Introduction to numerical astrophysics and python

Kristina Kislyakova
University of Vienna, 08.10.2025

Schedule

- 08.10.2025 Lecture #1 (Introduction)
- 15.10.2025 Lecture #2 (Visualization)
- 22.10.2025 Lecture #3 (Root finding)
- 29.10.2025 Lecture #4 (Interpolation and approximation)
- 05.11.2025 Lecture #5 (Derivation and integration)
- 19.11.2025 Lecture #6 (ODEs 1)
- 26.11.2025 Lecture #7 (ODEs 2)
- 03.12.2025 Lecture #8 (Fitting models to data)
- 10.12.2025 Lecture #9 (MCMC + sampling)
- 17.12.2025 Lecture #10 (DSMC, small lecture + **no exercise**)
- **Winter Break**
- 07.01.2026 Lecture #11 (FFT)
- 14.01.2026 Lecture #12 (Parallelisation)
- 21.01.2026 Lecture #13 (Machine learning)

Schedule

- 08.10.2025 Lecture #1 (Introduction)
- 15.10.2025 Lecture #2 (Visualization)
- 22.10.2025 Lecture #3 (Root finding)
- 29.10.2025 Lecture #4 (Interpolation and approximation)
- 05.11.2025 Lecture #5 (Derivation and integration)
- 19.11.2025 Lecture #6 (ODEs 1)
- 26.11.2025 Lecture #7 (ODEs 2)
- 03.12.2025 Lecture #8 (Fitting models to data)
- 10.12.2025 Lecture #9 (MCMC + sampling)
- 17.12.2025 Lecture #10 (DSMC, small lecture + **no exercise**)
- **Winter Break**
- 07.01.2026 Lecture #11 (FFT)
- 14.01.2026 Lecture #12 (Parallelisation)
- 21.01.2026 Lecture #13 (Machine learning)

EXAMS:

12.11.2025 Exam 1 (1st opportunity)
18.11.2025 Exam 1 (2nd opportunity)

28.01.2026 Exam 2 (1st opportunity)
04.02.2026 Exam 2 (2nd opportunity)

The students can select **only one date for each exam.** Registration for the exams will be organised on Moodle and announced in advance.

Schedule

- 08.10.2025 Lecture #1 (Introduction)
- 15.10.2025 Lecture #2 (Visualization)
- 22.10.2025 Lecture #3 (Root finding)
- 29.10.2025 Lecture #4 (Interpolation and approximation)
- 05.11.2025 Lecture #5 (Derivation and integration)
- 19.11.2025 Lecture #6 (ODEs 1)
- 26.11.2025 Lecture #7 (ODEs 2)
- 03.12.2025 Lecture #8 (Fitting models to data)
- 10.12.2025 Lecture #9 (MCMC + sampling)
- 17.12.2025 Lecture #10 (DSMC, small lecture + **no exercise**)
- **Winter Break**
- 07.01.2026 Lecture #11 (FFT)
- 14.01.2026 Lecture #12 (Parallelisation)
- 21.01.2026 Lecture #13 (Machine learning)

EXAMS:

12.11.2025 Exam 1 (1st opportunity)
18.11.2025 Exam 1 (2nd opportunity)

28.01.2026 Exam 2 (1st opportunity)
04.02.2026 Exam 2 (2nd opportunity)

The students can select **only one date for each exam.** Registration for the exams will be organised on Moodle and announced in advance.

Mini-projects:

The deadline for mini-projects is **February, 15th.** Mini-projects and exercises should be uploaded to Github.

Exercises

- Exercises take place in Seminarraum 1 (SE1) every Tuesday (see schedule on Moodle and u:find)
- The topic follows the previous lecture
- The next exercise becomes available on Wednesday after the lecture

EXERCISE GROUPS:

12.11.2025 Group 1 (Alina Böcker)
18.11.2025 Group 2 (Kristina Kislyakova)

Registration:

Registration for exercise groups opens after this lecture on October 8th at 12:00 and ends on October 10th at 23:59.

Exercises

- Exercises take place in Seminarraum 1 (SE1) every Tuesday (see schedule on Moodle and u:find)
- The topic follows the previous lecture
- The next exercise becomes available on Wednesday after the lecture

Swapping exercise groups: if you fail to register for the exercise group you wish to visit, please find a student from the other group willing to swap a place with you. Then write me an email with both of you in cc, and I will change the group assignment for both of you

EXERCISE GROUPS:

12.11.2025	Group 1 (Alina Böcker)
18.11.2025	Group 2 (Kristina Kislyakova)

Registration:

Registration for exercise groups opens after this lecture on October 8th at 12:00 and ends on October 10th at 23:59.

This lecture

- Basic concepts in numerical astronomy
- A few words about different languages
- Useful software
- Introduction to python
- Introduction to Git/Github

Concepts

Why to use numerical methods at all?

Bessel functions of the first and second kind

1) $\int xJ_0(x)\,dx = xJ_1(x)$

2) $\int x^2J_0(x)\,dx = x^2J_1(x) + x\,J_0(x) - \int J_0(x)\,dx$

3) $\int x^mJ_0(x)\,dx = x^mJ_1(x) + (m-1)x^{m-1}J_0(x) - (m-1)^2\int x^{m-2}J_0(x)\,dx$

4) $\int \frac{J_0(x)}{x^2}\,dx = J_1(x) - \frac{J_0(x)}{x} - \int J_0(x)\,dx$

5) $\int \frac{J_0(x)}{x^m}\,dx = \frac{J_1(x)}{(m-1)^2x^{m-2}} - \frac{J_0(x)}{(m-1)x^{m-1}} - \frac{1}{(m-1)^2}\int \frac{J_0(x)}{x^{m-2}}\,dx$

6) $\int J_1(x)\,dx = -J_0(x)$

7) $\int xJ_1(x)\,dx = -xJ_0(x) + \int J_0(x)\,dx$

8) $\int x^mJ_1(x)\,dx = -x^mJ_0(x) + m\int x^{m-1}J_0(x)\,dx$

9) $\int \frac{J_1(x)}{x}\,dx = -J_1(x) + \int J_0(x)\,dx$

10) $\int \frac{J_1(x)}{x^m}\,dx = -\frac{J_1(x)}{mx^{m-1}} + \frac{1}{m}\int \frac{J_0(x)}{x^{m-1}}\,dx$

