**✅ Simple Python Code Example**

Let’s say we’re implementing **matrix multiplication**. We’ll show two versions:

1. **Naive Algorithm** (CPU-style, not hardware-aware)
2. **Tiled Matrix Multiplication** (memory-efficient, system-aware for GPU or cache-friendly design)

**1️⃣ Naive Matrix Multiplication (Reference Algorithm)**

python

CopyEdit

import numpy as np

def naive\_matmul(A, B):

M, K = A.shape

K2, N = B.shape

assert K == K2

C = np.zeros((M, N))

for i in range(M):

for j in range(N):

for k in range(K):

C[i, j] += A[i, k] \* B[k, j]

return C

# Example

A = np.random.rand(128, 128)

B = np.random.rand(128, 128)

C = naive\_matmul(A, B)

💡 This is **slow** and doesn't take advantage of memory locality, vectorization, or GPUs.

**2️⃣ System-Aware (Tiled) Matrix Multiplication**

This one is **co-designed** with a system in mind — for example, CPU cache or GPU shared memory.

python

CopyEdit

def tiled\_matmul(A, B, tile\_size=32):

M, K = A.shape

K2, N = B.shape

assert K == K2

C = np.zeros((M, N))

for i in range(0, M, tile\_size):

for j in range(0, N, tile\_size):

for k in range(0, K, tile\_size):

# Extract tiles

A\_tile = A[i:i+tile\_size, k:k+tile\_size]

B\_tile = B[k:k+tile\_size, j:j+tile\_size]

C[i:i+tile\_size, j:j+tile\_size] += np.dot(A\_tile, B\_tile)

return C

# Example

C\_tiled = tiled\_matmul(A, B)

✅ This design:

* Reduces memory traffic
* Uses small matrix blocks (tiles) that fit into cache/shared memory
* Performs better for large matrices

**💬 Real-World Co-Design Example**

In deep learning:

* **Algorithm**: Choose convolution types, activation functions, or attention mechanisms that match hardware strengths.
* **System**: Customize for GPU (using CUDA), TPU (using XLA), or FPGA (custom ops).

**🧠 TL;DR**

**Algorithm-system co-design** = writing code that considers the **underlying hardware**, using patterns like tiling, memory reuse, and parallelism.

Want to see a GPU-accelerated version using PyTorch or CuPy next? Or a co-design example for transformers or CNNs?

4o

**You said:**

can you fix the code import numpy as np ModuleNotFoundError: No module named 'numpy'

**ChatGPT said:**

Sure! That error means Python can't find the numpy library, which is required for numerical operations like matrix multiplication.

**✅ To Fix It, You Need to Install NumPy**

If you're using a terminal or command line, run:

bash

CopyEdit

pip install numpy

If you're using **Jupyter Notebook**, you can run this inside a cell:

python

CopyEdit

!pip install numpy

Or if you're in a **conda environment**, you can use:

bash

CopyEdit

conda install numpy

**🔁 After Installing, Try Again**

Then re-run your Python script with:

python

CopyEdit

import numpy as np

And it should work without error. Let me know if you’re using a specific platform like Google Colab, VS Code, or Anaconda — I can give more tailored help!

4o

**You said:**

for pycharm how to correct it import numpy as np ModuleNotFoundError: No module named 'numpy'

**ChatGPT said:**

Ah, got it — you're using **PyCharm**! That error means NumPy isn’t installed in the Python environment PyCharm is using. Here's how to fix it:

**✅ Step-by-Step: Fix ModuleNotFoundError: No module named 'numpy' in PyCharm**

**🔍 1. Check Your Python Interpreter in PyCharm**

Make sure you're using the correct environment:

1. Go to **File > Settings** (or **PyCharm > Preferences** on macOS).
2. Navigate to **Project: [Your Project Name] > Python Interpreter**.
3. You’ll see a list of packages in the selected environment.

If NumPy is not in the list — let’s install it.

**⚙️ 2. Install NumPy in PyCharm**

**Option A: GUI Installation**

1. In the Python Interpreter window (above), click the **+** button (top-right).
2. Search for numpy.
3. Click **Install Package**.

**Option B: Terminal Installation**

1. Open the **terminal** in PyCharm (bottom panel).
2. Run:

bash

CopyEdit

pip install numpy

If that doesn't work, try:

bash

CopyEdit

python -m pip install numpy

**🔁 3. Re-run Your Script**

Try running your code again:

python

CopyEdit

import numpy as np

A = np.random.rand(3, 3)

B = np.random.rand(3, 3)

print(np.dot(A, B))

**🧠 Tip: Using Virtual Environments?**

If your project uses a **virtualenv or conda environment**, make sure:

* You’ve installed NumPy **inside** that environment
* PyCharm is **pointing to the same interpreter**

Let me know if you'd like help checking which Python environment is active, or if you want to set up a clean virtualenv in PyCharm.

