Introduction to  git

Anna Ivagnes, Federico Pichi, Gianluigi Rozza

October 1, 2025

Slides inspired by previous courses given by Marco Tezzele and Nicola Demo

The Multiple Roles of a Computational Scientist

Computational Scientists are expected to be:

- Physicist
- Mathematician/Numerical Analyst
- Software Developer??

Software Engineering is vital; need for

- Understandable, well documented code
- Performant code
- Verifiable code

The solution: a proper version control system

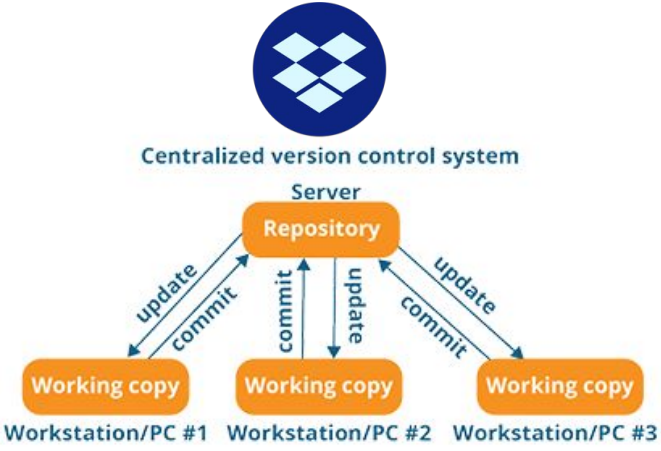
Codes evolve over time

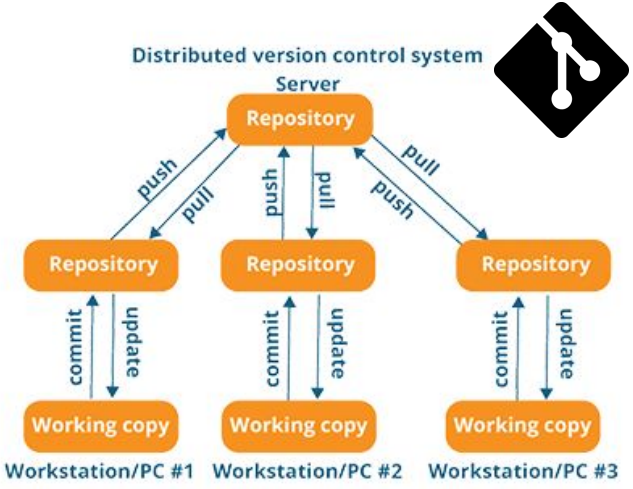
- Sometimes bugs creep in (by you or others)
- Sometimes the old way was right (or better)

Working with others is easier

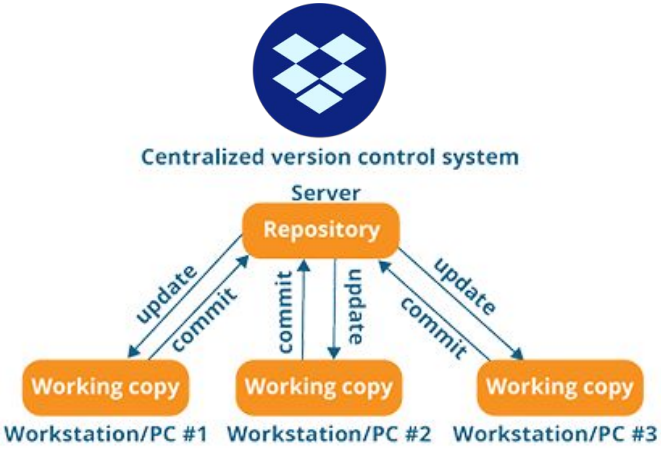
- Contribution transparency
- Working in teams (parallel development)
- Reproducibility
- You can synchronize all your files on your devices
- You can synchronize only what is necessary