11) $\int x^{\nu}J_{\nu-1}(x)\,dx = x^{\nu}J_{\nu}(x)$

12) $\int x^{-\nu}J_{\nu+1}(x)\,dx = -x^{-\nu}J_{\nu}(x)$

13) $\int x^mJ_{\nu}(x)\,dx = -x^mJ_{\nu-1}(x) + (m+\nu-1)\int x^{m-1}J_{\nu-1}(x)\,dx$

14) $\int xJ_{\nu}(\alpha x)J_{\nu}(\beta x)\,dx = \frac{x\left[\alpha J_{\nu}(\beta x)J_{\nu}'(\alpha x) - \beta J_{\nu}(\alpha x)J_{\nu}'(\beta x)\right]}{\beta^2 - \alpha^2}$

15) $\int xJ_{\nu}^2(\alpha x) = \frac{x^2}{2}\left[J_{\nu}'(\alpha x)\right]^2 + \frac{x^2}{2}\left(1 - \frac{\nu^2}{\alpha^2x^2}\right)\left[J_{\nu}(\alpha x)\right]^2$

Why to use numerical methods at all?

Bessel functions of the first and second kind

Analytical methods can be too complex or too time consuming

1) $\int xJ_0(x)\,dx = xJ_1(x)$

2) $\int x^2 J_0(x)\,dx = x^2 J_1(x) + x\,J_0(x) - \int J_0(x)\,dx$

3) $\int x^m J_0(x)\,dx = x^m J_1(x) + (m-1)x^{m-1} J_0(x) - (m-1)^2 \int x^{m-2} J_0(x)\,dx$

4) $\int \frac{J_0(x)}{x^2}\,dx = J_1(x) - \frac{J_0(x)}{x} - \int J_0(x)\,dx$

5) $\int \frac{J_0(x)}{x^m}\,dx = \frac{J_1(x)}{(m-1)^2 x^{m-2}} - \frac{J_0(x)}{(m-1)x^{m-1}} - \frac{1}{(m-1)^2}\int \frac{J_0(x)}{x^{m-2}}\,dx$

6) $\int J_1(x)\,dx = -J_0(x)$

7) $\int xJ_1(x)\,dx = -xJ_0(x) + \int J_0(x)\,dx$

8) $\int x^m J_1(x)\,dx = -x^m J_0(x) + m\int x^{m-1} J_0(x)\,dx$

9) $\int \frac{J_1(x)}{x}\,dx = -J_1(x) + \int J_0(x)\,dx$

10) $\int \frac{J_1(x)}{x^m}\,dx = -\frac{J_1(x)}{mx^{m-1}} + \frac{1}{m}\int \frac{J_0(x)}{x^{m-1}}\,dx$

11) $\int x^\nu J_{\nu-1}(x)\,dx = x^\nu J_\nu(x)$

12) $\int x^{-\nu} J_{\nu+1}(x)\,dx = -x^{-\nu} J_\nu(x)$

13) $\int x^m J_\nu(x)\,dx = -x^m J_{\nu-1}(x) + (m+\nu-1)\int x^{m-1} J_{\nu-1}(x)\,dx$

14) $\int xJ_\nu(\alpha x)J_\nu(\beta x)\,dx = \frac{x\left[\alpha J_\nu(\beta x)J_\nu'(\alpha x) - \beta J_\nu(\alpha x)J_\nu'(\beta x)\right]}{\beta^2 - \alpha^2}$

15) $\int xJ_\nu^2(\alpha x) = \frac{x^2}{2}\left[J_\nu'(\alpha x)\right]^2 + \frac{x^2}{2}\left(1 - \frac{\nu^2}{\alpha^2 x^2}\right)\left[J_\nu(\alpha x)\right]^2$

Important concepts

ODEs and PDEs

$$\frac{df}{dt} = t^2 + 1 \qquad \frac{\partial \rho}{\partial t} + \frac{\partial(\rho v)}{\partial r} = 0$$

Ordinary and Partial Differential Equations are very important in most large numerical tasks. They usually require very different methods for solving.

Important concepts

ODEs and PDEs

$$\frac{df}{dt} = t^2 + 1 \qquad \frac{\partial \rho}{\partial t} + \frac{\partial(\rho v)}{\partial r} = 0$$

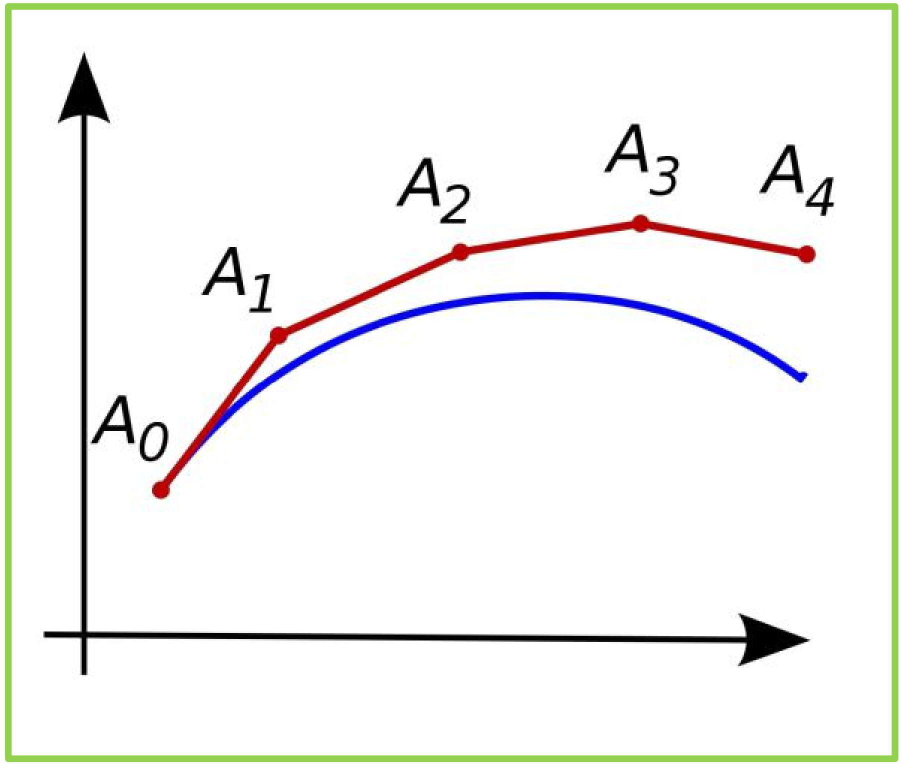
Ordinary and Partial Differential Equations are very important in most large numerical tasks. They usually require very different methods for solving.

