

Programowanie i metody numeryczne, 23.05.2024

Krótki plan zajęć:

- Rozwiązania numeryczne równań różniczkowych

Zadania:

Zadanie 1

Napisać klasę, która będzie służyła rozwiązywaniu równań różniczkowych. Klasa ta powinna mieć jako składowe: zakres całkowania, ilość kroków w tym zakresie, wskaźnik do funkcji odcałkowywanej oraz parametry i stan początkowy funkcji. Klasa ta powinna mieć dwie możliwości rozwiązywania równań różniczkowych: metodę Eulera oraz Rungego-Kutty 4 rzędu. Przykładowa deklaracja klasy może wyglądać następująco:

```
class ODESolver {

public:
    typedef void(ODESolver::*MethodPtr)(std::vector< std::vector<double> >&);
    typedef std::vector<double> (*DiffFuncType)(std::vector<double>&,
                                                std::vector<double>&, double);

private:

    DiffFuncType DiffFunc = nullptr;
    MethodPtr CurrentMethod = nullptr;

    double RangeBegin = 0.0;
    double RangeEnd = 1.0;
    double TimeStep = 0.01;
    int PointNum = 100;

    std::vector<double> FuncParams, InitParams;

    void EulerMethod(std::vector< std::vector<double> > &Solution);
    void RK4Method(std::vector< std::vector<double> > &Solution)

public:

    ODESolver(DiffFuncType Func = nullptr,
              double int_begin = 0.0, double int_end = 1.0,
              int point_num = 100,
              std::vector<double> func_params = {}, std::vector<double> init_params = {});

    void SetFunction( DiffFuncType Func );
    void SetRange(double int_begin, double int_end);
    void SetPoints(int point_num);
    void SetParams(std::vector<double> func_params);
    void SetInitState(std::vector<double> init_params);

    void SetMethodEuler();
    void SetMethodRK4();

    std::vector<std::vector<double>> Solve();

};
```