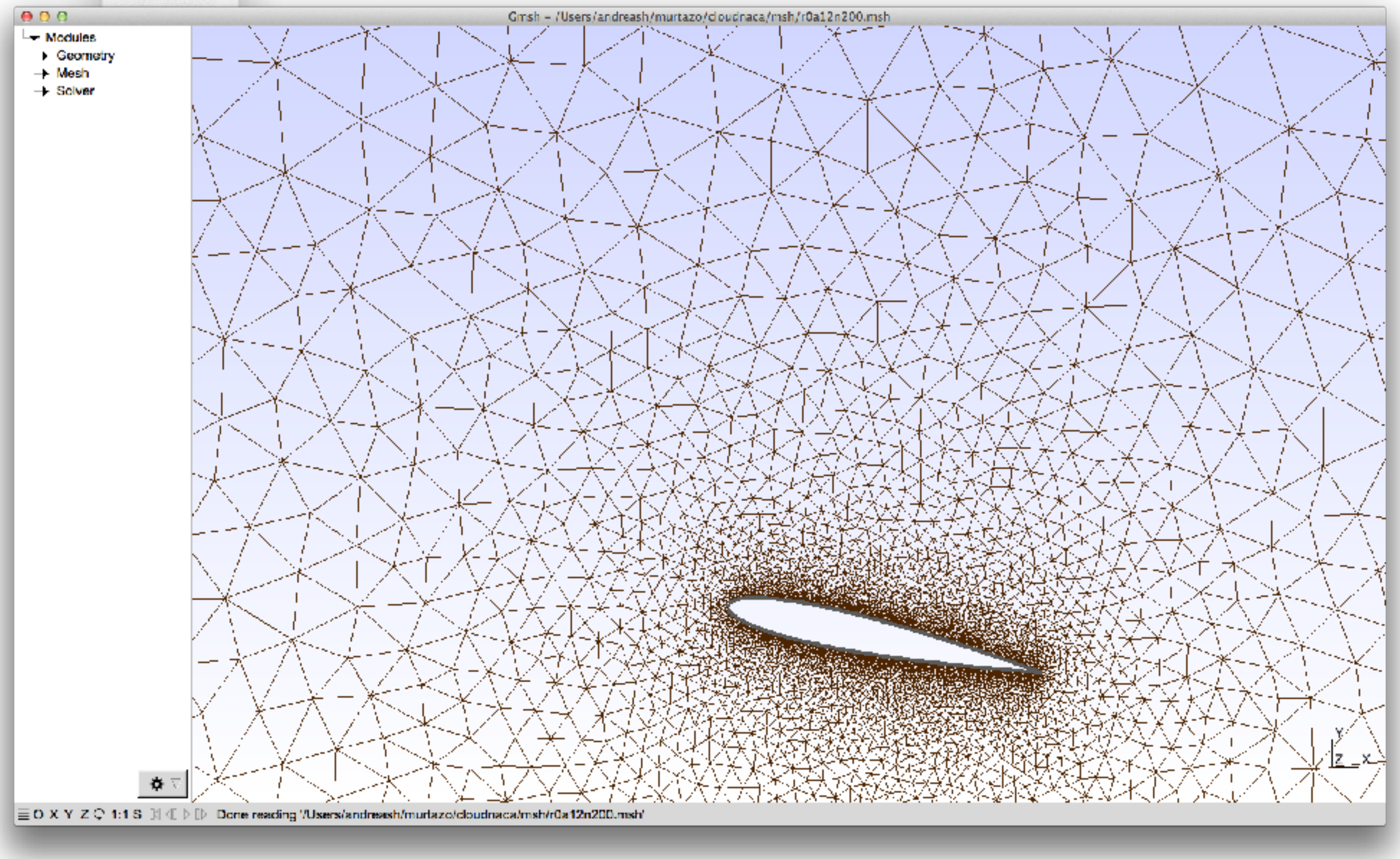
Please use the public mailing list gmsh@geuz.org to send questions or ask for help. If you think you have found a bug in the program, you can file a report directly here.

Licensing



Gmsh - /Users/andreash/murtazo/cloudnaca/msh/r0a12n200.msh

Done reading '/Users/andreash/murtazo/cloudnaca/msh/r0a12n200.msh'