We will discuss the solution for ODEs in the lectures dedicated to them. One can often use similar or the same solvers for PDEs.

Important concepts

Initial conditions

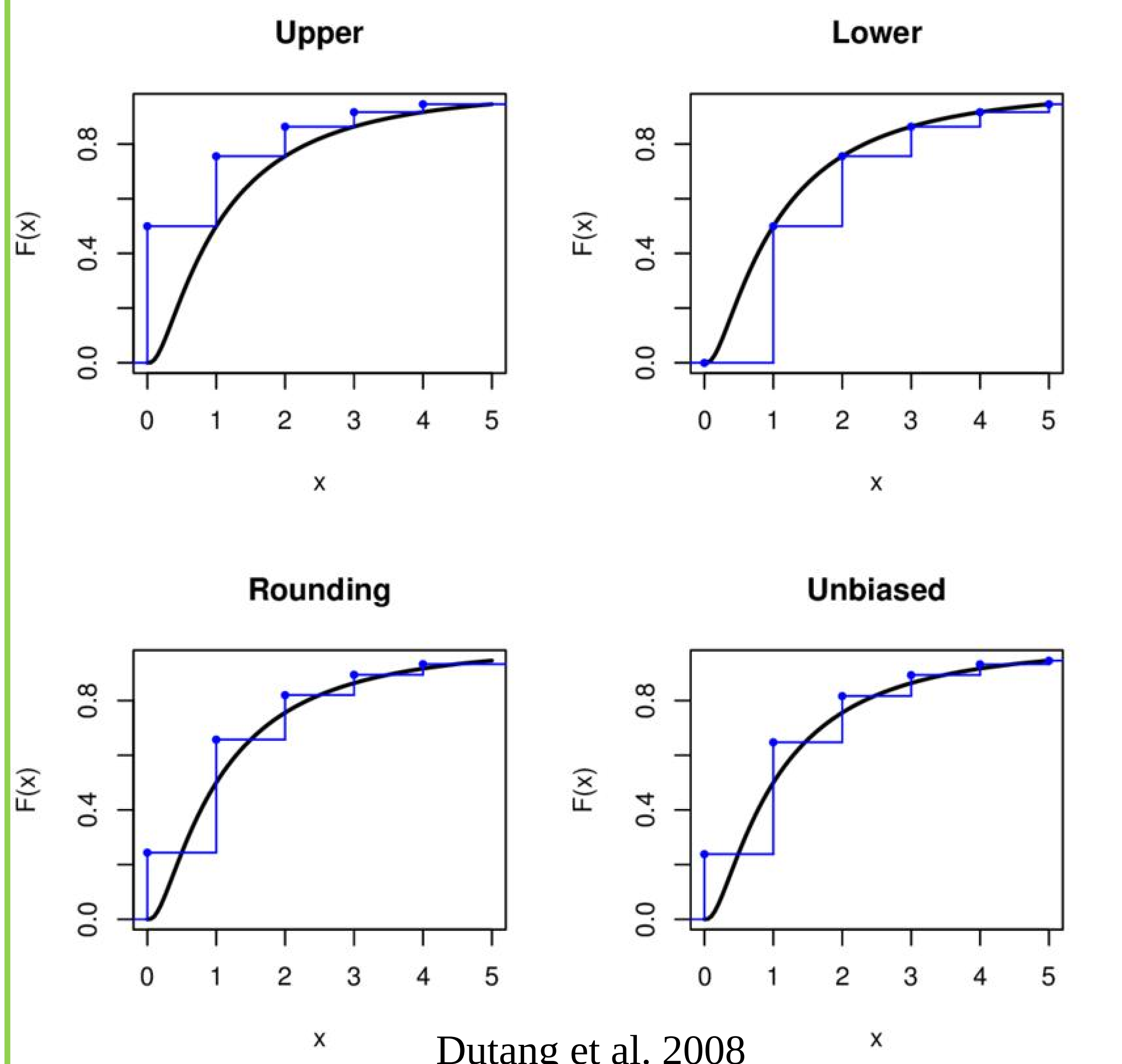
In many methods, before iteration can start, a guess of the solution is necessary. Quite often this guess will need to be close to the true solution.



Important concepts

Discretisation:

the process of turning a continuous problem into a discrete problem. For example, you might break down space into smaller regions, often called ‘cells’.



Dutang et al. 2008

Important concepts

Errors, for instance:

- round-off error due to finite accuracy of computers

$$\boxed{(1/3 = 0.33333....)}$$

- discretization error because discretization is only approximate

$$\boxed{\frac{\partial y}{\partial x} \approx \frac{y_2 - y_1}{x_2 - x_1}}$$

- truncation error

$$\boxed{e^x = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \frac{x^4}{4!} + \cdots}$$

