Master in High Performance Computing

Theory and Practice of Finite Element Methods

Setting up a “best practice” FEM development environment
(some slides taken from prof. Wolfgang Bangerth)

Luca Heltai <luca.heltai@sissa.it>

International School for Advanced Studies (www.sissa.it)
Mathematical Analysis, Modeling, and Applications (math.sissa.it)
Master in High Performance Computing (www.mhpc.it)
SISSA mathLab (mathlab.sissa.it)

sissa mathLab innovating with mathematics

ICTP The Abdus Salam International Centre for Theoretical Physics

SISSA

Why deal.II (or any other Finite Element library)

- The numerical solution of partial differential equations is an immensely vast field!
- It requires us to know about:
  - Partial differential equations
  - Methods for discretizations, solvers, preconditioners
  - Programming
  - Adequate tools
  - **This course will cover all of this to some degree!**

Numerics of PDEs

There are 3 standard tools for the numerical solution of PDEs:

Finite element method (FEM)

Finite volume method (FVM)

Finite difference method (FDM)

Common features:

Split the domain into small volumes (cells)

Define balance relations on each cell

Obtain and solve very large (non-)linear systems

Problems:

Every code has to implement these steps

There is only so much time in a day

There is only so much expertise anyone can have

Numerics of PDEs

There are 3 standard tools for the numerical solution of PDEs:

Finite element method (FEM)

Finite volume method (FVM)

Finite difference method (FDM)

Common features:

Split the domain into small volumes (cells)

Define balance relations on each cell

Obtain and solve very large (non-)linear systems

In addition:

We don't just want a simple algorithm

We want state-of-the-art methods for everything

Numerics of PDEs

Examples of what we would like to have:

Adaptive meshes

Realistic, complex geometries

Quadratic or even higher order elements

Multigrid solvers

Scalability to 1000s of processors

Efficient use of current hardware

Graphical output suitable for high quality rendering

Q: How can we make all of this happen in a single code?

The hard reality

- Most research software today:
  - **Written by graduate students**
    - without a good overview of existing software
    - with little software experience
    - with little incentive to write high quality code
  - **Maintained by postdocs**
    - with little time
    - who need to consider the software primarily as a tool to publish papers
  - **Advised by faculty**
    - with no time
    - oftentimes also with little software experience

How we develop Software

Q: How can we make all of this happen in a single code?

Not a question of feasibility but of how we develop software:
Is every student developing their own software?
Or are we re-using what others have done?

Do we insist on implementing everything from scratch?
Or do we build our software on existing libraries?

How we develop Software

Q: How can we make all of this happen in a single code?

Not a question of feasibility but of how we develop software:
Is every student developing their own software?
Or are we re-using what others have done?

Do we insist on implementing everything from scratch?
Or do we build our software on existing libraries?

There has been a major shift on how we approach the second question in scientific computing over the past 10-15 years!

The secret to good scientific software is (re)using existing libraries!

Existing Software

There is excellent software for almost every purpose!

Basic linear algebra (dense vectors, matrices):
BLAS
LAPACK

Parallel linear algebra (vectors, sparse matrices, solvers):
PETSc
Trilinos

Meshes, finite elements, etc:
deal.II – the topic of this class
…

Visualization, dealing with parameter files, ...

Our experience

It is realistic for a student developing numerical methods using external libraries to have a code at the end of a PhD time that:

- Works in 2d and 3d
- On complex geometries
- Uses higher order finite element methods
- Uses multigrid solvers or preconditioners
- Solves a nonlinear, time dependent problem

Doing this from scratch would take 10+ years.

Common arguments...

Arguments against using other people's packages:

I would need to learn a new piece of software, how it works, its conventions. I would have to find my way around its documentation. Etc.

I think I'll be faster writing the code I want myself!

Common arguments...

Arguments against using other people's packages:

I would need to learn a new piece of software, how it works, its conventions. I would have to find my way around its documentation. Etc.

I think I'll be faster writing the code I want myself!