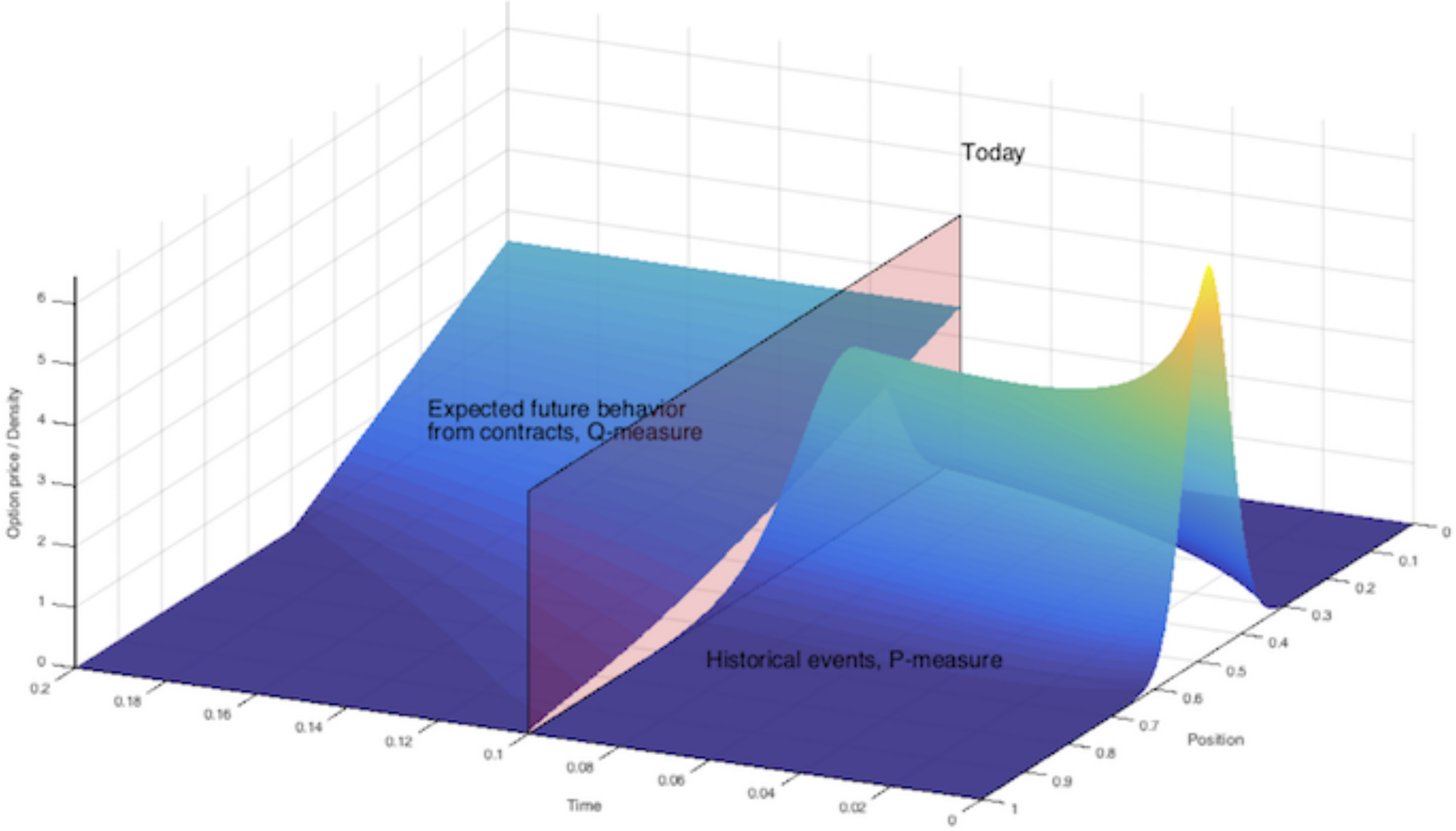
Gmsh - /Users/andreash/murtazo/cloudnaca/msh/r0a12n200.msh

Done reading '/Users/andreash/murtazo/cloudnaca/msh/r0a12n200.msh'

Tasks

- Design and implement a cloud-based solution and service for conducting experiments to assess the influence of the angle of attack on the lift force.
- As a minimal requirement, the system needs to work for a statically configured input (apart from the meshes) and there needs to be an assessment of the horizontally scalability of your solution.
- …

BENCHOP-as-a-Service (BaaS)



http://www.it.uu.se/research/project/compfin/benchop/

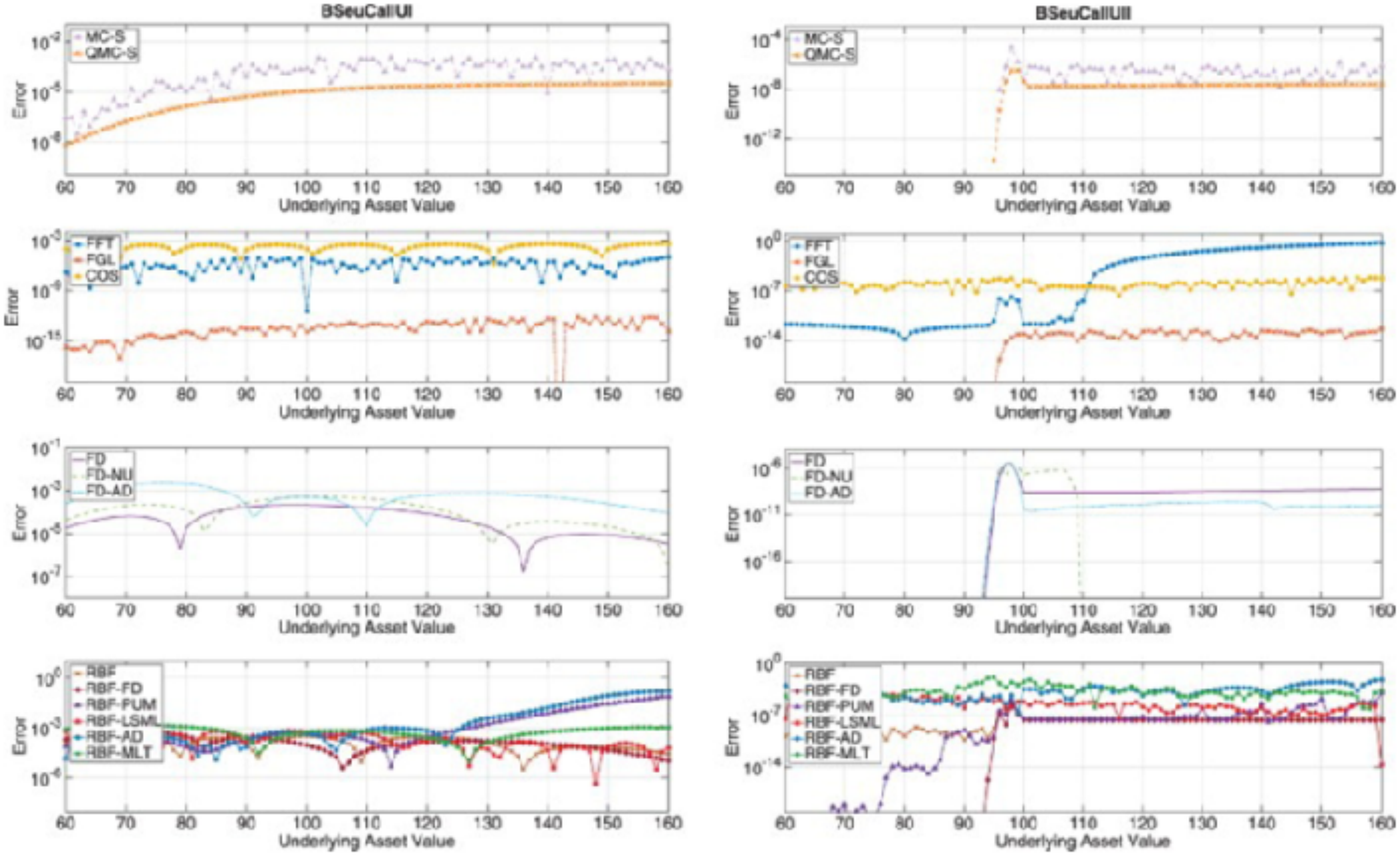
BENCHOP

- The BENCHOP project is a large, international collaborative project initiated by the Computational Finance research group at the Division of Scientific Computing.