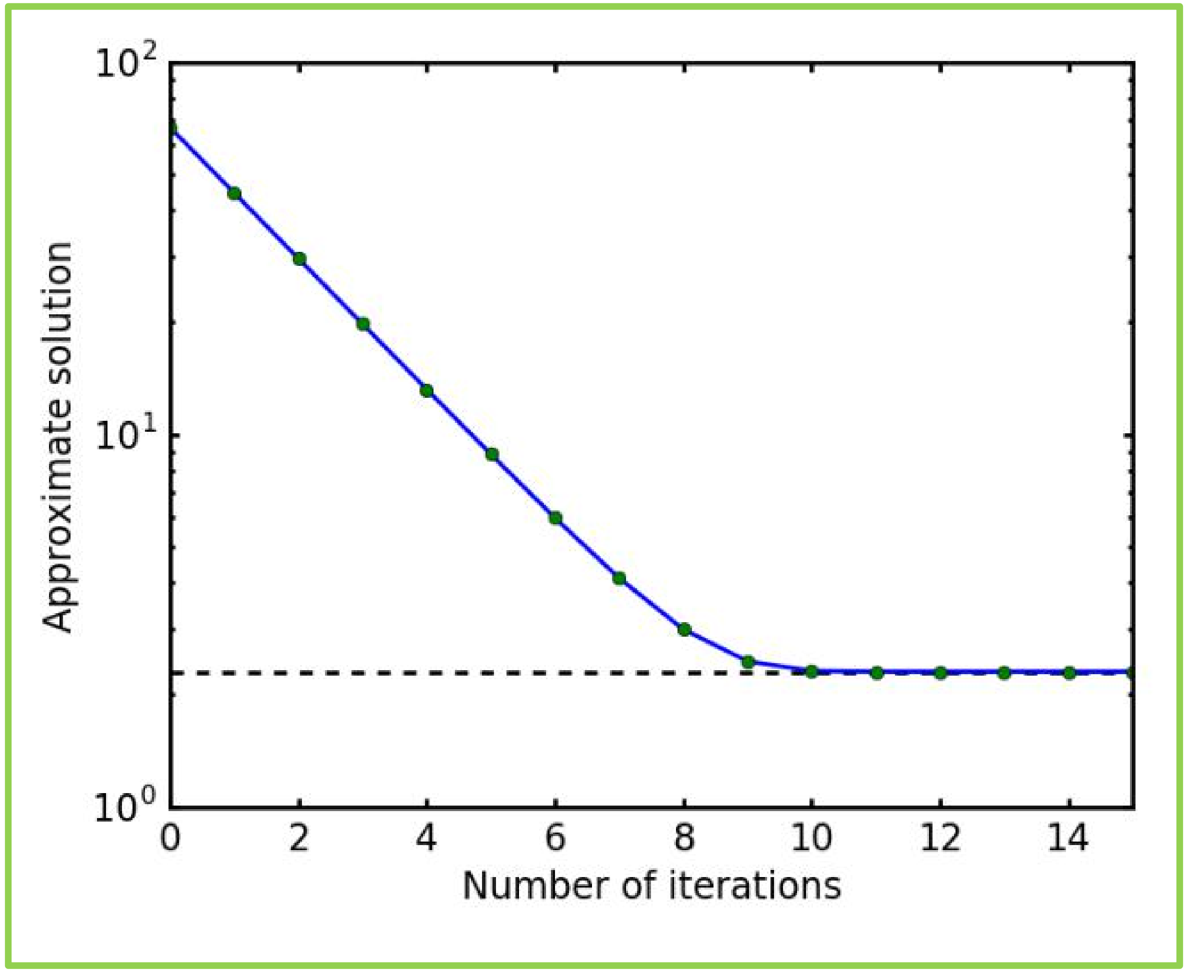
Important concepts

Convergence

You should stop iterating once the approximation stops to change with each iteration.