Answers:

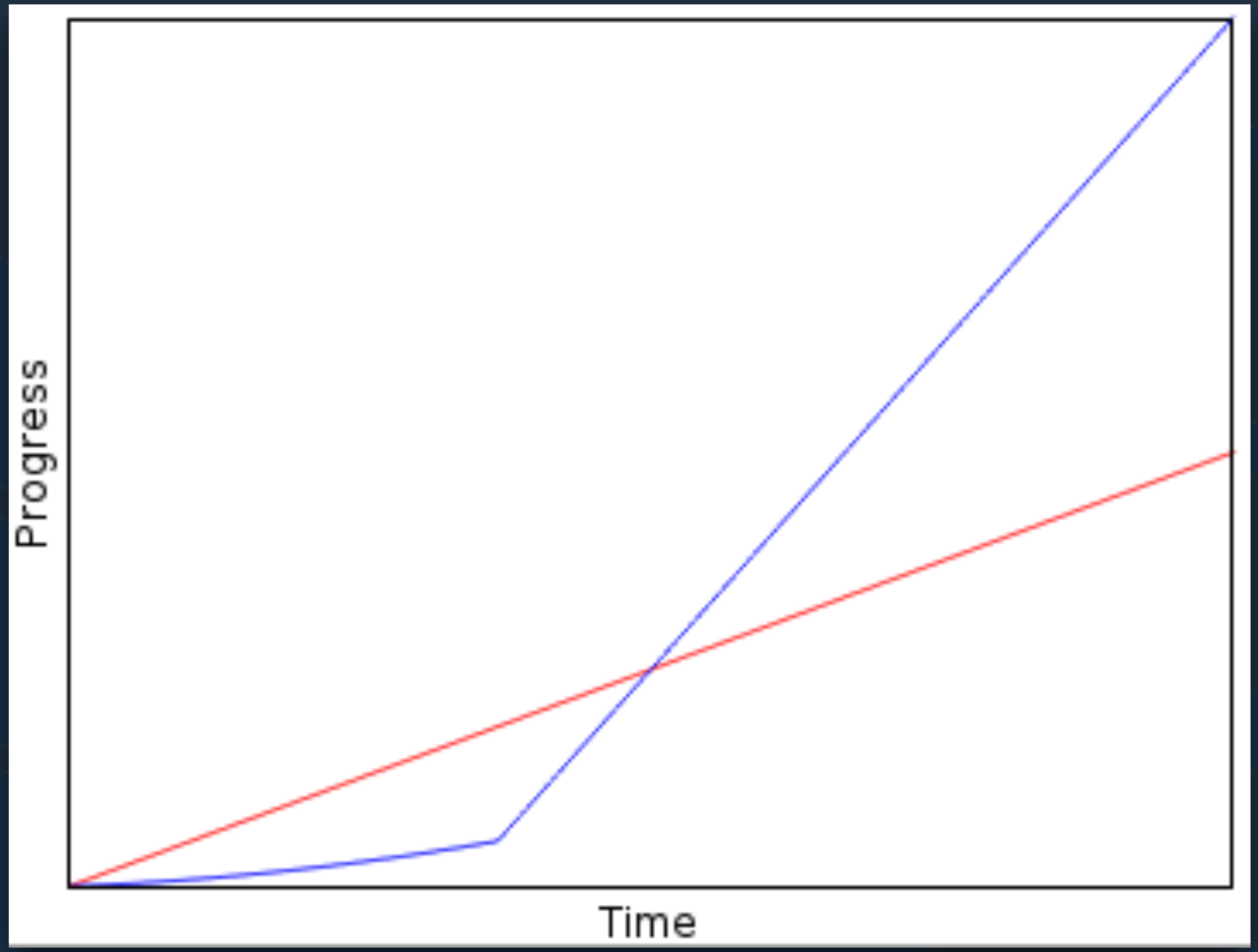
The first part is true.

The second is not!

You get to use a lot of functionality you could never in a lifetime implement yourself.