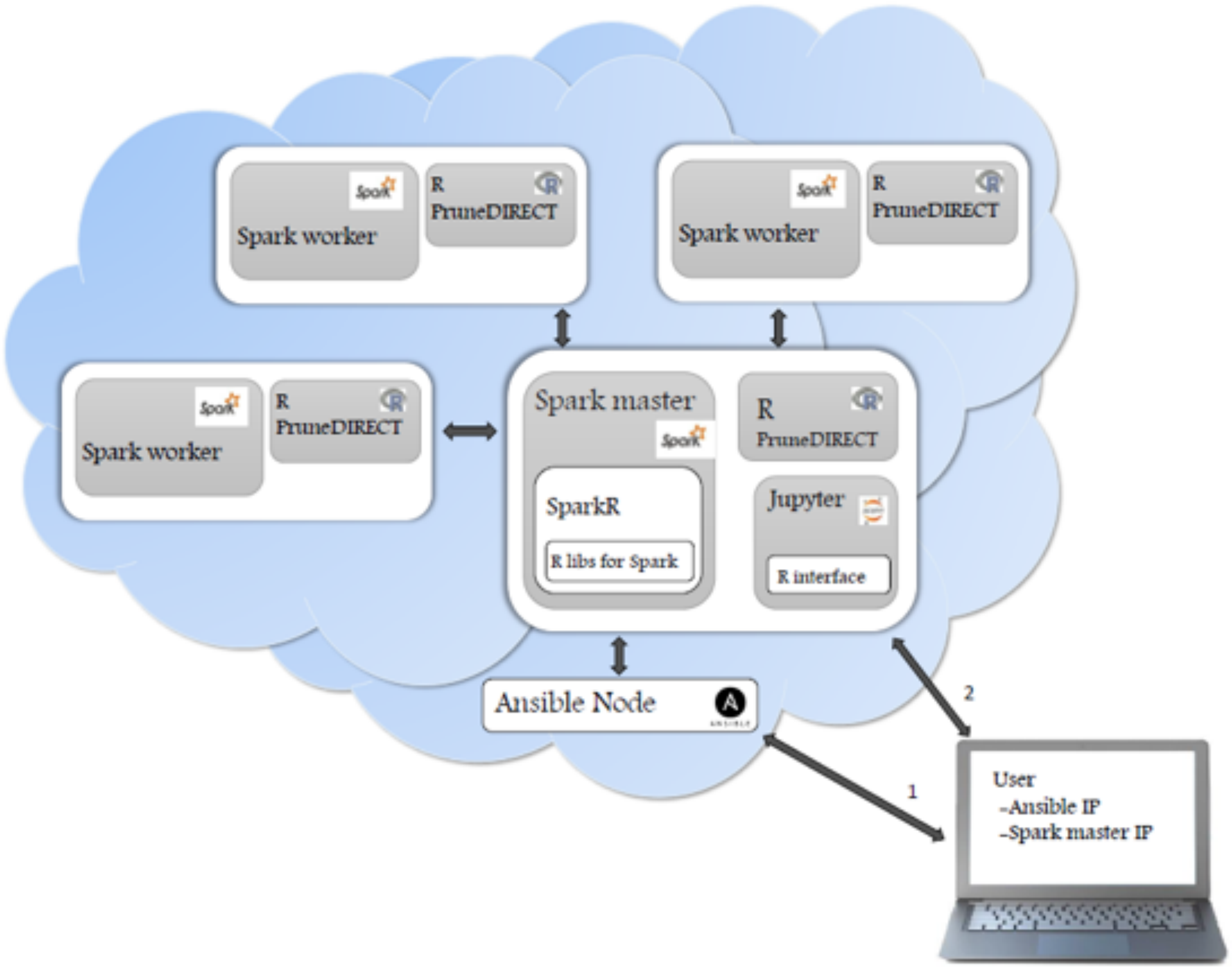
http://www.it.uu.se/research/scientific_computing/project/compfin

BENCHOP

- BENCHOP aims at defining a standard benchmark to compare existing and new numerical methods for option pricing.



Project 3: QTL as a Service (QTLaaS), a cloud service for genetic analysis



QTL Analysis

- Abbreviation of “ Quantitative Trait Loci ”
- Understanding the relation between genes and traits is a fundamental problem in genetics.
- Such knowledge can eventually lead to e.g. the identification of possible:
  - drug targets,
  - treatment of heritable diseases, and
  - efficient designs for plant and animal breeding.
- QTL mapping is the way to locate such regions in genome.