Steady state

The system has reached the final result and stopped changing

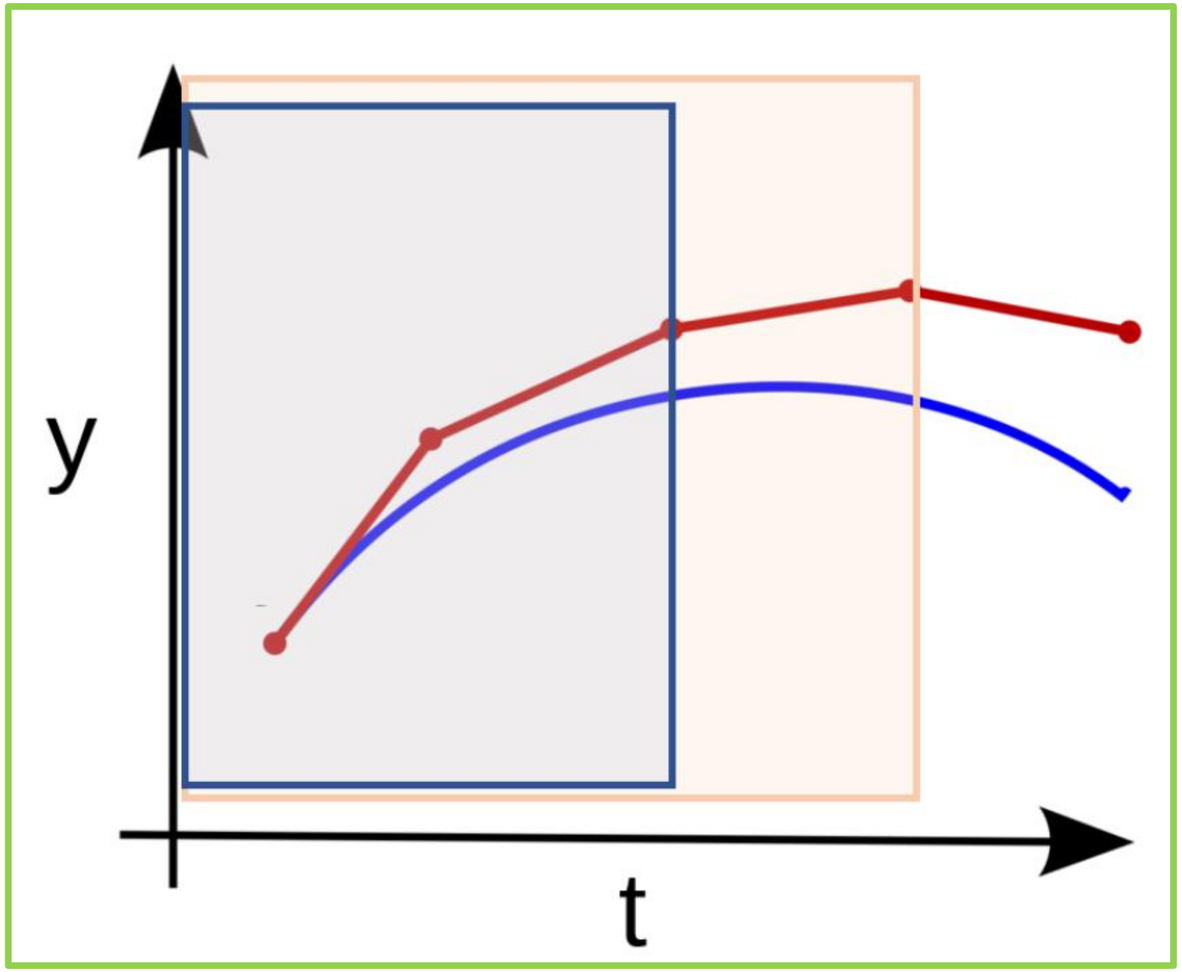


Important concepts

Explicit and implicit methods

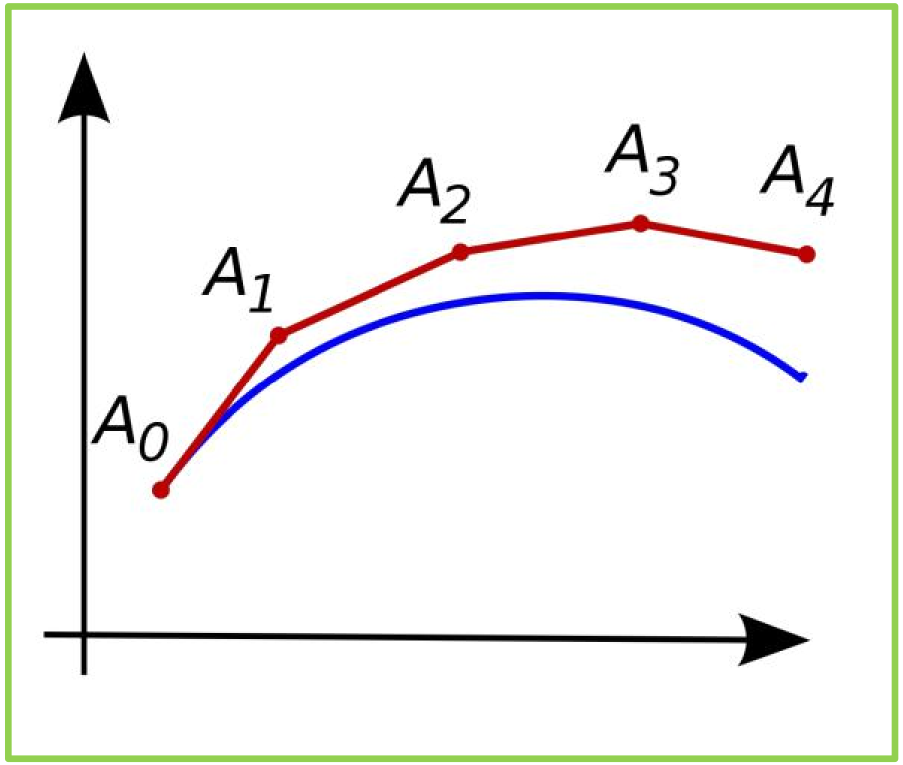
The implicit method uses the slope of the next unknown time-step, and therefore must be solved numerically.

The explicit method uses the slope at a known time-step to find the slope at the next time-step, so it can be determined directly



Important concepts

Iteration is the process of repeating a calculation many times. Most numerical methods use iteration in some form.