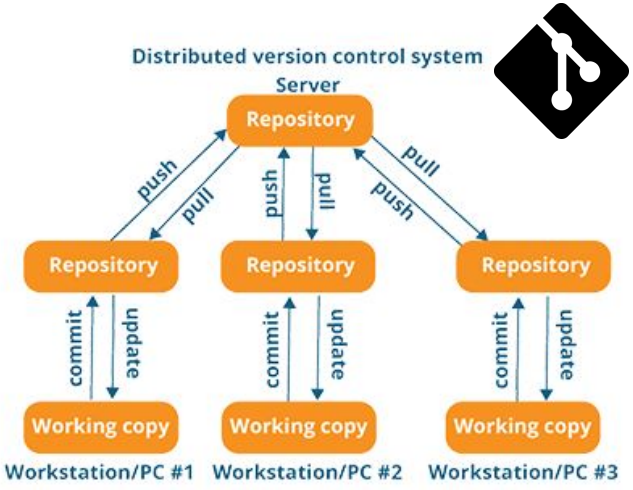
Version control systems: dropbox vs git





Version control systems: dropbox vs git





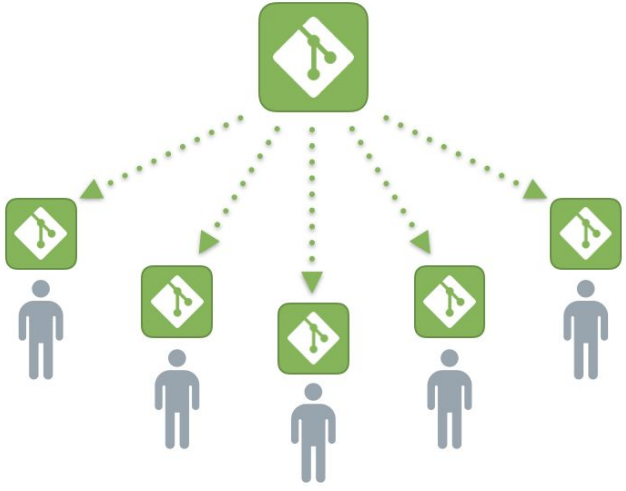
- Traces history only up to 30-180 days
- Create “copies” of the files
- Cannot go back

- Traces all history
- Has memory of every modified line
- Can always go back

git

Pros

- Local copy for each user
- Every user can edit their own local repository without affecting others
- The online repository is public and contains only working code
- A high number of users can contribute to the same software



Installing git

https://git-scm.com/download/

Open a terminal and check with **`git --version`** if it is already installed!

MacOS

```
brew install git
```

Linux

Different commands depending on the distro.

For Debian/Ubuntu:
`apt-get install git`

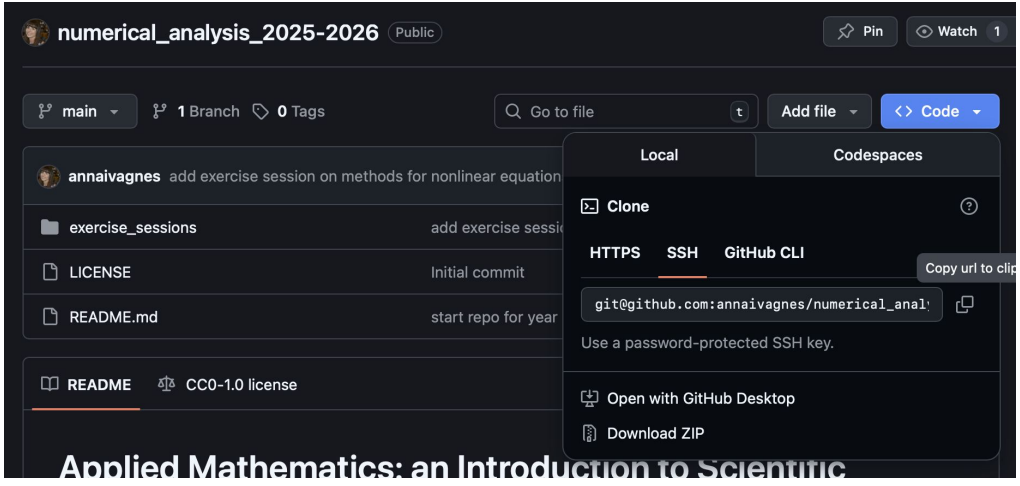
Windows

- Download the installer from the website
- Execute the file `.exe` and follow the steps
- Launch Git Bash (should do automatically)
- type `git --version` to check

