LA-II Assignment 3 Report

Artem Matevosian | DSAI-03 | a.matevosian@innopolis.university

May 04, 2023

Source code

See the whole solution at the GitHub repository in "Assignment 3" folder. (https://github.com/ArtMGreen/LA-Assignments) **Note:** the repository is made private in order to avoid plagiarism by other students. If you cannot access the repository, contact @artmgreen in Telegram.

```
int LA_Plotting_Lotka_Volterra(double v0, double k0,
                               double a1, double b1,
                               double a2, double b2,
                               double T, double N,
                               bool time_parametrized) {

    double equilibrium_v = a2 / b2;
    double equilibrium_k = a1 / b1;

    v0 -= equilibrium_v;
    k0 -= equilibrium_k;

    vector<double> t, v, k;

    cout << "t:" << endl;
    for (int i = 0; i <= N; i++) {
        cout << fixed << setprecision(2) << i * (T / N) << " ";
        t.push_back(i * (T / N));
    }
    cout << endl;

    cout << "v:" << endl;
    for (int i = 0; i <= N; i++) {
        double t_i = i * (T / N);
        double theta = t_i * pow(a1*a2, 0.5);
        double dv = v0 * cos(theta) - k0 * pow(a2/a1, 0.5) * (b1/b2) * sin(theta);
        cout << fixed << setprecision(2) << dv + equilibrium_v << " ";
        v.push_back(dv + equilibrium_v);
    }
    cout << endl;

    cout << "k:" << endl;
    for (int i = 0; i <= N; i++) {
        double t_i = i * (T / N);
        double theta = t_i * pow(a1*a2, 0.5);
        double dk = v0 * pow(a1/a2, 0.5) * (b2/b1) * sin(theta) + k0 * cos(theta);
        cout << fixed << setprecision(2) << dk + equilibrium_k << " ";
        k.push_back(dk + equilibrium_k);
    }
    cout << endl;
```

```
    Gnuplot gp;

    double min_x = numeric_limits<double>::max();
    double max_x = numeric_limits<double>::min();
    double min_y = numeric_limits<double>::max();
    double max_y = numeric_limits<double>::min();

    if (time_parametrized) {
        for (int i = 0; i <= N; i++) {
            if (v[i] < min_y) min_y = v[i];
            if (v[i] > max_y) max_y = v[i];
            if (k[i] < min_y) min_y = k[i];
            if (k[i] > max_y) max_y = k[i];
        }

        min_x = 0; min_y -= 2; max_x = T; max_y += 2;
    } else {
        for (int i = 0; i <= N; i++) {
            if (v[i] < min_y) min_y = v[i];
            if (v[i] > max_y) max_y = v[i];
            if (k[i] < min_x) min_x = k[i];
            if (k[i] > max_x) max_x = k[i];
        }

        min_x -= 2; min_y -= 2; max_x += 2; max_y += 2;
    }

    gp << "set xrange [" << min_x << ":" << max_x << "]\n";
    gp << "set yrange [" << min_y << ":" << max_y << "]\n";

    if (time_parametrized) {
        gp << "plot '-' using 1:2 with lines linecolor rgb \"#00AA00\" title 'v(t)', ";
        gp << "'-' using 1:2 with lines linecolor rgb \"#AA0000\" title 'k(t)'\n";
        for (int i = 0; i <= N; i++) {
            double x = t[i];
            double y = v[i];
            gp << to_string(x)+"\t"+ to_string(y)+"\n";
        }

        gp << "e\n";

        for (int i = 0; i <= N; i++) {
            double x = t[i];
            double y = k[i];
            gp << to_string(x)+"\t"+ to_string(y)+"\n";
        }
    } else {
        gp << "plot '-' using 1:2 with lines linecolor rgb \"#666600\" title 'v(k)'\n";
        for (int i = 0; i <= N; i++) {
            double x = k[i];
            double y = v[i];
            gp << to_string(x)+"\t"+ to_string(y)+"\n";
        }
    }

    return 0;
}
```

```cpp
int main() {
    double v0, k0, N;
    double a1, a2, b1, b2, T;
    cin >> v0 >> k0 >> a1 >> b1 >> a2 >> b2 >> T >> N;

    LA_Plotting_Lotka_Volterra(v0, k0, a1, b1, a2, b2, T, N, true);
    LA_Plotting_Lotka_Volterra(v0, k0, a1, b1, a2, b2, T, N, false);

    return 0;
}
```

Variant of predator-prey system (test case)

`Assignment 3/test_case.txt` at GitHub repository contains the same test case:

```
45
15
0.4
0.01
0.3
0.005
100
500
```

Plotting & results

To execute `main.cpp`, compile it using `g++ -o LA_Plotting main.cpp -lboost_iostreams -lboost_system -lboost_filesystem`, run it by `./LA_Plotting` and copy the test case into the console.

The result should be as follows: