

課題（常微分方程式の数値計算：高精度化）解答例

```

/* Runge-Kutta法により、Mass-Spring-Damper系の運動を数値シミュレートする */
/* プログラムの高精度化バージョン */
#include <stdio.h>
#include <stdlib.h>
#include <math.h>

#define NSIZE 10001 /* 配列の要素数の定義：必要に応じて大きく設定すること */

void input(); /* 関数のプロトタイプ宣言 */
void runge4(double *, double *, double *, double);
void f_output(int, double *, double *, double *);

double K, M, B; /* グローバル変数の定義 */
double x0, v0, h0, tol;

void main(void)
{
    int n, nmax, i;
    double t[NSIZE], x[NSIZE], v[NSIZE], tt, xx, vv, h;
    double t2, x2, v2, h2;

    input(); /* パラメータ値の入力 */

    x[0]=x0;
    v[0]=v0;
    t[0]=0.0;
    h=h0;
    printf(" No. Iter. Delta t (h) Time Position Velocity\n");
    /* 解の初期値を画面出力 */
    printf("%5d %5d %12.5e %12.5e %12.5e\n", n=0, i=0, t[0], x[0], v[0]);

    /* 時刻0秒から20秒に達するまで、常微分方程式を数値的に解く */
    for(n=0; t[n]<20.0; n++){
        if(n>=NSIZE-1) break; /* 与えられた配列長を超える場合に計算を終了させる */

        for(i=0; i<100; i++){ /* 誤差が許容値よりも小さくなるまでhを小さくするループ */
            xx=x[n];
            vv=v[n];
            tt=t[n];
            x2=xx;
            v2=vv;
            t2=tt;
            h2=h/2;
            runge4(&xx, &vv, &tt, h); /* 4次精度Runge-Kutta法により時刻がh進む時の解を求める */

            runge4(&x2, &v2, &t2, h2); /* 4次精度Runge-Kutta法により時刻がh/2だけ */
            runge4(&x2, &v2, &t2, h2); /* 2回進む時（合計hだけ進む時）の解を求める */

            if( (fabs(xx-x2)+fabs(vv-v2))<tol ) break; /* ループから出るための誤差の判定 */
            h=h2; /* 時間刻み幅hをh/2の値に設定 */
        }
    }
}

```

```

}
/* 新たに求めた解を画面出力 */
printf("%5d %5d %12.5e %12.5e %12.5e %12.5e\n", n+1, i, h, tt, xx, vv);
/* 時刻がhだけ進んだ時の解(x,v)と時刻tを配列に保存する */
x[n+1]=xx; /* 解 x */
v[n+1]=vv; /* 解 v */
t[n+1]=tt; /* 時刻 t */
if( n%100==99 ) h=h*16; /* 時間を100回進めるごとに、hの値を16倍する */
}
nmax=++n;

f_output(nmax, t, x, v); /* 入力パラメータと数値計算結果をファイル出力 */
}

/* パラメータ値の入力 */
void input()
{
printf("バネ定数K, 減衰定数B, 質量M の値を入力してください。 %n");
printf("K====>"); scanf("%lf", &K);
printf("B====>"); scanf("%lf", &B);
printf("M====>"); scanf("%lf", &M);

printf("初期変位x0, 初速度v0 の値を入力してください。 %n");
printf("x0====>"); scanf("%lf", &x0);
printf("v0====>"); scanf("%lf", &v0);

printf("時間刻みh0, 許容誤差tol の値を入力してください。 %n");
printf("h0====>"); scanf("%lf", &h0);
printf("tol====>"); scanf("%lf", &tol);
}

/* 第nステップ(時刻 t_{n}) の(x,v) から、第n+1ステップ(時刻 t_{n+1}=t_{n}+h) */
/* の(x,v)を4次精度Runge-Kutta法により求める */
void runge4(double *xx, double *vv, double *tt, double h)
{
double kx1,kx2,kx3,kx4,kv1,kv2,kv3,kv4;
double x, v, t;

x=*xx;
v=*vv;
t=*tt;

kx1=h*v;
kv1=h*(-(B/M)*v-(K/M)*x);
kx2=h*(v+0.5*kv1);
kv2=h*(-(B/M)*(v+0.5*kv1)-(K/M)*(x+0.5*kx1));
kx3=h*(v+0.5*kv2);
kv3=h*(-(B/M)*(v+0.5*kv2)-(K/M)*(x+0.5*kx2));
kx4=h*(v+kv3);
kv4=h*(-(B/M)*(v+kv3)-(K/M)*(x+kx3));

*xx=x+(kx1+2.0*kx2+2.0*kx3+kx4)/6.0;

```

```
*vv=v+(kv1+2.0*kv2+2.0*kv3+kv4)/6.0;
*tt=t+h;
}

/* 入力パラメータ値と数値計算結果のファイル出力 */
void f_output(int nmax, double *t, double *x, double *v)
{
    int n;
    FILE *fp;
    char fname[20];

    printf("数値計算結果を保存するファイルの名==>");
    scanf("%s",fname);

    if( (fp=fopen(fname,"w"))==NULL ){
        printf("ファイル %s をオープンできません。¥n",fname);
        exit(1);
    }

    fprintf(fp, "K=%f B=%f M=%f x0=%f v0=%f h0=%f tol=%10.3e¥n",K,B,M,x0,v0,h0,tol);
    fprintf(fp, "    n      Time      Position      Velocity¥n");

    for(n=0; n<nmax; n++)
        fprintf(fp, "%5d %14.7e %14.7e %14.7e¥n", n, t[n], x[n], v[n] );

    fclose(fp);
}
```