For the course

After installation, just go to the folder you want on your PC, and type:

```
git clone git@github.com:annaivagnes/numerical_analysis_2025-2026.git
```



This will create a local copy of the project on your laptop.

Wonderful, but how does it work?

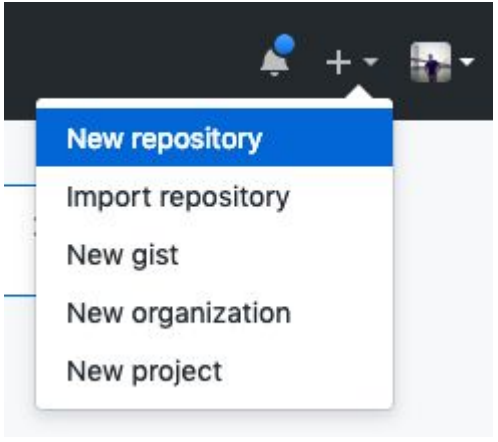
(not necessary for the course)

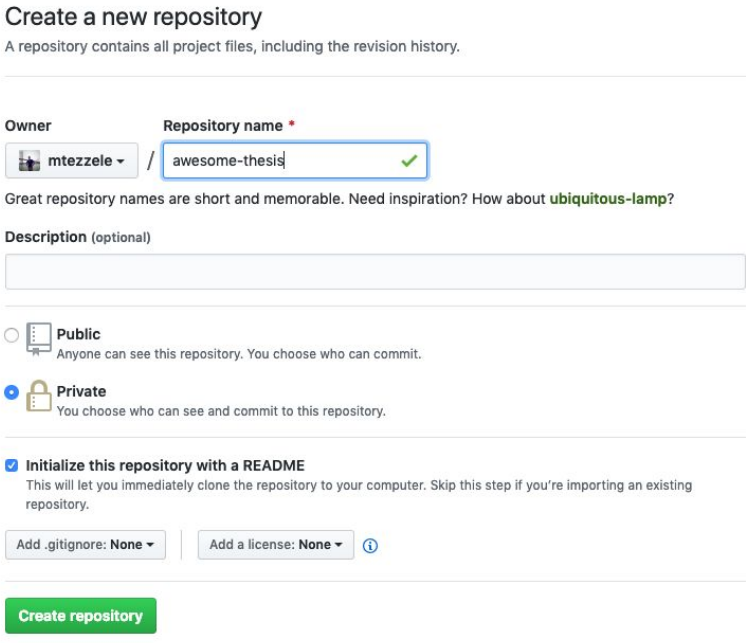
Let's get our hands dirty:

