Electromagnetism Simulation Suite Manual

This user's manual provides a centralized guide for the **Electromagnetism Simulation Suite**. It allows for the calculation and visualization of electrostatics for an arbitrary number of point charges.

---

1. Prerequisites

Ensure the following tools are installed in your environment:

- **Python 3.x:** With numpy and matplotlib libraries.
- **Gnuplot:** (v5.0 or higher recommended).
- **Bash-compatible Shell:** (Linux Terminal, macOS Terminal, or Git Bash on Windows).

---

2. File Architecture

- charges.txt: Input data file containing charge locations and magnitudes.
- coulombLawyer.py: Python engine for physical calculations.
- plot_field.gp: Gnuplot script for vector and contour rendering.
- run_all.sh: Automation script to execute the full workflow.

---

3. Data Preparation (charges.txt)

Create a space-separated text file where each line defines a point charge in the 2D plane:

```
# x-coord  y-coord  charge(q)
2.0       0.0     1.0
-2.0      0.0     -1.0
```

---

4. Execution

Open your terminal in the project directory and run:

```
chmod +x run_all.sh
```

`./run_all.sh`

The script will first compute the physics, display a 3D interactive plot of the potential Φ, and then generate a high-resolution 2D image (electrostatics_combined.png).

---

5. Understanding the Physics Output

The final visualization overlays two critical datasets:

1. **Electric Field Vectors ($\mathbf{E}$):** Black arrows indicating the force direction on a positive test charge. They diverge from positive sources and converge into negative sinks.
2. **Equipotential Contours (Φ):** Colored curves representing lines of constant potential. Note that the $\mathbf{E}$ vectors are mathematically required to be locally perpendicular to these lines.

---

6. Tuning for Quality

- **Smoothing:** If contours appear jagged, increase the grid_res in the Python script or adjust the gauss parameters in the dgrid3d command within plot_field.gp.
- **Contour Density:** Adjust set cntrparam levels incremental [min], [step], [max] in the Gnuplot script to focus on specific potential ranges.
- **Vector Density:** Change the every n:n parameter in the Gnuplot plot command to make the vector field more or less dense.

---

7. Troubleshooting

- **IndentationError:** Ensure your Python script uses consistent 4-space indentation and no mixed tabs.
- **Command Not Found:** On Windows, ensure you are using **Git Bash** or **WSL** to run the .sh script.
- **Singularities:** The code "softens" the $1/r$ singularity at the exact charge location to prevent infinite values from breaking the plot. If you see blank spots, increase the "softening" constant in the Python loop.