Important concepts

Initial value problem

Often you need a very educated guess about your solution *in advance*

INITIAL CONDITIONS: $y_0 = y(t_0)$

PHYSICAL MODEL: $\dfrac{dy}{dt} = f(y, t)$

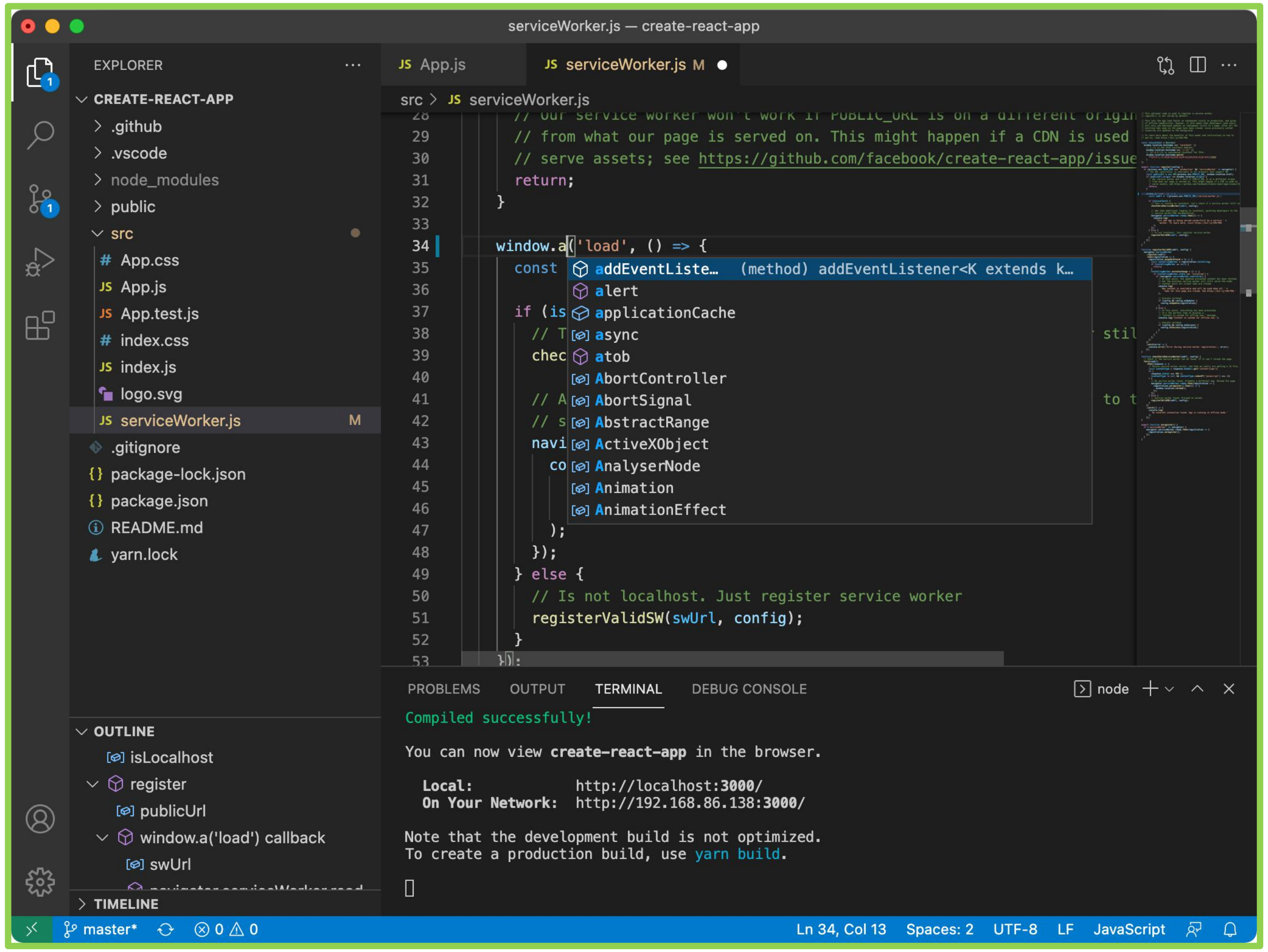
KNOWN

DESIRED: $y(t)$

Useful software / tools

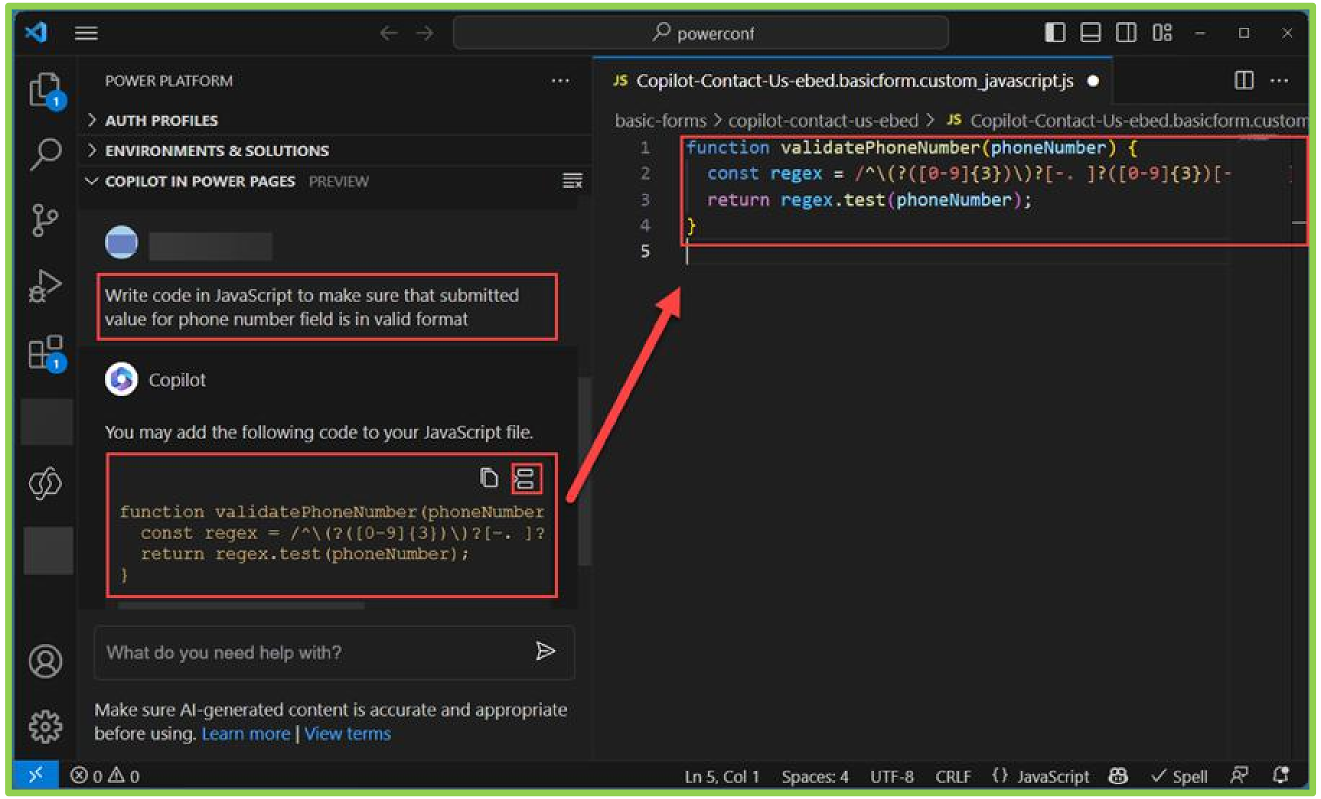
Visual studio code

- Free to use
- Plugins. So many plugins!
- Integratable with github, co-pilot, etc.
- Support of multiple languages