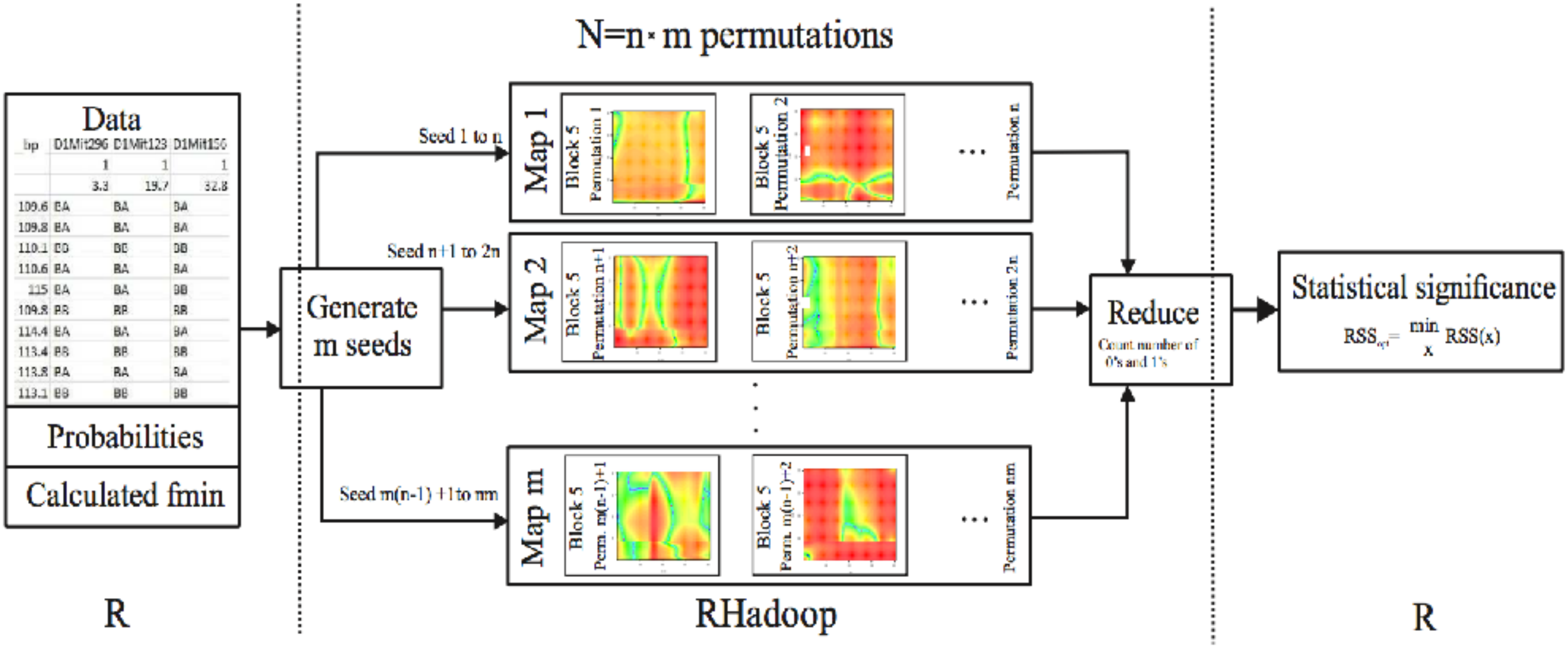
UPPSALA
UNIVERSITET

A flexible computational framework using R and Map-Reduce for permutation tests of massive genetic analysis of complex traits

Citation information: DOI 10.1109/TCBB.2016.2527639, IEEE/ ACM Transactions on Computational Biology and Bioinformatics

Authors: Behrang Mahjani, Salman Toor, Carl Nettelblad,
Sverker Holmgren

Analysis Framework



N=n·m permutations

Data

bp	D1Mit296	D1Mit123	D1Mit156
	1	1	1
	3.3	19.7	32.8
109.6	BA	BA	BA
109.8	BA	BA	BA
110.1	BB	BB	BB
110.6	BA	BA	BA
115	BA	BA	BB
109.8	BB	BB	BB
114.4	BA	BA	BA
113.4	BB	BB	BB
113.8	BA	BA	BA
113.1	BB	BB	BB

Probabilities

Calculated fmin

Generate m seeds

Seed 1 to n

Seed n+1 to 2n

Seed m(n-1) +1 to nm

Map 1

Block 5 Permutation 1

Block 5 Permutation 2

...

Permutation n

Map 2

Block 5 Permutation n+1

Block 5 Permutation n+2

...

Permutation 2n

Map m

Block 5 Perm. m(n-1)+1

Block 5 Perm. m(n-1)+2

...

Permutation nm

Reduce

Count number of 0's and 1's

Statistical significance

$$RSS_{opt} = \min_{x} RSS(x)$$

R

RHadoop

R

Fig. 4: Distributed computing of the significance level using map-reduce setting; Details of block 4

