AI coding Session 1

Vitaly Meursault and Minchul Shin

2025-06-16

Note: This document was prepared for an internal LLM workshop at the Federal Reserve Bank of Philadelphia. Because AI-assisted coding tools improve rapidly, some content may be outdated.

Disclaimer: The views expressed herein are solely those of the authors and do not necessarily reflect the views of the Federal Reserve Bank of Philadelphia or the Federal Reserve System. Nothing in the material should be construed as an endorsement of any organization or its products or services. Any errors or omissions are the responsibility of the authors. No statements here should be treated as legal advice. This is informational without any representations or assurances on applicability to others.

Introduction

Today, we define what an LLM agent is and how it works. Now, we'll dive deeply into one of its primary applications: AI-assistant coding.

- **This session**: We'll explore how LLM+tool(s) work within a coding environment.
- **Next session**: We'll examine how more sophisticated LLM coding agents operate.

As an economist, we code extensively in our work:

- Data analysis, quantitative modeling, and econometric analysis using tools like R, Python, Matlab, STATA, etc.
- Writing drafts using LaTeX.
- Creating slide decks with LaTeX (e.g., beamer)

In this session, I will focus on using **VS code** + **Github Copilot** (in our OoD environment).

- **VScode**: Microsoft's versatile open-source IDE that supports various languages (R, Python, MATLAB, STATA, LaTeX, etc.)
- **Github Copilot**: One of the AI coding tools on the market. It offers a generous free tier and is approved for use in our OpenOnDemand (OoD) environment.

VScode can compile LaTeX and run code in Python, STATA, R, MATLAB.

- For Python, install the Python extension.
- For MATLAB, install the MATLAB extension and add your MATLAB path to the settings.
- Check out Paul Goldsmith-Pinkham's blog post. He discusses how to set up VScode to use LaTeX, STATA, and R. Link here: https://paulgp.com/2024/06/24/llm_talk.html

AI coding assistant tool in general

AI coding assistants are wrappers around LLM APIs.

- They use models from Claude, Open AI, Google, and others to help you write code, papers, slides, letters, and more.

How LLMs are used:

- At their core, LLMs are good at (and, trained in partly for) ***next-word prediction***. The most obvious application is **autocompletion**.
- But LLMs are capable of much more than just autocompletion.

Most AI coding tools, including GitHub Copilot, operate in three main modes:

1. Ask Mode
2. Edit Mode / Inline Editing
3. Agent Mode

Today, I'll show examples based on the first two modes. The third will be covered in the next session.

I'll illustrate the AI coding tool's capabilities in the context of three typical use cases for economists.

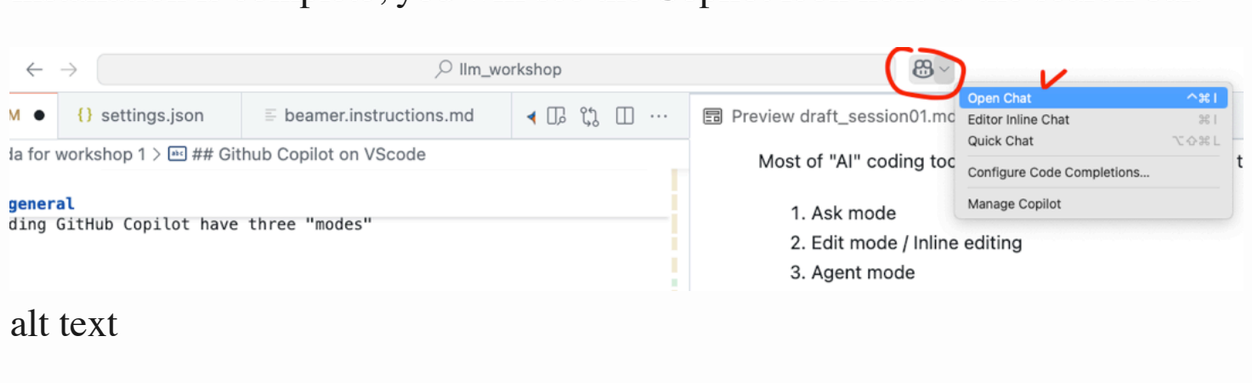
1. **Example 1**: Understanding the current directory structure and content.
2. **Example 2**: Modifying existing code.
3. **Example 3**: Drafting with LaTeX.

These are just examples to give you a rough idea of what you can do with it. Of course, its applications to your own work are limitless, and I encourage you to explore them on your own.