4o

**You said:**

PS C:\Users\melde\OneDrive\Desktop\ECE 510 Spring2015\week 2> & C:/Users/melde/AppData/Local/Programs/Python/Python313/python.exe "c:/Users/melde/OneDrive/Desktop/ECE 510 Spring2015/week 2/week2 proj" File "c:\Users\melde\OneDrive\Desktop\ECE 510 Spring2015\week 2\week2 proj", line 1 ef tiled\_matmul(A, B, tile\_size=32): ^^^^^^^^^^^^ SyntaxError: invalid syntax PS C:\Users\melde\OneDrive\Desktop\ECE 510 Spring2015\week 2>

**ChatGPT said:**

Ah — I see the issue! The error is:

javascript

CopyEdit

SyntaxError: invalid syntax

And the arrow points to:

python

CopyEdit

ef tiled\_matmul(A, B, tile\_size=32):

^^^^^^^^^^^^

**✅ The Fix**

You just have a **typo** in the function definition. You wrote:

python

CopyEdit

ef tiled\_matmul(A, B, tile\_size=32):

It should be:

python

CopyEdit

def tiled\_matmul(A, B, tile\_size=32):

Just add the missing **d** at the beginning (def instead of ef).

**🔁 After Fixing It**

Your full code should start like this:

python

CopyEdit

import numpy as np

def tiled\_matmul(A, B, tile\_size=32):

M, K = A.shape

K2, N = B.shape

assert K == K2

C = np.zeros((M, N))

for i in range(0, M, tile\_size):

for j in range(0, N, tile\_size):

for k in range(0, K, tile\_size):

A\_tile = A[i:i+tile\_size, k:k+tile\_size]

B\_tile = B[k:k+tile\_size, j:j+tile\_size]

C[i:i+tile\_size, j:j+tile\_size] += np.dot(A\_tile, B\_tile)

return C

# Example usage

A = np.random.rand(128, 128)

B = np.random.rand(128, 128)

C = tiled\_matmul(A, B)

print(C)

Let me know if you want to add a performance timer or compare it to regular matrix multiplication using np.dot() or @.

4o

**You said:**

[{ "resource": "/c:/Users/melde/OneDrive/Desktop/ECE 510 Spring2015/week 2/week2 proj", "owner": "python", "code": { "value": "reportUndefinedVariable", "target": { "$mid": 1, "path": "/microsoft/pyright/blob/main/docs/configuration.md", "scheme": "https", "authority": "github.com", "fragment": "reportUndefinedVariable" } }, "severity": 4, "message": "\"np\" is not defined", "source": "Pylance", "startLineNumber": 13, "startColumn": 52, "endLineNumber": 13, "endColumn": 54 }]

**ChatGPT said:**

That error means you're using **np** in your code, but Python doesn't know what np is — because **NumPy hasn't been imported** properly.

**🔧 Fix It: Add This at the Top of Your Script**

Make sure your code begins with this line:

python

CopyEdit

import numpy as np

This tells Python:

* "Hey, I’m using the numpy library"
* And I'm going to refer to it as np (a common convention)

**✅ Example (Corrected)**

Here’s a working version of your code with the proper import:

python

CopyEdit

import numpy as np # ✅ This is the important missing line

def tiled\_matmul(A, B, tile\_size=32):

M, K = A.shape

K2, N = B.shape

assert K == K2

C = np.zeros((M, N))

for i in range(0, M, tile\_size):

for j in range(0, N, tile\_size):

for k in range(0, K, tile\_size):

A\_tile = A[i:i+tile\_size, k:k+tile\_size]

B\_tile = B[k:k+tile\_size, j:j+tile\_size]

C[i:i+tile\_size, j:j+tile\_size] += np.dot(A\_tile, B\_tile)

return C

# Example usage

A = np.random.rand(128, 128)

B = np.random.rand(128, 128)

C = tiled\_matmul(A, B)

print(C)

**💡 Bonus Tip: Pylance / Pyright**

That error message came from **Pylance** or **Pyright**, your Python code checker in VS Code or PyCharm. It's just helping you catch problems early — which is great!

Let me know if you want help enabling autocomplete or fixing other lint warnings too.

4o

**You said:**

thon313/python.exe "c:/Users/melde/OneDrive/Desktop/ECE 510 Spring2015/week 2/week2 proj" PS C:\Users\melde\OneDrive\Desktop\ECE 510 Spring2015\week 2>

**ChatGPT said:**

It looks like you ran your script using:

powershell

CopyEdit

python.exe "c:/Users/melde/OneDrive/Desktop/ECE 510 Spring2015/week 2/week2 proj"

But nothing happened — no output or error — right?

**✅ Here's What Might Be Going On**

1. **The script is valid**, but you're not printing anything.
2. Or the script file has no code (or it’s all commented out).
3. Or it's not running due to the **missing .py extension**.

**🔧 Let's Fix It**

**🛠 1. Make Sure the File Has a .py Extension**

Your script should be named something like:

CopyEdit

week2\_proj.py

Right now it’s "week2 proj" — no extension, and spaces in the name (which can confuse things).

You can:

* Rename the file to week2\_proj.py
* Or use quotes if keeping the space (but .py extension is still important)

**🛠 2. Add Output to See If It Runs**

Add this to the end of your script to make sure it runs something:

python

CopyEdit

print("Script is working!")

# Example matrix multiplication

import numpy as np

A = np.random.rand(4, 4)

B = np.random.rand(4, 4)

C = np.dot(A, B)

print("Matrix A:")

print(A)

print("Matrix B:")

print(B)

print("Result C = A \* B:")

print(C)

**🛠 3. Re-run It**

**A screenshot of a computer program

Description automatically generated**