Results

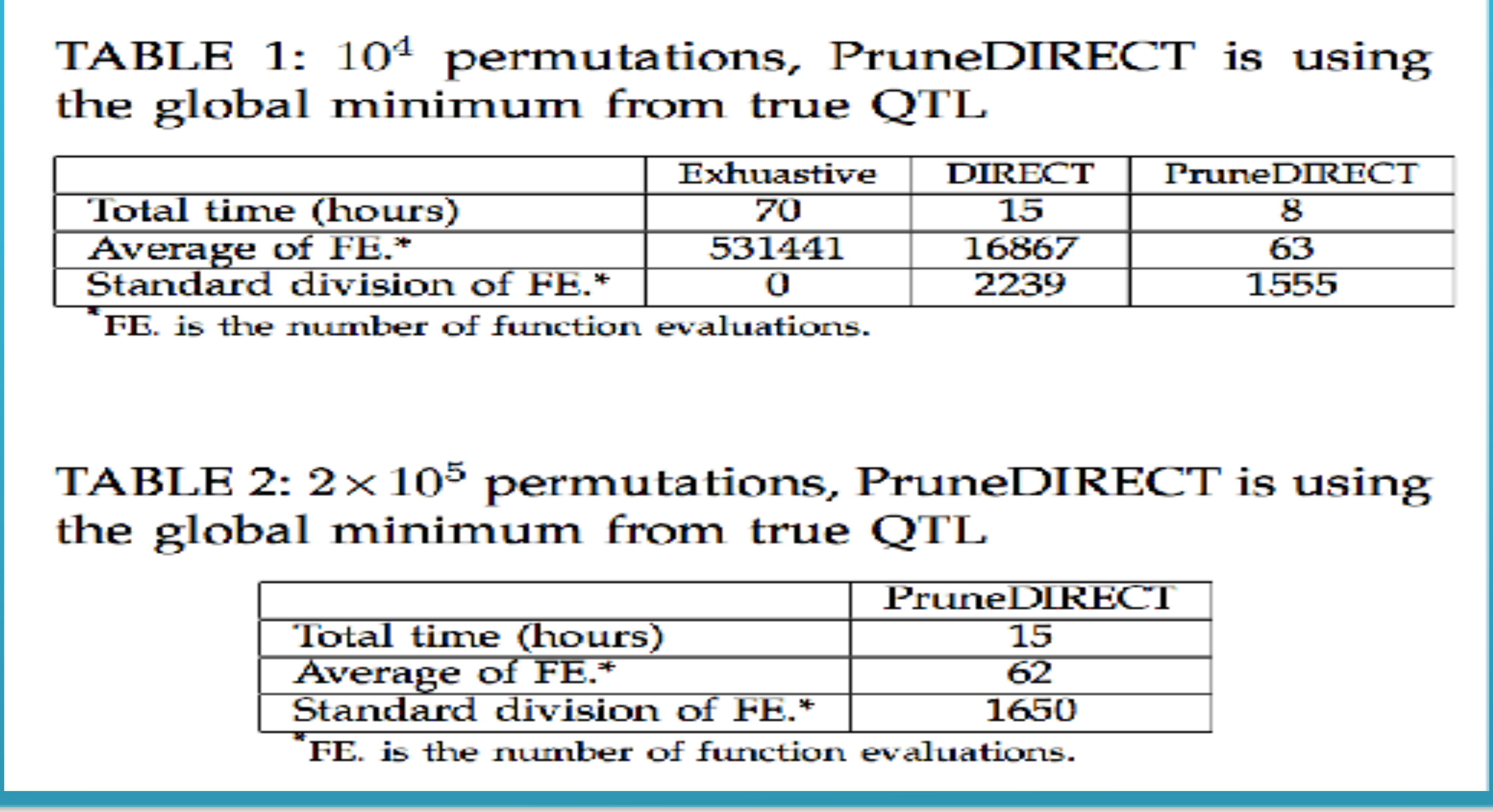


TABLE 1: 10^4 permutations, PruneDIRECT is using the global minimum from true QTL

	Exhuastive	DIRECT	PruneDIRECT
Total time (hours)	70	15	8
Average of FE.*	531441	16867	63
Standard division of FE.*	0	2239	1555

*FE. is the number of function evaluations.

TABLE 2: 2×10^5 permutations, PruneDIRECT is using the global minimum from true QTL

	PruneDIRECT
Total time (hours)	15
Average of FE.*	62
Standard division of FE.*	1650

*FE. is the number of function evaluations.

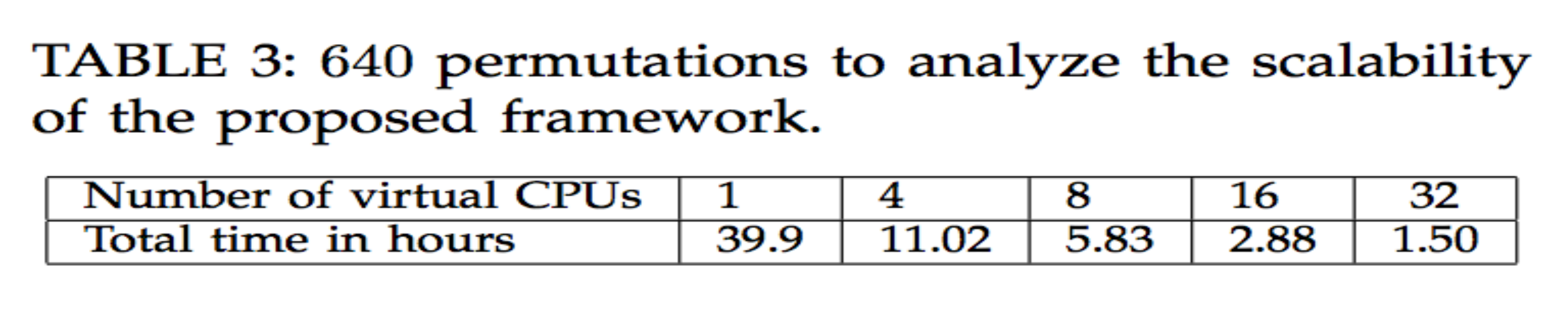


TABLE 3: 640 permutations to analyze the scalability of the proposed framework.

Number of virtual CPUs	1	4	8	16	32
Total time in hours	39.9	11.02	5.83	2.88	1.50

Questions

- Can you speed up the evaluation of the benchmark?

- Can you make a REST API/interface that makes it possible to recompute the benchmark for e.g. the problem “1a” but for different sets of parameters?

- Can you automate the process of uploading a new method?

- The performance of a program always depends on the hardware and software environment it is executed in. To what degree depends on the particular code. Will the relative ranking of methods in the original BENCHOP stay the same on different instance flavors?

Mini-Project Goals

Learning Outcomes

- Use public and private cloud solutions for computational science and engineering applications.

- Discuss key concepts of cloud computing services, such as Infrastructure as a Service (IaaS), Platform as a Service (PaaS) and Software as a Service (SaaS).

- Asses the suitability of cloud computing infrastructures for different scientific applications.

- Implement, in collaboration with peers, software for cloud-based distributed computing using the technology presented in the course.

- Critically analyze and present solutions and implementations in writing and orally.

Tips for getting started

Set up a start-up meeting and:

- Decide on “rules” for the group work. For example, how will you communicate, timelines etc.
- Make sure that you all have access to the Git/BitBucket repository.
- Agree on how to work with the code, is it OK to push to the master branch? Should you use feature branches, or will you work mainly with forks and Pull requests. Policies for code review?
- Assign shared responsibilities, is it possible to divide up some responsibilities on individuals, such as reading selected references and summarizing them to each other?

Writing the report

- Should be formatted as a short scientific paper
- Model it after a paper you have read during the course.
- Max 3000 words.
- Even though short, it should have all the typical parts of a scientific paper, including references to relevant literature.