Github Copilot on VScode

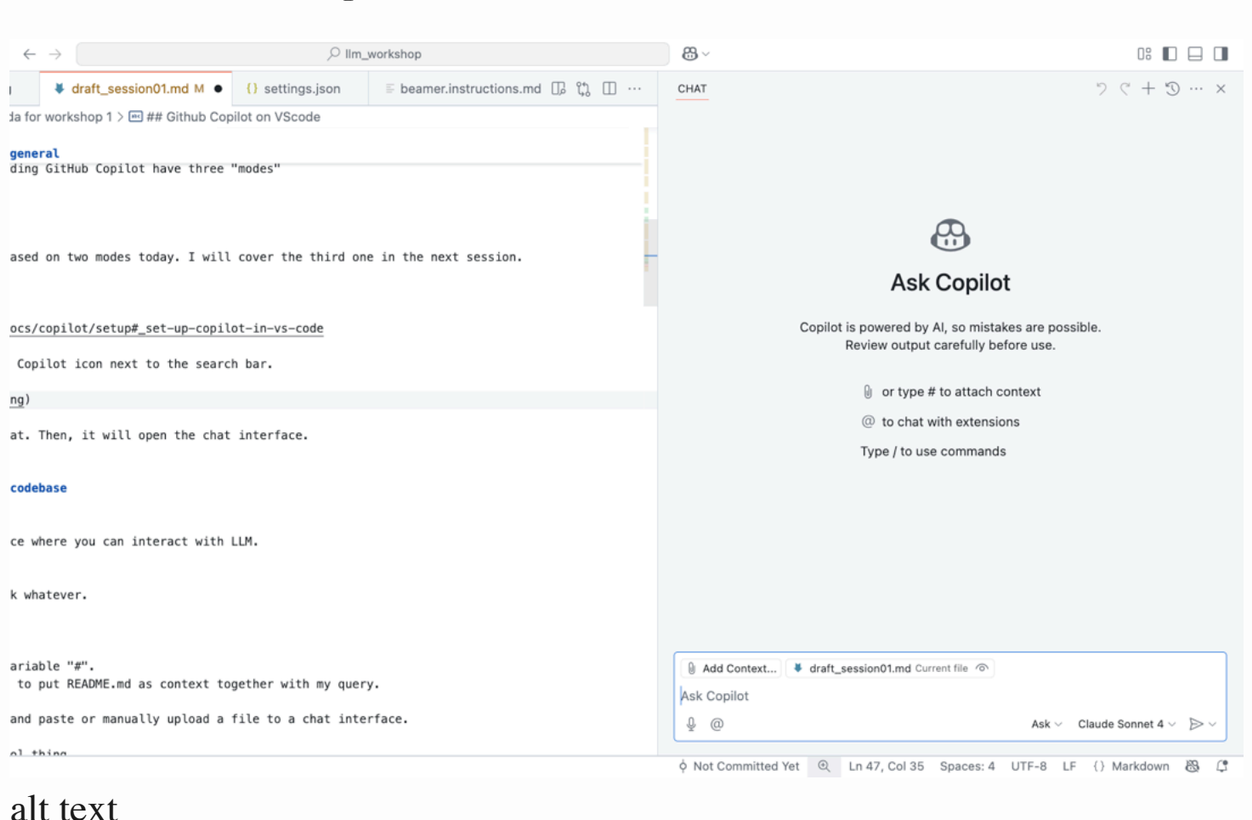
Installing Copilot in VS Code. Follow the instructions here: https://code.visualstudio.com/docs/copilot/setup#_set-up-copilot-in-vs-code

Once installation is complete, you will see the Copilot icon next to the search bar.



alt text

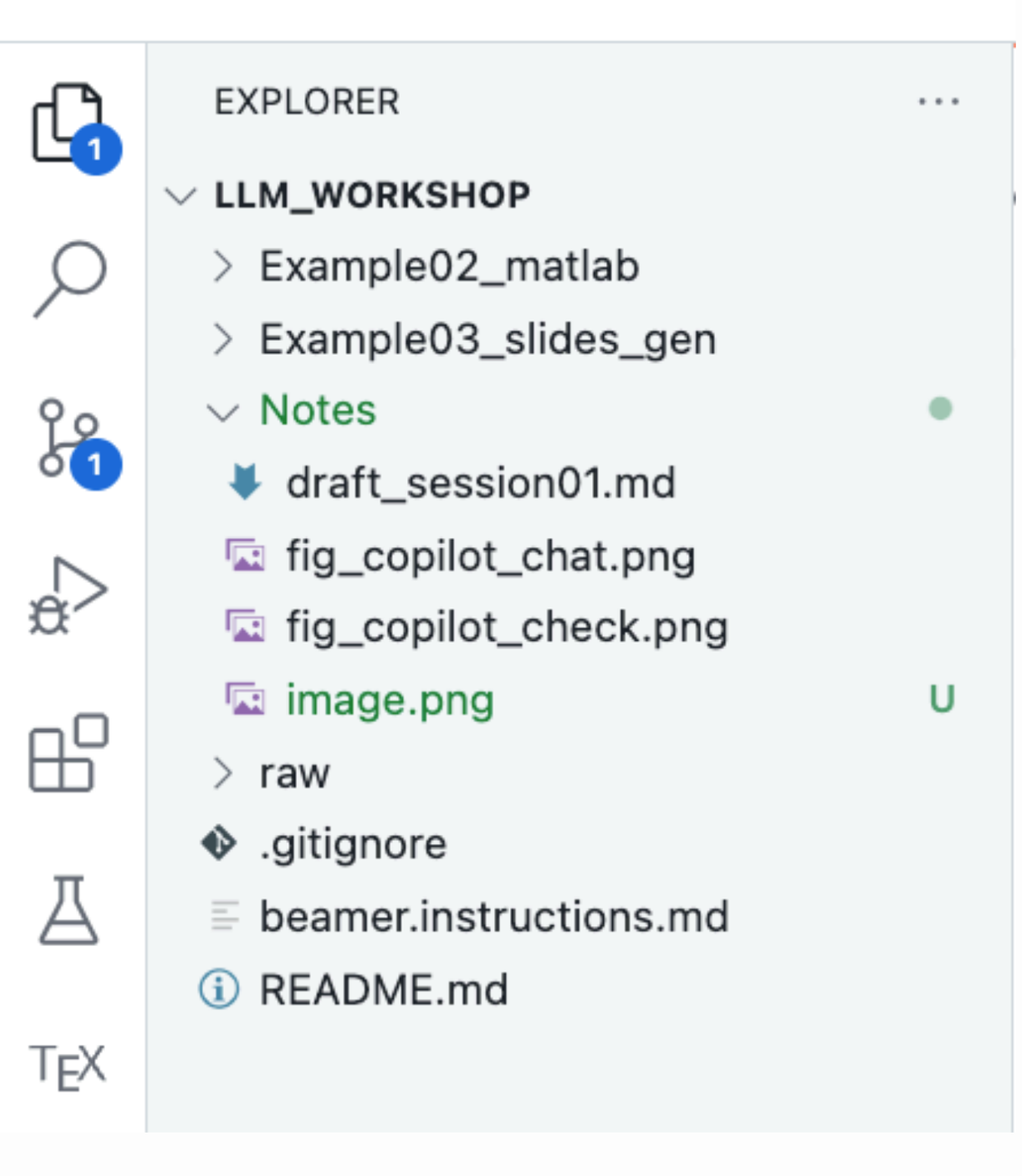
Click the icon and select Open Chat. This will launch the chat interface.