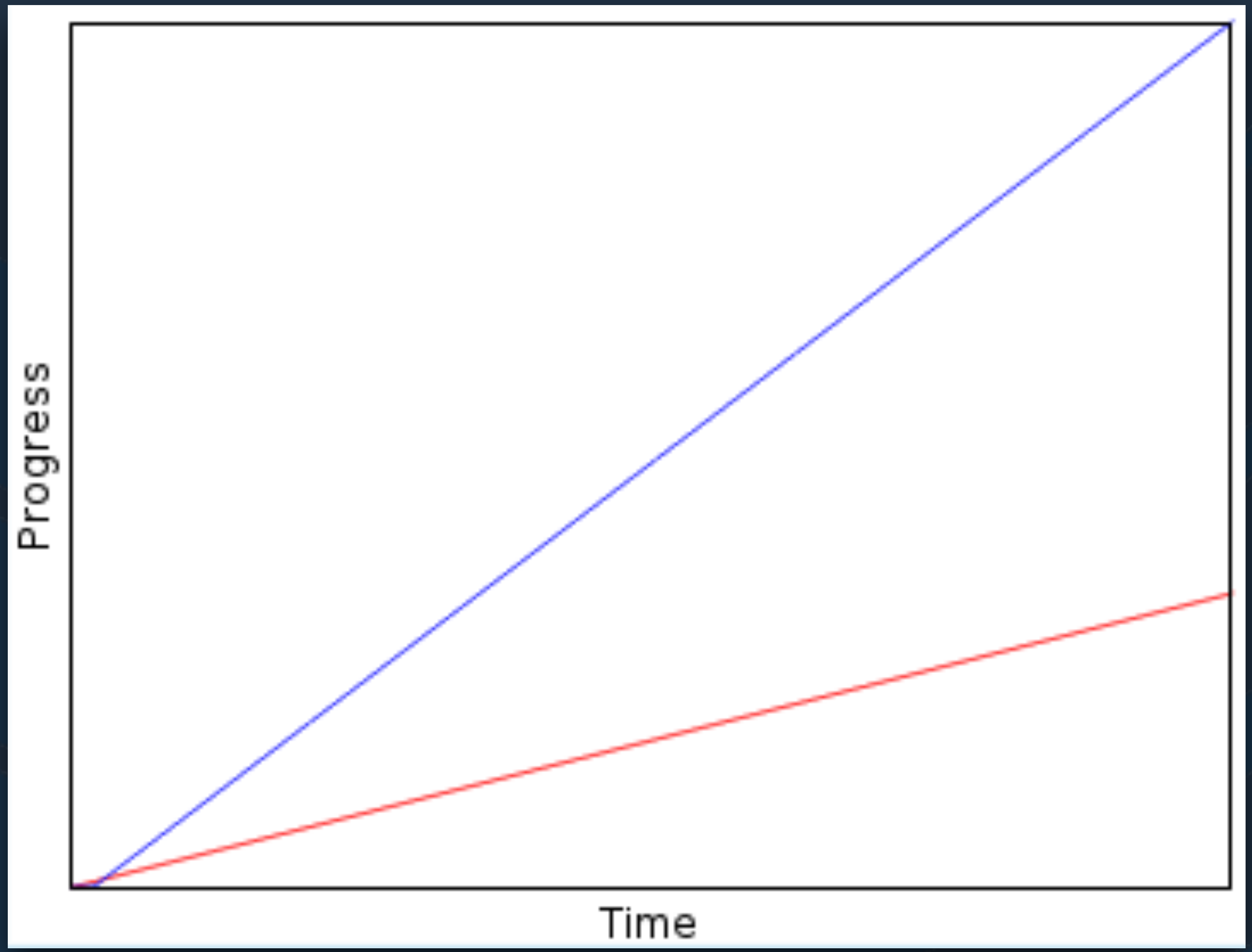
Think of how we use Matlab today!

I’m faster!