Grading Criteria

Quality of analysis and software development
To assess the quality of the proposed solution we will look at:

- How well does the taken approach fit the problem?
- How well can we expect the chosen approach to scale?
  - You need to demonstrate the scaling properties of your solution.

- We will look at the activity in GitHub or BitBucket, such as commit histories, issues/discussions, pull requests etc.
- How well is the code structured, annotated and usable by a third party?
  - Good solutions may not require a lot of code to be written. Attempt to use external libraries/tools whenever possible.
  - Is there some degree of automation? Is resources used statically or dynamically etc.?
  - How hard is it to use or extend the code/service?

Grading Criteria

Quality of the project paper

To assess the quality of the project paper, we will look at:

- How clearly is the problem stated?
- How well is the problem contextualized?
- How well does it cover related, relevant work?
    - This is supposed to be a short project, and we do not expect an extensive bibliography. We will look at how relevant the cited related work is and if it is accurately used to support the discussion, rather than the sheer amount of references. Aim at approximately 4-5 references. You may of course cite papers from the literature seminar if they are relevant.

- How well do you argue for you proposed solution?
- How well do you demonstrate the suitability of your solution?
    - Performance benchmarks? Weak scaling properties?
    - How well are limitations discussed?

- Language: Is the paper clearly written using correct English?

Introduction to version control and project management with Git

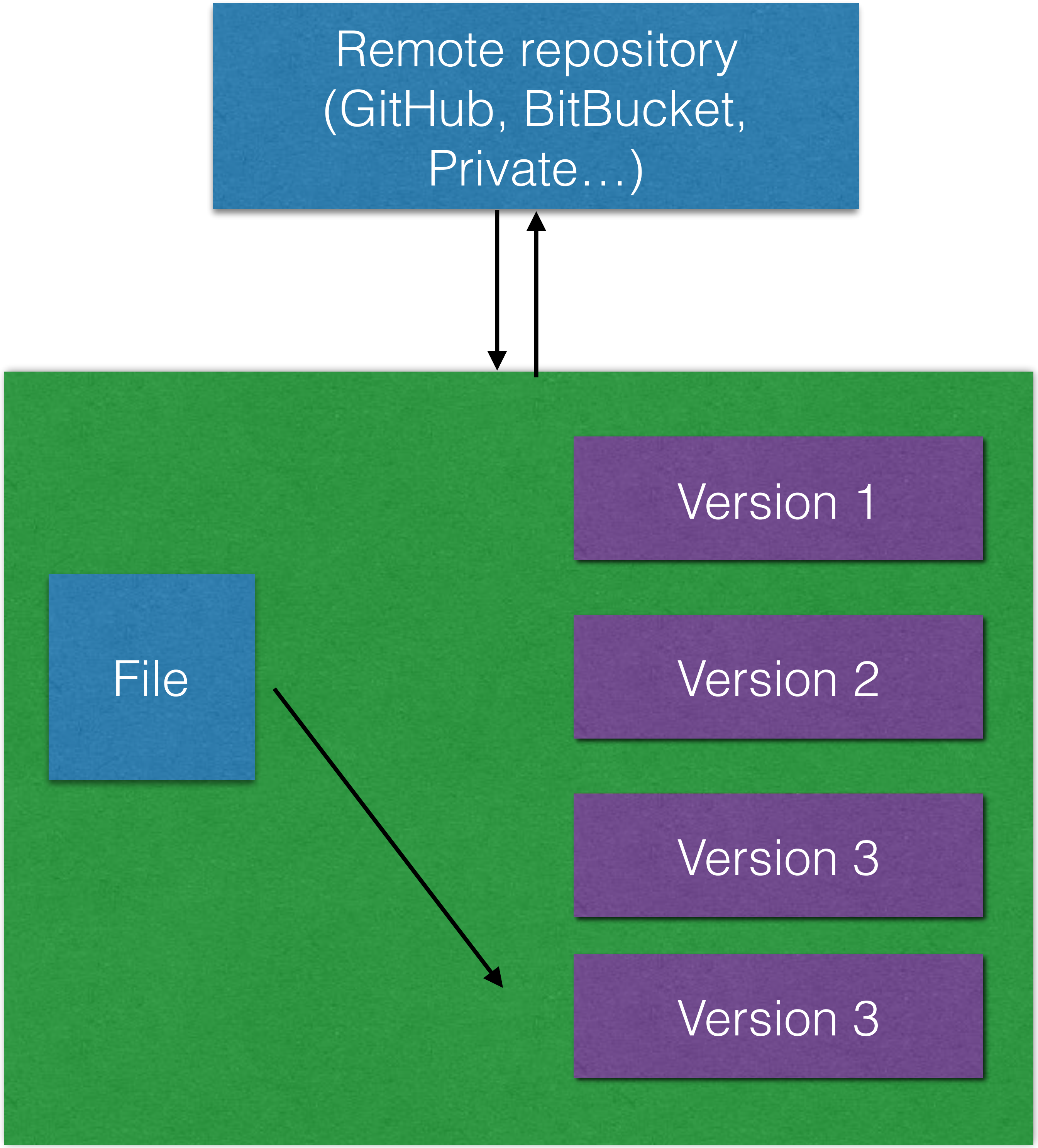
Use of a version control system is essential for all code development. There are nice, simple, popular web services for making that easy, and that add a lot of value in terms of project management. Very useful for own development, but the benefits are even more apparent when working in teams.

Have you worked with Subversion (SVN)?