alt text

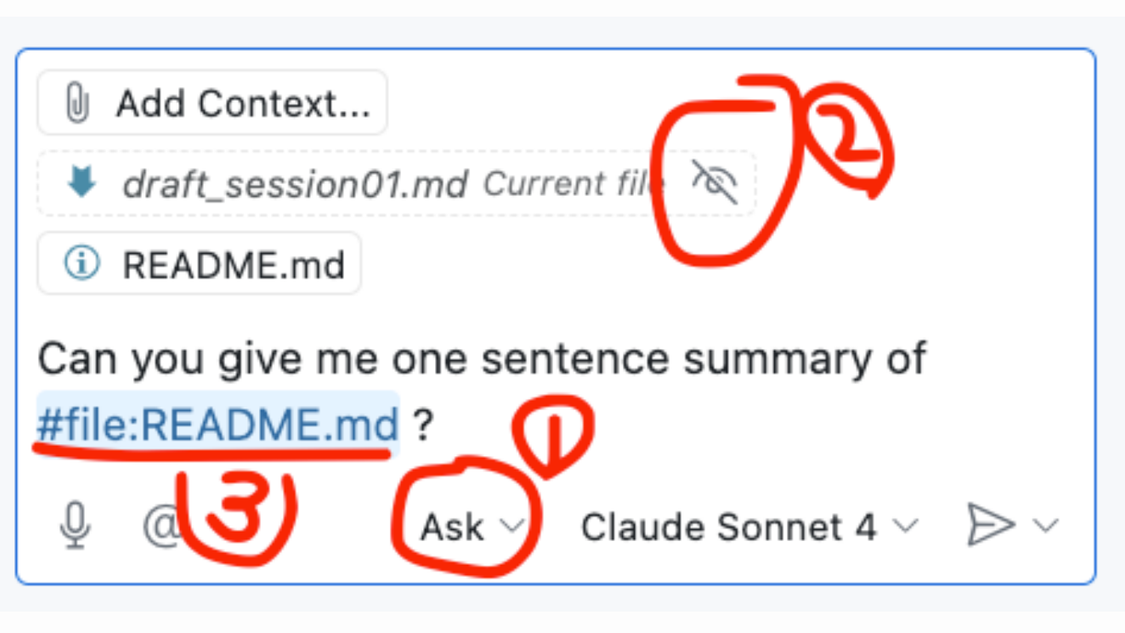
Example 1: Using AI to understand the current directory and content

Here is a typical working folder for your application. It contains multiple folders and several files. You may have written this application yourself, it may have come from a coauthor, or it could be code from a paper you're interested in.



Let's try to understand what this directory does.

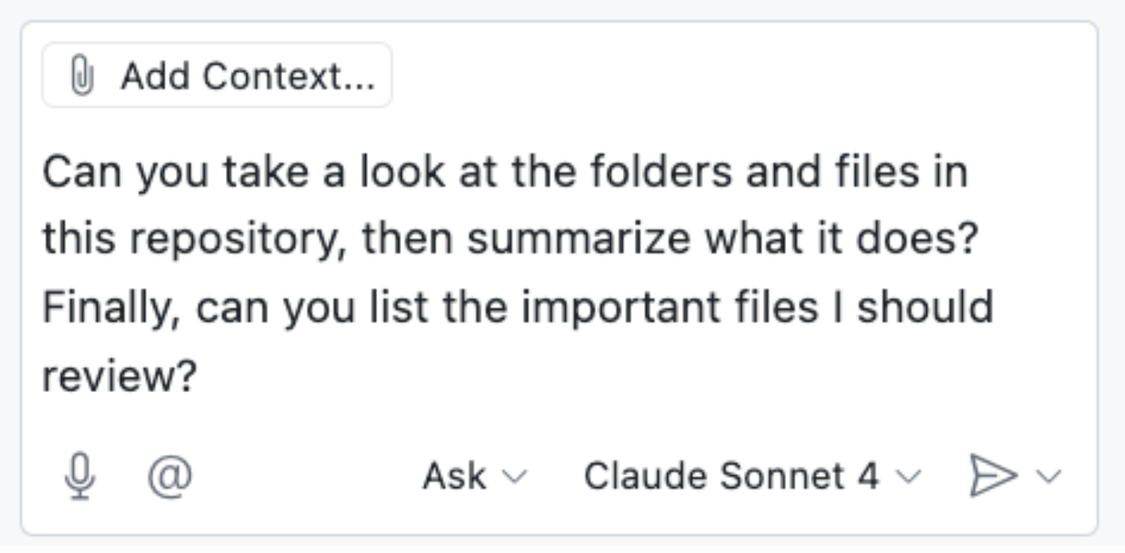
Thankfully, there is README.md file for this repository. Let's first look at this file.



1. Set it to **Ask Mode**. Then, it will not try to change the file.
2. Current viewing file is automatically added to the chat. Turn it off if that is not part of your request.
3. The symbol # is used to denote a context variable. #file:README.md means "I want to include README.md as context toegether with my query."

We can ask copilot about the entire codebase as well.

Sometimes, I ask my RA to explain me someone else's code from an interesting paper that I read. It may take a long time for many reasons. I still do think that it is a good exercise for a training purpose. But, it takes time. Now, I use AI tool.



Example 2: MATLAB example

Now, let us take a look at more realistic use case.