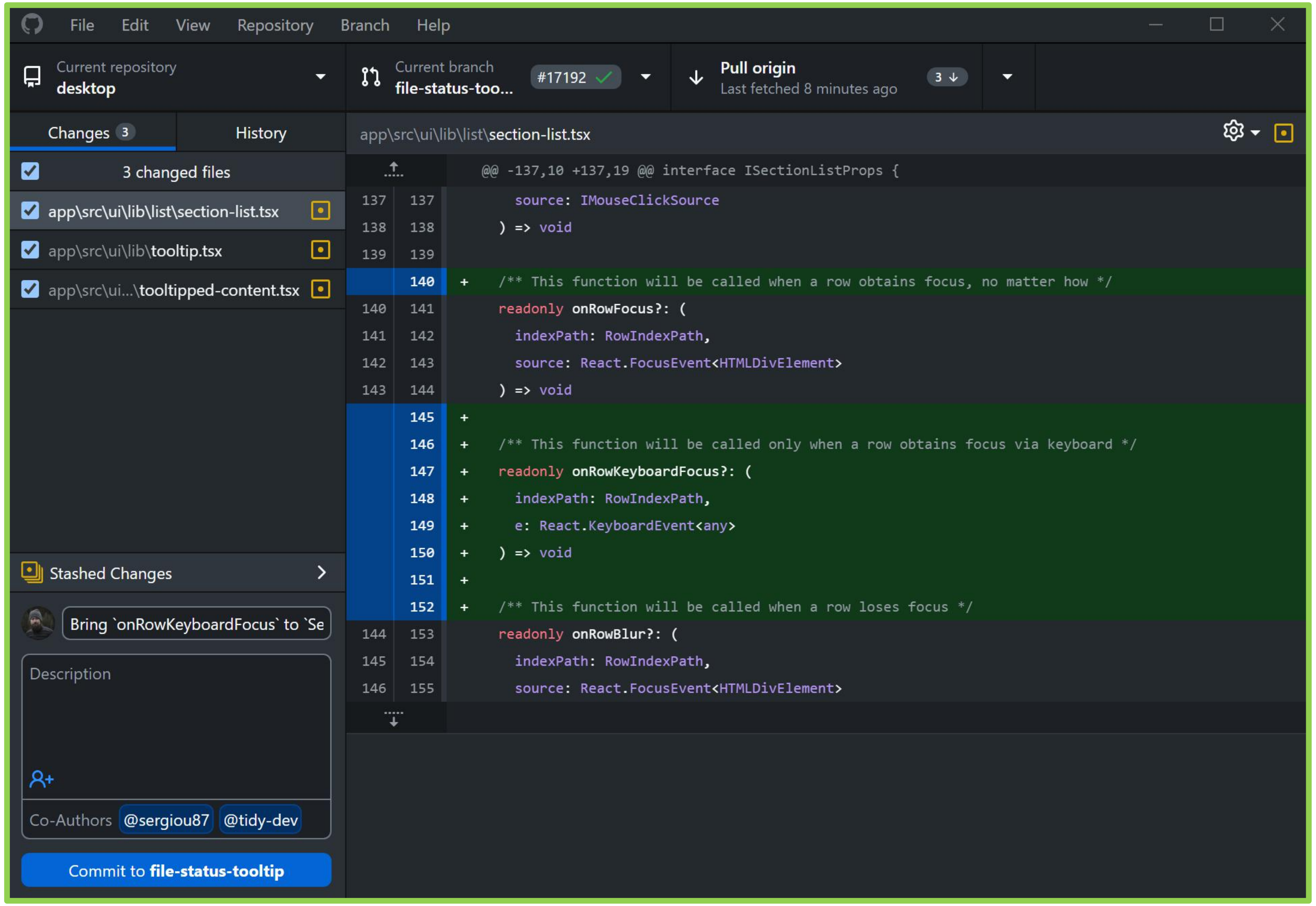
Visual studio code & Co-pilot

- Co-P is not free to use, but there is an (old, limited) version available
- Integratable with VSC
- Support of multiple languages
- Existing solutions at your fingertips