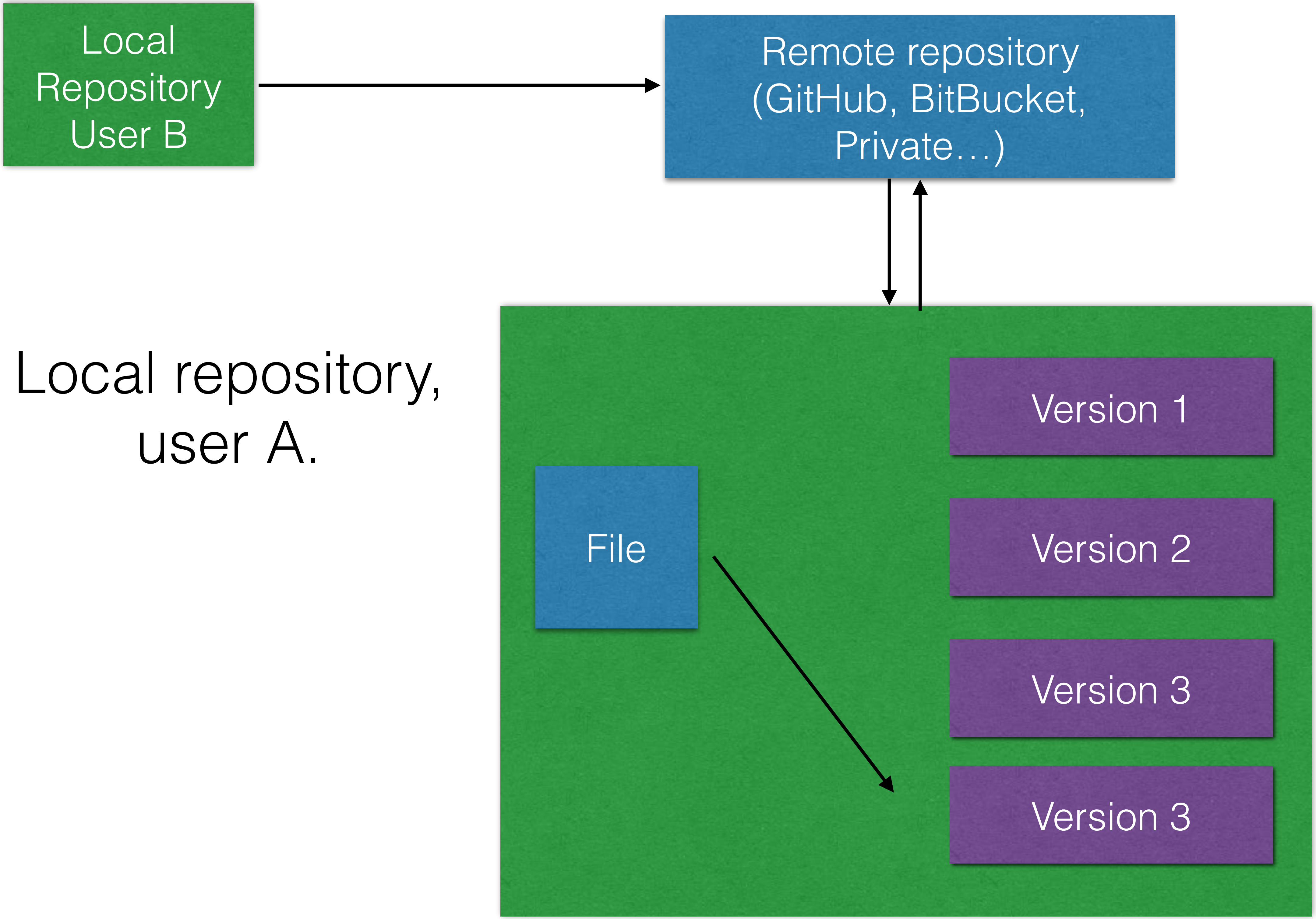
Then you will understand the basics. Git supports the typical centralized workflow often used together with SVN. But beware, there are important differences.

Git:

Local repository, user A.



Local repository, user A.



Local Repository User B

Remote repository (GitHub, BitBucket, Private…)