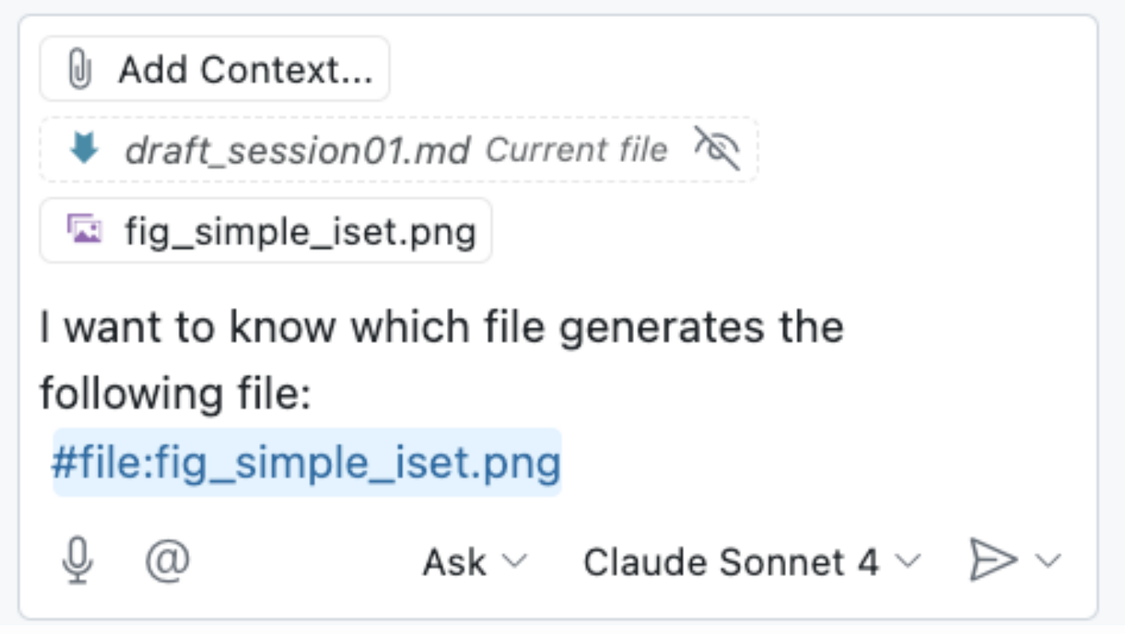
This folder contains the replication code for a paper that Jonas and I just finished.

We are about to submit this paper. Suppose Jonas asked me to chnage the color of one of the figures I created (fig_simple_iset.png). Over the past two weeks, I've been busy preparing this presentation and handling other tasks, so I forgot what I did.

If this were five years ago, I would have asked my RA to go through the code and make the modification. The RA would then:

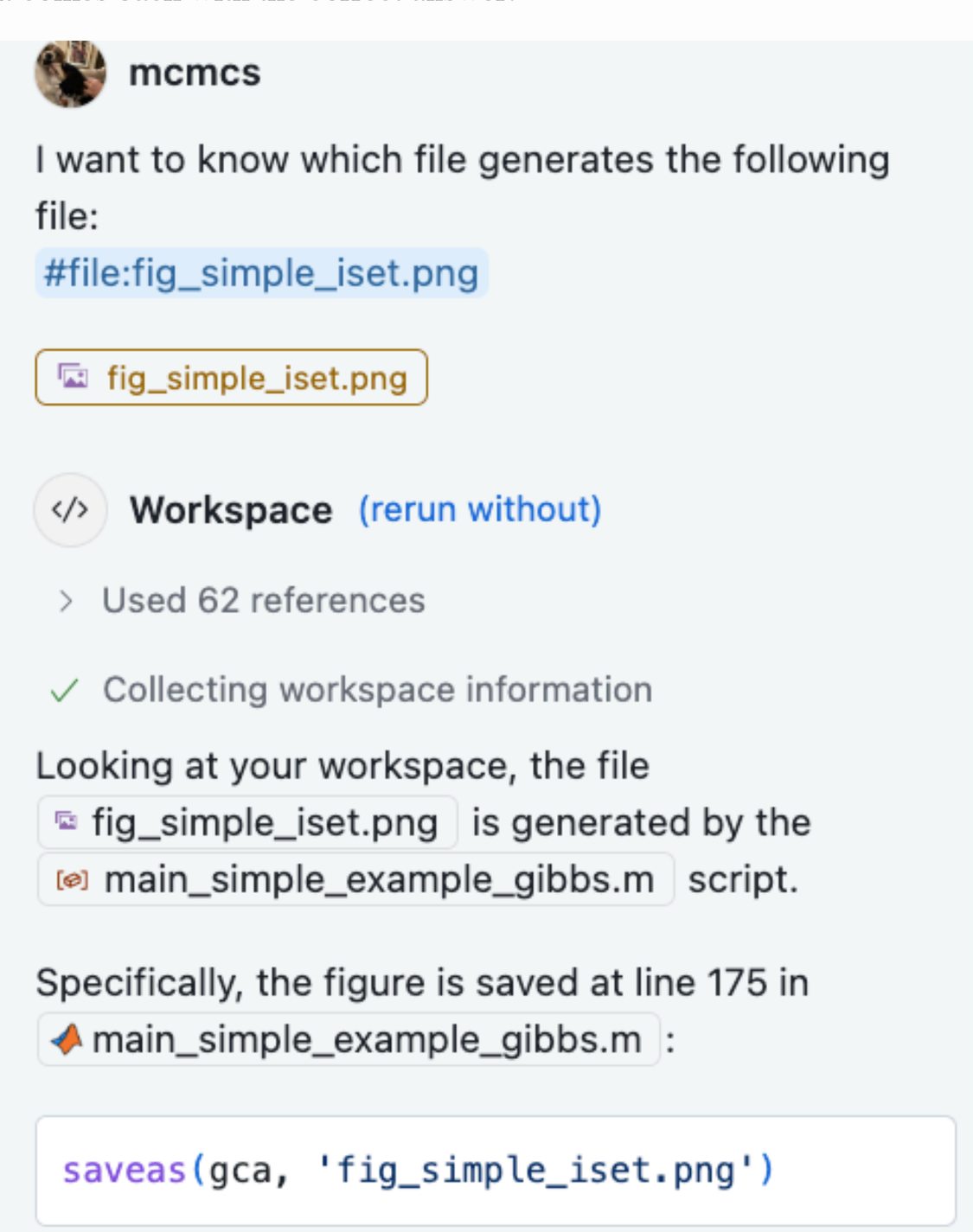
- First, find the code that generates the figure.
- Second, identify the specific part that creates the figure.
- Third, modify that part accordingly.

Let's see if we can replicate this course of action using an AI tool.



- Note that I used the "#" symbol to tell the AI that I want to include this specific file as context.

