ソースコード

オイラー法

```
#include "iostream"
#include "math.h"

using namespace std;

double dxdy(double x, double y){
    return pow(x, 2) - y;
}

int main (void) {

    double a, b, eta;
    int n;

    cin >> a >> b >> eta >> n;

    cout << "x," << "y" << endl;

    double h = (b - a) / pow(2, n);

    double x, y;
    x = a;
    y = eta;

    while (x < 2) {
        y = y + h * dxdy(x, y);
        x += h;
    }

    cout << x << "," << y << endl;

    return 0;
}
```

ルンゲ・クッタ法

```cpp
#include "iostream"
#include "math.h"

using namespace std;

double f(double x, double y){
    return 3 * y / (1 + x);
}

int main (void) {

    double a, b, y0;
    int n;

    cin >> a >> b >> y0 >> n;

    cout << "n," << "x," << "y" << endl;

    double x, y;
    x = a;
    y = y0;

    double k1, k2, k3, k4;

    double h = (b-a) / pow(2,n);

    while (x < b) {
        k1 = f(x, y);
        k2 = f((x + h / 2.0), (y + k1 * h / 2.0));
        k3 = f((x + h / 2.0), (y + k2 * h / 2.0));
        k4 = f((x + h), (y + k3 * h));

        y = y + h * (k1 + 2.0*k2 + 2.0*k3 + k4) / 6;
        x += h;
    }

    cout << n << "," << x << "," << y << endl;

    return 0;
}
```