1. How to create a repo on GitHub
2. Copying the fresh repo
3. Adding the first file

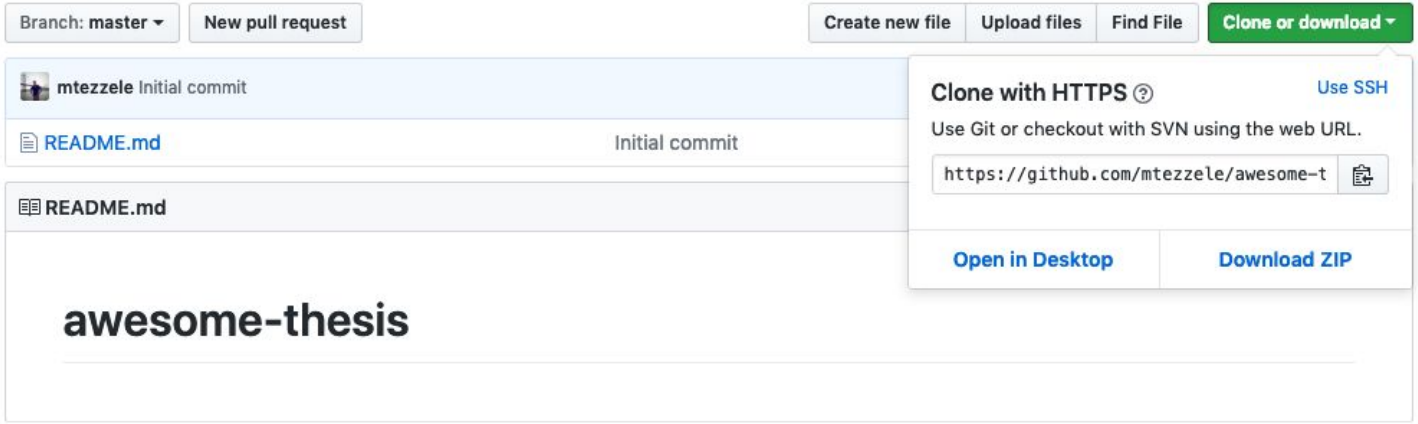
GitHub

1. Create a project





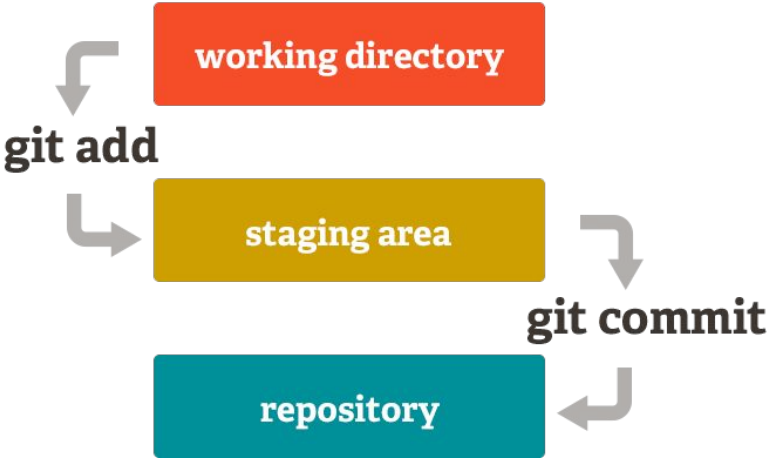
2. Clone the project




```
dhcp-128-215:Git marcotezzele$ git clone https://github.com/mtezzele/awesome-thesis.git
Cloning into 'awesome-thesis'...
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (3/3), done.
```

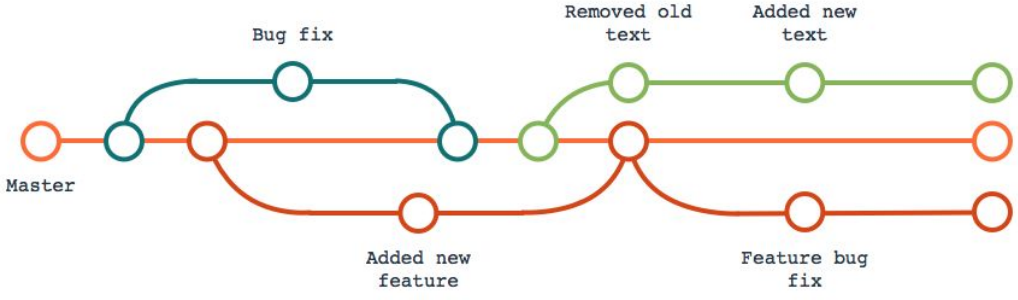
3. Add a file

- cd awesome-thesis
- touch thesis.tex
- git add thesis.tex
- git commit -m “main file”
- git push origin main