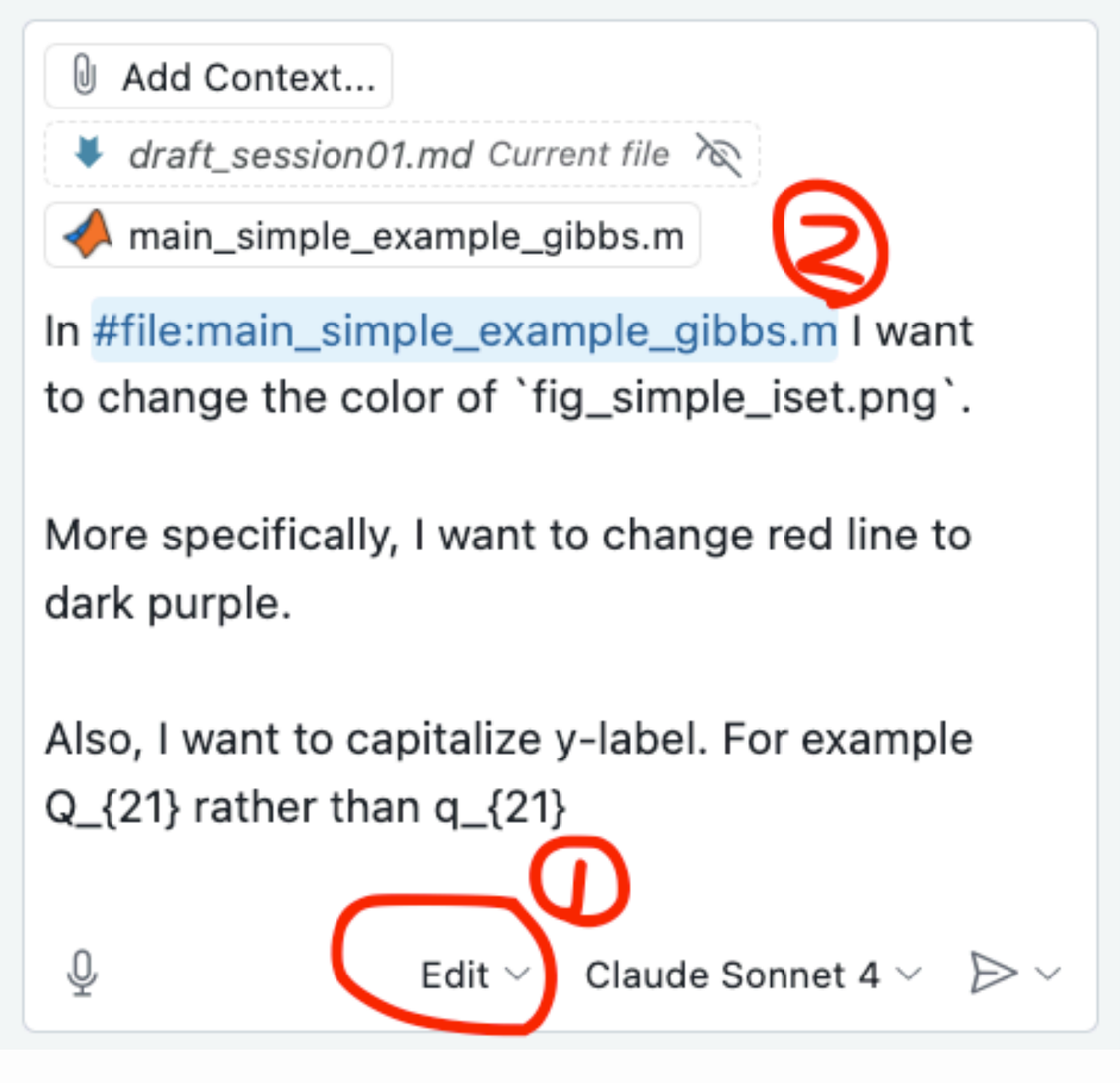
It comes back with the correct answer:



Under the hood, it runs a couple of function/tool calls such as search, read, find files. Then, figures out

- Which file creates the figure.
- In addition, it tells me exact line of the code where this figure is created.

Now, let's see if we can modify the code. There are a couple of ways to do this. First, let's ask the AI to edit the code.



1. Note that we switched to Edit mode so that the AI can edit our code.
2. Again, use # to include a specic file as context.

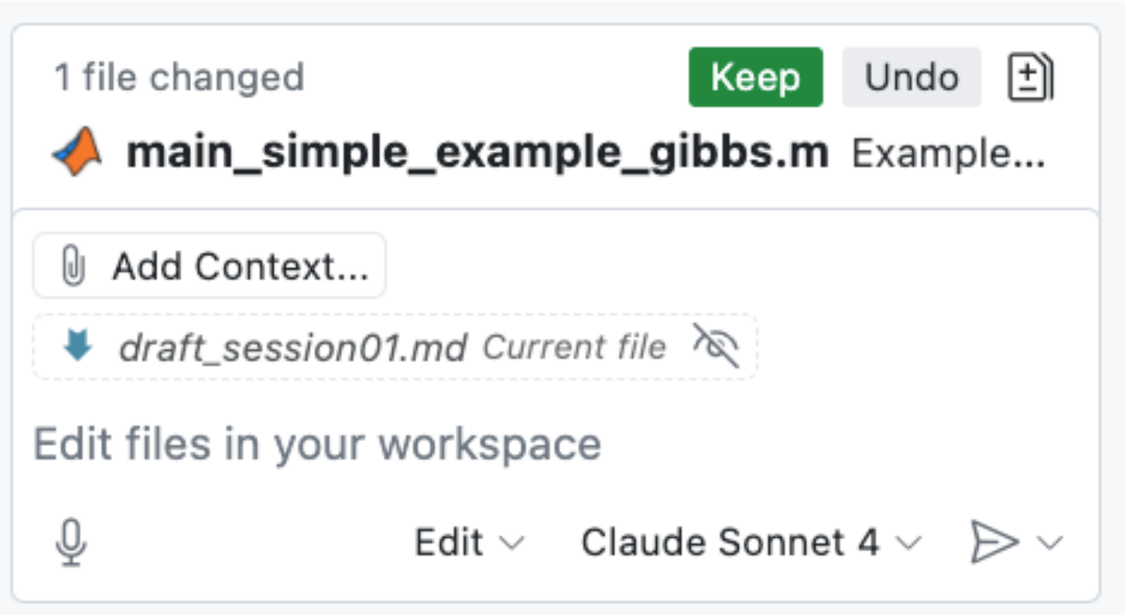
The AI returns with its proposal:

- Red: lines to be deleted
- Green: lines to be added

```
% 2) mat_q2 ▸ light-red '+' markers
% 2) mat_q2 ▸ dark purple 'o' markers
scatter(mat_q2(:,1), mat_q2(:,2), ...
        75              , ...                  % marker size (bigger f
       'o'             , ...                   % marker symbol
       'LineWidth'     ,2, ...
       'MarkerEdgeColor',lightRed);            % colour
       'MarkerEdgeColor',darkPurple);          % colour

hold off
grid on
xlabel('$q_{11}$');
ylabel('$q_{21}$');
ylabel('$Q_{21}$');
set(gca, 'linewidth', 2, 'fontsize', 15);
```

Click "V" to "accept" the change line by line. Or, Click "Keep" to "accept" all changes simultaneously.