Blue: use external libraries
Red: do it yourself

The real picture…



Blue: use external libraries
Red: do it yourself

Common Arguments…

Arguments against using other people's packages:

I want my students to actually understand the methods they are doing. So I let them code things from scratch!

Common Arguments…

Arguments against using other people's packages:

I want my students to actually understand the methods they are doing. So I let them code things from scratch!

Answers:

Yes, there is value to that.

But: if you know quadrature in 2d, why implement it again in 3d?

So let them write a toy code and throw it away after 3 months and do it right based on existing software.

Common Arguments…

Arguments against using other people's packages:

How do I know that that software I'm supposed to use doesn't have bugs? How can I *trust* other people's software?

With my own software, at least I know that I don't have bugs!

Answer 1:

You can't be serious to think that your own software has no bugs!

Common Arguments…

Arguments against using other people's packages:

How do I know that that software I'm supposed to use doesn't have bugs? How can I *trust* other people's software?

With my own software, at least I know that I don't have bugs!

Answer 2:

The packages I will talk about are developed by professionals with a lot of experience

They have extensive testsuites

For example, deal.II runs 3,000+ tests **after every single change**

Bottomline:

When having to implement software for a particular problem, re-use what others have done already

There are many high-quality, open source software libraries for every purpose in scientific computing

Use them:
– You will be far more **productive**
– You will be able to use **state-of-the-art** methods
– You will have far **fewer bugs** in your code

If you are a graduate student:

Use them because you will be able to impress your advisor with quick results!

Roadmap for next lectures:

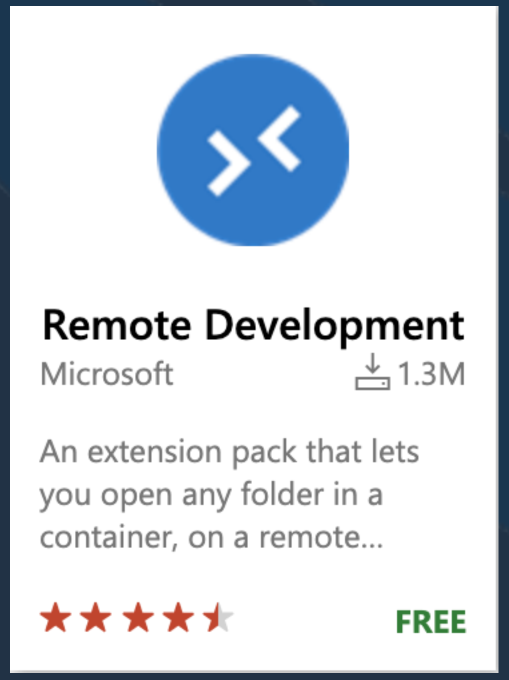
- Version control system (git)
- Modern IDE (VSCode)
- Cross platform build systems (cmake)
- Automatic formatting (clang-format)
- Test driven development (google test, deal.II testing framework)
- Inline documentation (doxygen)
- External visualisation tools (paraview)

Version control systems

- We'll use GitHub classrooms:
  - Understand how GitHub classrooms work
    https://github.com/jfiksel/github-classroom-for-students
    (the guide shows how to install R-Studio. We won't need that. Don't do that part.)
    After setting up git…
  - Accept the first assignment:
    https://classroom.github.com/a/SxO9Wq2M
  - If you are new to git, follow the tutorial at
    http://learngitbranching.js.org/
    to learn how to use git. I'll give a very short overview today.

Setting up VSCode

- Download and install **Docker**: https://www.docker.com/products/docker-desktop
  - Read some doc: https://www.docker.com/get-started
- Download and install: https://code.visualstudio.com/download
  - Read some doc: https://code.visualstudio.com/docs
  - Install the following extension:



Open the assignment repo

- Open the directory containing the assignment. The directory contains a hidden folder, called “**.devcontainer**”
- VSCode should ask you if you want to run the folder in the container. Say yes.
- VSCode will now download a docker image. The first time around, this will take some time.