I branch therefore I am

- 1 feature = 1 branch
- master/main branch is for <u>working code only</u>
- You can have more than one branch concurrently
- To see all your branches: **git branch**
- To create a new branch from master: **git checkout -b my_new_branch**
- To move between branches: **git checkout branch_name**



Pull, fetch, merge, rebase: keep it synchronized

(may be useful for the course)

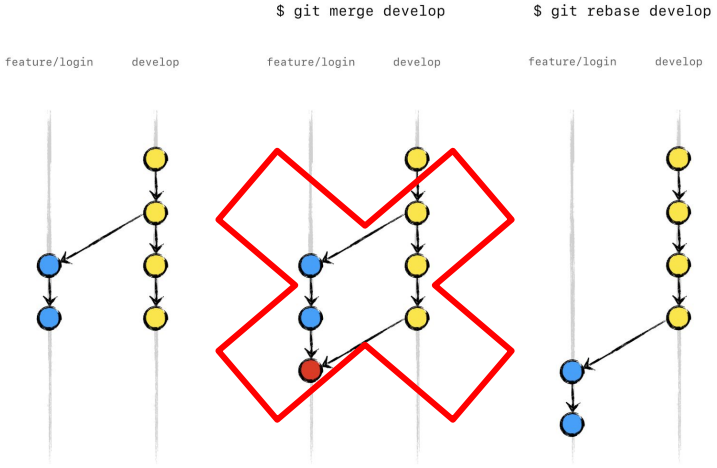
Keep a linear history:

- **git fetch origin**

  Download the last updates WITHOUT merging them with your local changes

- **git rebase -i origin/main**

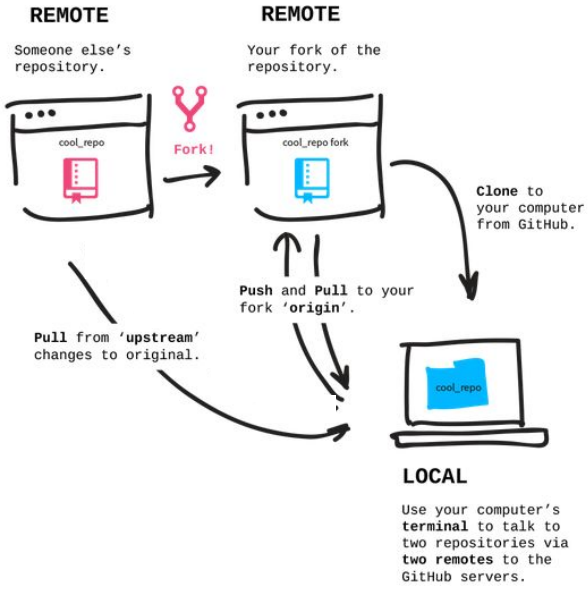
  Perform a rebase of the current branch wrt the main branch on origin



Utilize GitHub

(may be useful for the course)

- Fork public repositories
- Pull request (PR) to contribute to well organized projects
- Add a remote to your local repo:
  git remote add remote_name remote_url
- To see all your remotes:
  git remote -v



Two options for the course

Passive user

- `git clone git@github.com:annaivagnes/numerical_analysis_2025-2026.git`
- Sometimes (for updating your copy with recent lectures):
  - `git fetch origin`
  - `git rebase origin/main`

Active user

- Create your fork (on Github)
- `git clone git@github.com:yourname/numerical_analysis_2025-2026.git`
- `git remote add upstream git@github.com:annaivagnes/numerical_analysis_2025-2026.git`
- Do your modifications:
  - `git add solution_to_exercise.ipynb`
  - `git commit -m “upload exercise 1”`
- Sometimes (for updating your copy with recent lectures):
  - `git fetch upstream` (or how you called it)
  - `git rebase upstream/main`

Get your hands dirty

The only way to learn git is to bang your head against it.

GitHub cheat sheet:
education.github.com/git-cheat-sheet-education.pdf

Hold on, it's worth!