AI coding tools make our lives easier:

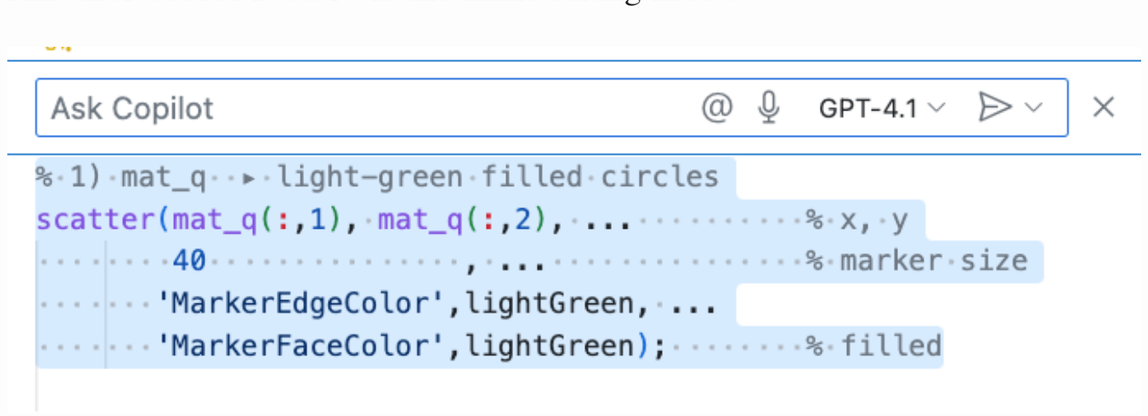
- They automatically add relevant context for you (for example, the currently viewed file).
- They automatically apply suggested changes from the language model.
- If needed, they use built-in functions (like search, read, find, etc.) to provide better context to the LLM and generate more accurate answers.

You can always revert changes using "Ctl + Z" as usual.

Of course, you should use version control software like Git.

Example 3: Inline editing

Another useful feature is the inline editing mode.

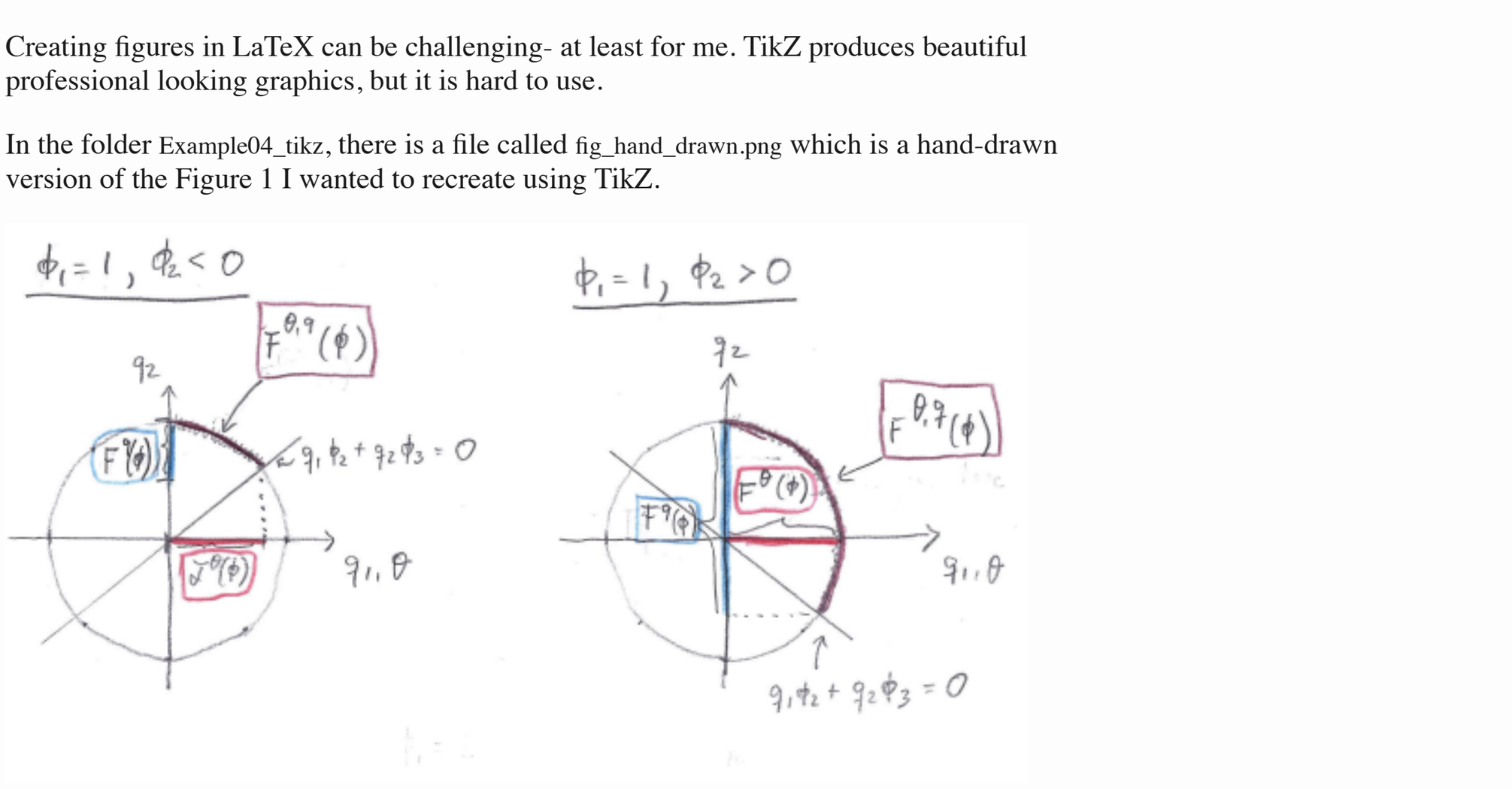


- When you want to change a specific part of the code, select the lines you want to work on and press Win + I (or CMD + I). This will trigger inline editing mode.
- You can then ask the AI to code edits related to the selected lines.