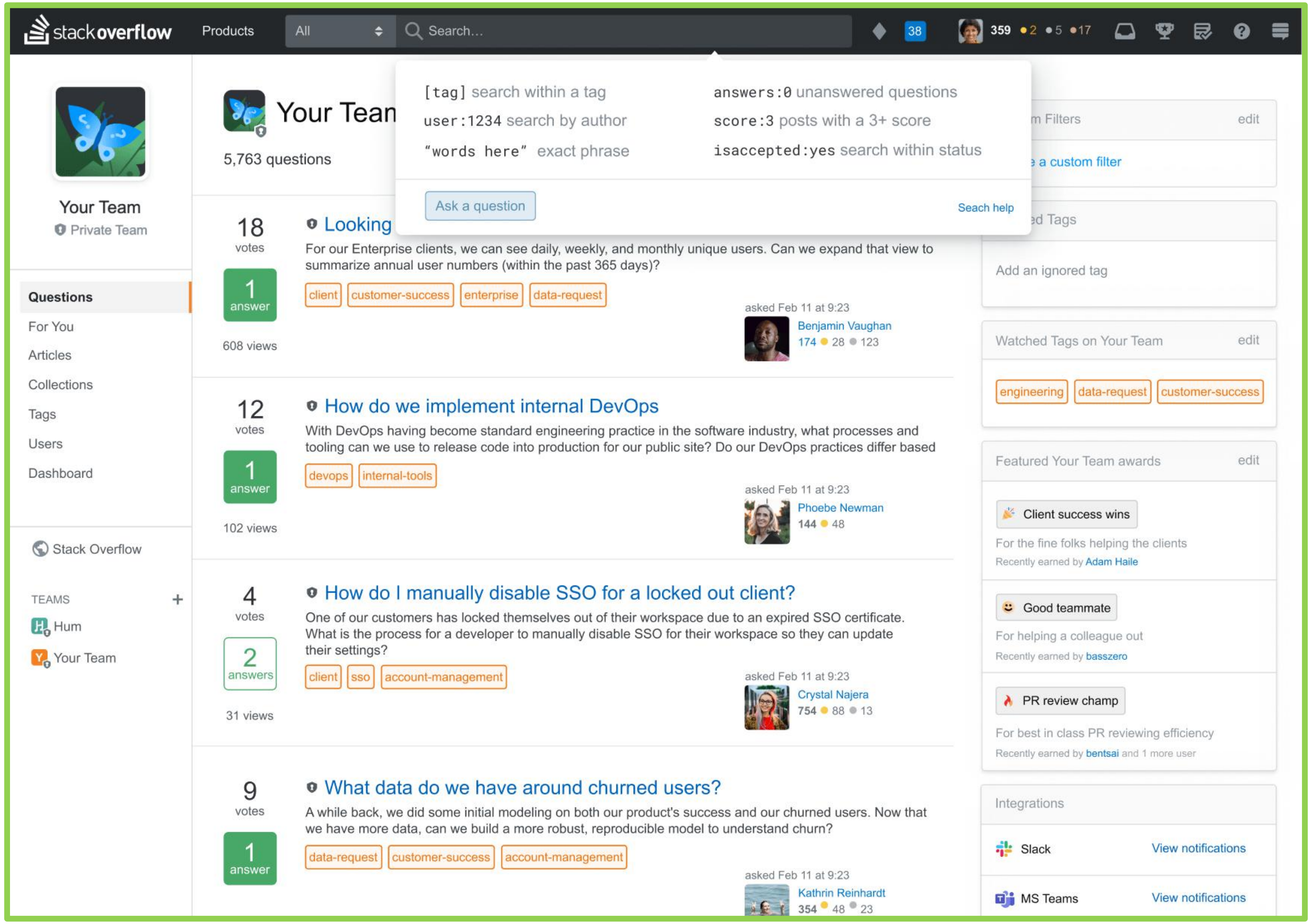
Github

- Free to use
- Version control
- Code available everywhere
- Easy to share



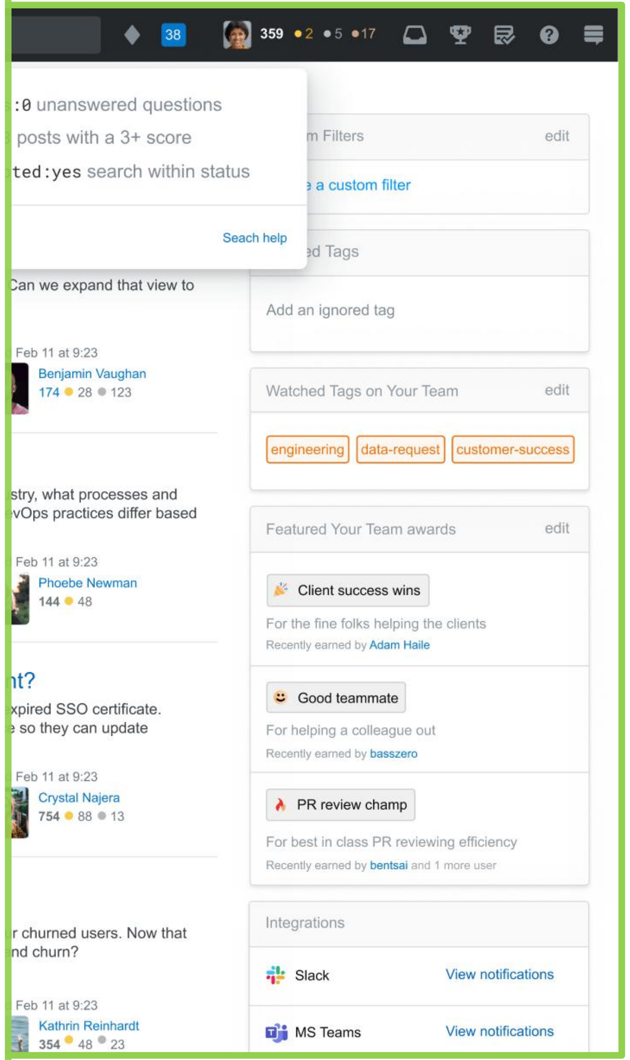
Stack Overflow

- Free to use
- The power of the hive mind



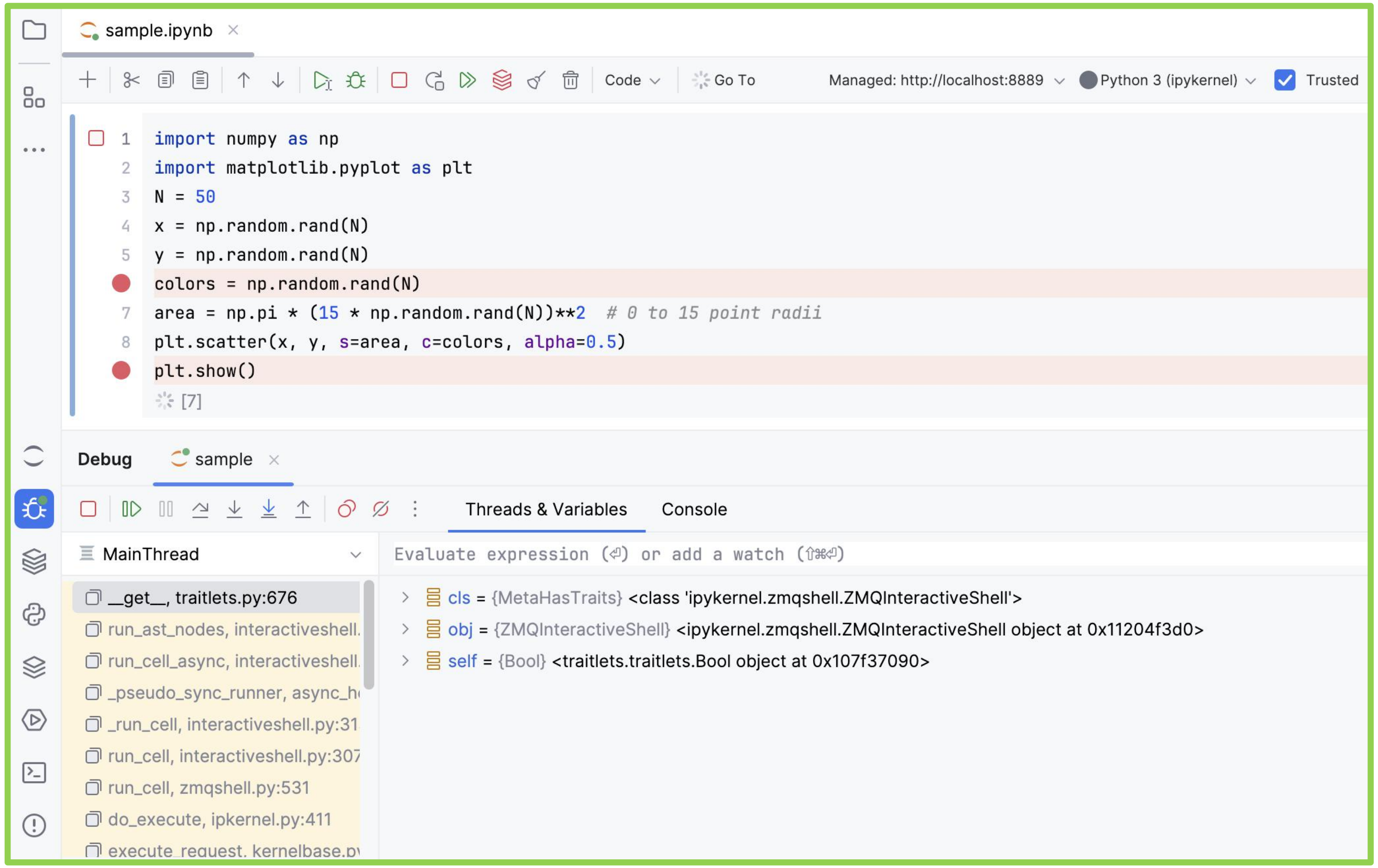
Stack Overflow

- Free to use
- The power of the hive mind



Python-specific: Jupyter notebook

- Includes debugging tools
- Generally easy to use



A quick note on different languages

~Two big groups:

- **C, C++, Fortran:**

  - Compilation needed
  - Advantages: fast!!! Should be used for large calculations.
  - Disadvantages: can be slower to program with,no built-in plotting procedures

- **Python, IDL, R:**

  - Interpreted → no compilation
  - Advantages: often very convenient to program with built-in procedures; plotting is easy
  - Disadvantages: slow, really really slow

Examples

Lecture1_MainExample.py
Lecture1_MainExample_Classes.py
Intro_to_python-JupyterNotebook.pdf
Intro_to_python.ipynb
Python syntax hints.ipynb

A great lecture on Git and Github:
https://people.irisa.fr/Anthony.Baire/git/git-for-beginners-handout.pdf

Also see the files on Github uploaded to Moodle