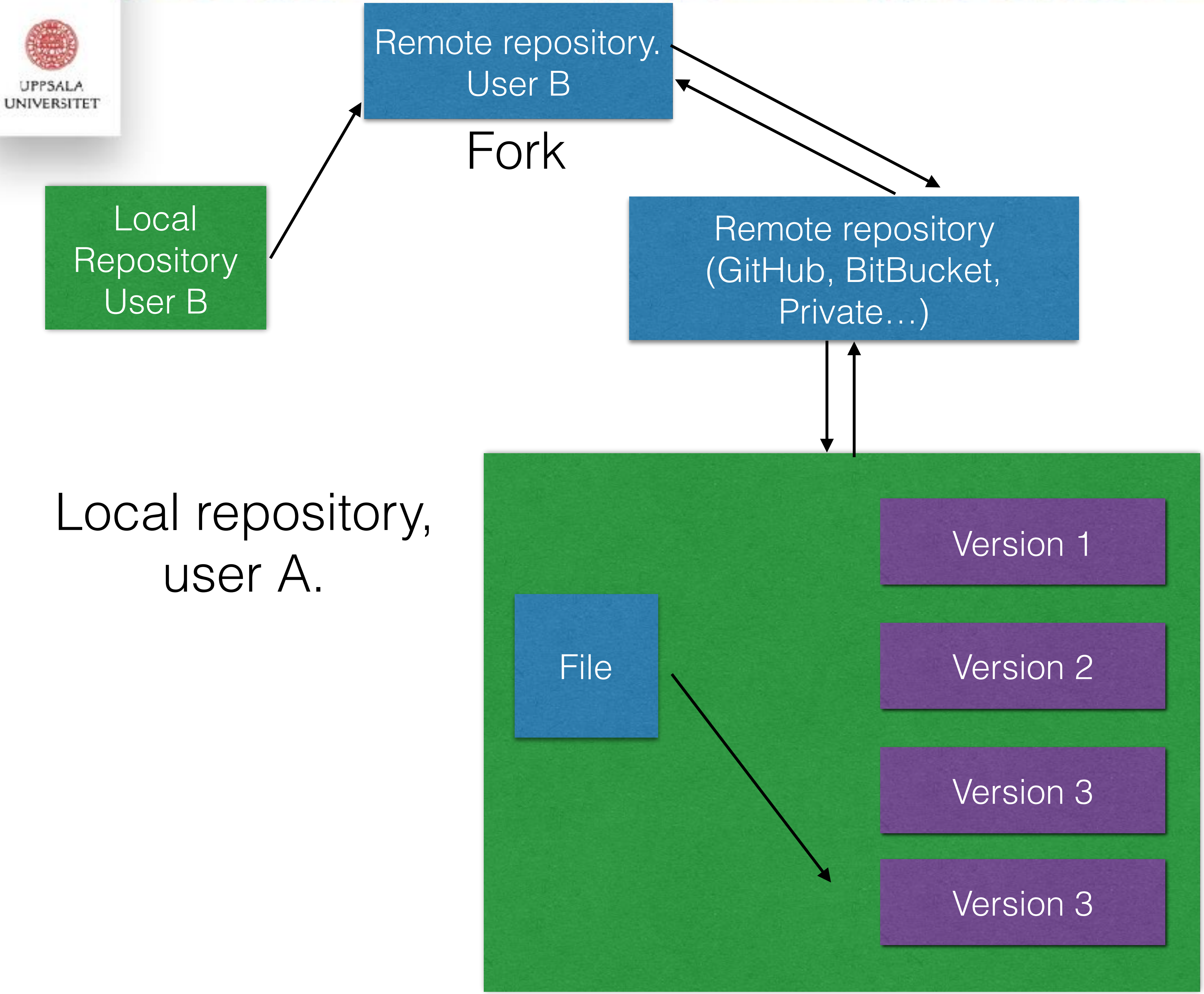
File

Version 1

Version 2

Version 3

Version 3



Remote repository. User B

Fork

Local Repository User B

Remote repository (GitHub, BitBucket, Private…)

Local repository, user A.

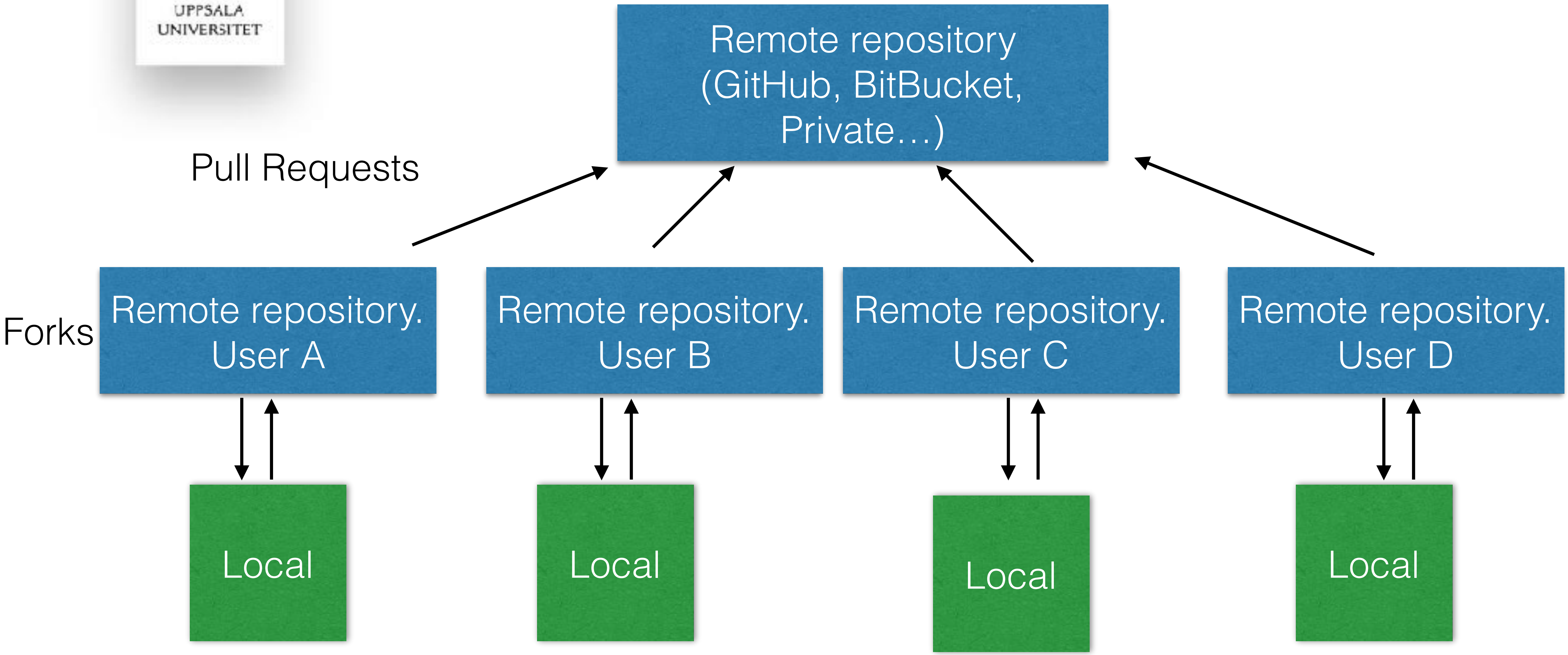
File

Version 1

Version 2

Version 3

Version 3


Remote repository (GitHub, BitBucket, Private…)
Pull Requests
Forks
Remote repository. User A
Remote repository. User B
Remote repository. User C
Remote repository. User D
Local
Local
Local
Local

A fork is not a way to “steal code”, it is a convenient way to start contributing to a project that does not require granting write permissions to the original repository. It is a popular model in the OpenSource community.

“Practical Tips”…

Don’t treat your VMs as your babies. Rather, treat it as another software component.

- Count on the VMs failing occasionally.
- If you have put in extensive work in manually installing software, consider using some contextualization tool to be able to automate the process.
- Consider the complexity in maintaining the environment when designing your solution, a complex stack takes more time to maintain.
- Keep critical data, such as your code, in external repositories at all times.
- Terminate unused VMs, they may put unnecessary strain on the system.

Are now requirements of a good solution.