Example 4: LaTeX example (TikZ)

Creating figures in LaTeX can be challenging- at least for me. TikZ produces beautiful professional looking graphics, but it is hard to use.

In the folder Example04_tikz, there is a file called fig_hand_drawn.png which is a hand-drawn version of the Figure 1 I wanted to recreate using TikZ.



See email_from_frank.docx for the motivation of this example. This is a complex figure with many components crowded into one image.

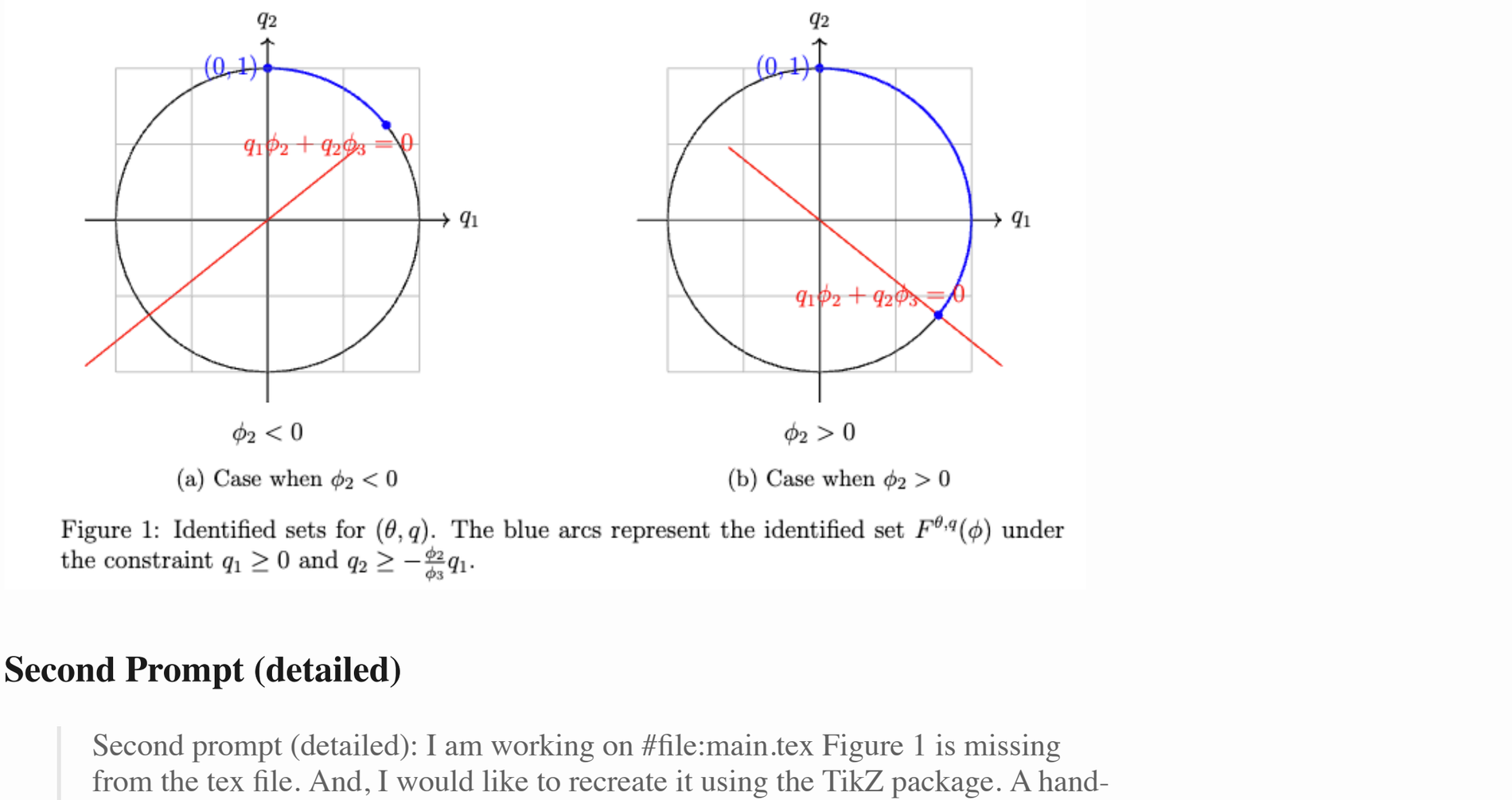
There are two example prompts:

- The first prompt is brief and AI will make mistake with high chance.
- The second prompt is more detailed and has a higher chance of producing an accurate figure.

First Prompt (simple)

> I am working on #file:main.tex Figure 1 is missing from the tex file. And, I would like to recreate it using the TikZ package. A hand-drawn version of the figure is available in #file:fig_hand_drawn.png. Could you recreate it using TikZ? The figure consists of two panels. Ensure that both panels fit on the page and are appropriately sized.

Result: Not bad as a starting point. How can we make AI work better? Garbage in, garbage out? Let's try to be more specific when we ask AI to work for us.



