In Challenge #20, the goal was to implement a 4×4 resistive crossbar network to perform analog matrix-vector multiplication using SPICE. The crossbar consisted of fixed-value resistors connecting each input row to each output column, forming a structure that mimics a dot product between a 4×1 input voltage vector and a 4×4 weight matrix. The output was measured as the current flowing through each column, determined by Ohm’s Law via 1 Ω resistors connected to ground.

The implementation was carried out using Ngspice in a Google Colab environment. We wrote the SPICE netlist using the %%writefile command to define input voltage sources, the weight matrix (resistors at each crosspoint), and ground path resistors for current measurement. Ngspice was executed using Python’s subprocess.run(), and the output was parsed directly in Python to extract node voltages at each column output. These voltages were then divided by 1 Ω to compute the output currents, effectively yielding the matrix-vector product.

During the process, a few challenges were encountered. Initially, attempts to use the .print directive inside and outside of .control blocks led to errors stating that the node voltages (e.g., v(col1)) were not found. This issue was resolved by using .control mode with the print all command, which ensured all node voltages were included in the output, making them accessible to our Python parser. The output of the SPICE simulation, which includes voltage values like col1 = 1.2456 V, was processed in Python and converted to corresponding currents (1.2456 A) using the known 1 Ω resistors.

This challenge successfully demonstrated an analog form of matrix-vector multiplication using SPICE simulation in a cloud-based Python environment. We verified the system’s functionality by confirming that the output currents responded correctly to different input voltage vectors and resistor configurations. While the current implementation was static, it lays a solid foundation for further expansion—such as parameterizing the input vector from Python, visualizing results with matplotlib, or packaging the entire simulation into a reusable Python function or notebook.

Overall, Challenge #20 integrated SPICE circuit modeling with modern data science tools, illustrating how hardware-based computation principles can be explored interactively using Python and cloud simulation environments like Colab.