

SOLUTION of (4-1.b):

C++ code 4.1.3: Plotting of spectrum of the images.

```
2 void plot_freq(double focus) {
3     constexpr unsigned int a = 0;
4     constexpr unsigned int b = 8000;
5     auto clamp = [](double x) { return x < a ? a : x > b ? b : x; };
6
7     // TODO: (4-1.b) compute D containing the spectrum of
8     set_focus(focus).
9     // "clamp" the data between 0 and 8000.
10    // START
11    Eigen::FFT fft;
12    Eigen::MatrixXd D = fft.fwd(set_focus(focus)).cwiseAbs().unaryExpr(clamp) / b;
13    // END
14
15    // Plot values of X.
16    // Axis labels
17    std::vector<double> xticks(5);
18    for (int i = 0; i < 5; ++i) {
19        xticks[i] = i * (D.cols() - 1) / 4;
20    }
21    std::vector<double> yticks(5);
22    for (int i = 0; i < 5; ++i) {
23        yticks[i] = i * (D.rows() - 1) / 4;
24    }
25    std::vector<std::string> labels{"-1", "-0.5", "0", "0.5", "1"};
26
27    plt::figure();
28    plt::imshow(D, {{ "cmap", "viridis" }, { "origin", "lower" }});
29    plt::colorbar();
30    plt::xticks(xticks, labels);
31    plt::yticks(yticks, labels);
32    std::stringstream ss;
33    ss << "Spectrum with f = " << focus << ".";
34    plt::title(ss.str().c_str());
35    std::stringstream ss2;
36    ss2 << "./cx_out/spectrum_focus=" << focus << ".png";
37    plt::savefig(ss2.str().c_str());
38 }
```

Get it on 🐱 GitLab (autofocus.hpp).