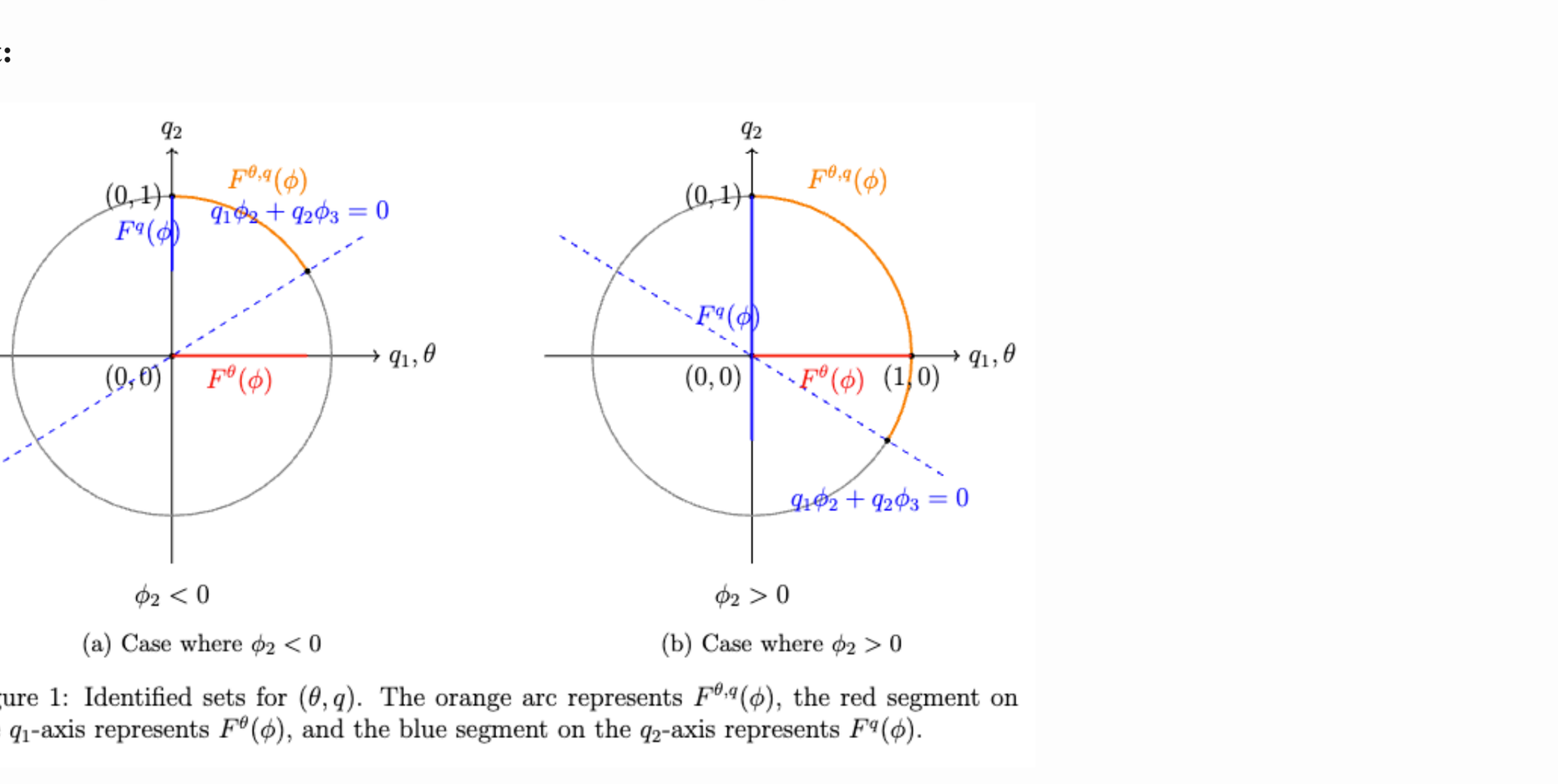
(a) Case when $\phi_2 < 0$ (b) Case when $\phi_2 > 0$

Figure 1: Identified sets for (θ, q). The blue arcs represent the identified set $F^{\theta,q}(\phi)$ under the constraint $q_1 \geq 0$ and $q_2 \geq -\frac{\phi_2}{\phi_3} q_1$.

Second Prompt (detailed)

> Second prompt (detailed): I am working on #file:main.tex Figure 1 is missing from the tex file. And, I would like to recreate it using the TikZ package. A hand-drawn version of the figure is available in #file:fig_hand_drawn.png. Could you recreate it using TikZ? The figure consists of two panels. Ensure that both panels fit on the page and are appropriately sized. Below is a detailed description of the first panel: This is a 2D geometric figure in the (q_1, q_2) space with the following components:
>
> **Axes**: - Cartesian coordinate axes: - Horizontal axis labeled as q_{1}, - Vertical axis labeled as q_{2} - Origin is at the intersection of these axes.
>
> **Circle**: - A unit circle is drawn centered at the origin.
>
> **Constraint line**: - There is a line defined by q_{1} *{2} + q{2} {3} = 0 where* {1} = 1 and _{2} <0.
>
> **Arc Segment**: - A thick arc (approximately from the top to upper-right quadrant) is drawn over the circle (orange color). - Left end point of the arc is (0,1) - Right end point of the arc is the point where the circle and the constraint line meets. - This arc is labeled F^{, q}()
>
> **Identified sets**: - There are two additional identified sets shown in this figure. - On q_{1} axis, there is red area denoted by $F^{\theta}(\phi)$ from (0,0) to q_1 value of the right end point of the arc. - On q_{2} axis, there is blue area denoted by $F^{q}(\phi)$ from q_2 value of the right end point of the arc to (0,1).
>
> Description for the second figure is here:
>
> Second figure is similar to the first one but with positive ϕ_2 value. So that line is negative sloping. - Arc should go from (0,1) to the point where it meets the line (lower right quadrant) - Blue area is extended accordingly. (1,0) to the q_2 value of the right end of the arc. - Red area is extended accordingly. (0,0) to (0, 1).

Output:



(a) Case where $\phi_2 < 0$ (b) Case where $\phi_2 > 0$

Figure 1: Identified sets for (θ, q). The orange arc represents $F^{\theta,q}(\phi)$, the red segment on the q_1-axis represents $F^{\theta}(\phi)$, and the blue segment on the q_2-axis represents $F^{q}(\phi)$.

For a complex task like this, you need to give detailed instructions to the AI.

Even with the detailed prompt, there's still a chance the output will be incorrect or the resulting file won't compile. In such cases, copy and paste the error messages into the chat. After a few iterations, the AI should be able to identify and fix the problem.