```
#ifndef RK4_SOLVER_HH
#define RK4_SOLVER_HH

#include <iostream>
#include <cassert>
#define N 1250

class rk4 {
  public:
    double AA;
    double BB;
    double J;
    double K;
    double h_step;
    double last_h_step;
    double Atol;
    double Rtol;
    double fac;
    double facmin;
    double facmax;
    double T_final;
    int n_intvls;
    double dense_stpsze;
    double* A;
    double* B;
    double* C;
    double* sc;
    double* x0;
    double* y0;
    double* theta0;
    double* vx0;
    double* vy0;
    double* omega0;
    double* x1;
    double* y1;
    double* theta1;
    double* x1h;
    double* y1h;
    double* theta1h;
    double* vx1;
    double* vy1;
    double* omega1;
    double* sc_x;
    double* sc_y;
    double* sc_theta;
    int step_counter;


    rk4(double hi_step, double tol, double J_, double K_, int n_intvls_){
      AA = 1.;
      BB = 1.;
      J  = J_;
      K  = K_;
      step_counter = 0;
      n_intvls = n_intvls_;
      x0 = new double[N];
      y0 = new double[N];
      theta0 = new double[N];
      vx0 = new double [N];
      vy0 = new double [N];
      omega0 = new double [N];
      x1 = new double[N];
      y1 = new double[N];
      theta1 = new double[N];
      x1h = new double[N];
      y1h = new double[N];
      theta1h = new double[N];
      sc_x = new double[N];
      sc_y = new double[N];
      sc_theta = new double[N];
      Rtol = tol;
      Atol = tol;
      fac = 0.9;
```

```
      facmax = 3.;
      facmin = 1./3.;
      h_step = hi_step;
      last_h_step = hi_step;
      A = new double[10];
      B = new double[9];
      C = new double[5];
      for(int i=0; i<10; i++){
        switch(i){
          case 0: A[i]=1./3.;
                  B[i]=1./8.;
                  C[i]=0.;
                  break;
          case 1: A[i]=-1./3.;
                  B[i]=3./8.;
                  C[i]=1./3.;
                  break;
          case 2: A[i]=1.;
                  B[i]=3./8.;
                  C[i]=2./3.;
                  break;
          case 3: A[i]=1.;
                  B[i]=1./8.;
                  C[i]=1.;
                  break;
          case 4: A[i]=-1.;
                  B[i]=1./12.;
                  C[i]=1.;
                  break;
          case 5: A[i]=1.;
                  B[i]=1./2.;
                  break;
          case 6: A[i]=1./8.;
                  B[i]=1./4.;
                  break;
          case 7: A[i]=3./8.;
                  B[i]=0.;
                  break;
          case 8: A[i]=3./8.;
                  B[i]=1./6.;
                  break;
          case 9: A[i]=1./8.;
                  break;
          default: break;
        }
      }

    };
    void initialize();
    void compute_solution(double T_final_);
    void compute_xx(double t_, double* x_, double* y_, double* theta_, double* outpu
tX, double* outputY, double* output_theta);
    void compute_Gs(double t, double* Gs_x, double* ff_x, double* Gs_y, double* ff_y
, double* Gs_theta, double* ff_theta);
    void compute_y1y1h(double t, double* Gs_x, double* ff_x, double* Gs_y, double* f
f_y, double* Gs_theta, double* ff_theta);
    void dense_output(double t_);
    void hermite(double actual_t, double myTheta, char* filenameDense);
    void nextStep();

    void zap(double* myArray){
      {assert(myArray!=NULL);}
      delete [] myArray;
      myArray = NULL;
    };
    void terminate(){
      zap(x0);
      zap(y0);
      zap(theta0);
      zap(x1);
      zap(y1);
      zap(theta1);
      zap(x1h);
```

```
      zap(y1h);
      zap(theta1h);
      zap(vx0);
      zap(vy0);
      zap(omega0);
      zap(sc_x);
      zap(sc_y);
      zap(sc_theta);
      zap(A);
      zap(B);
      zap(C);
      std::cout<<"Class successfully terminated.\n";
    };

};

#endif
```