

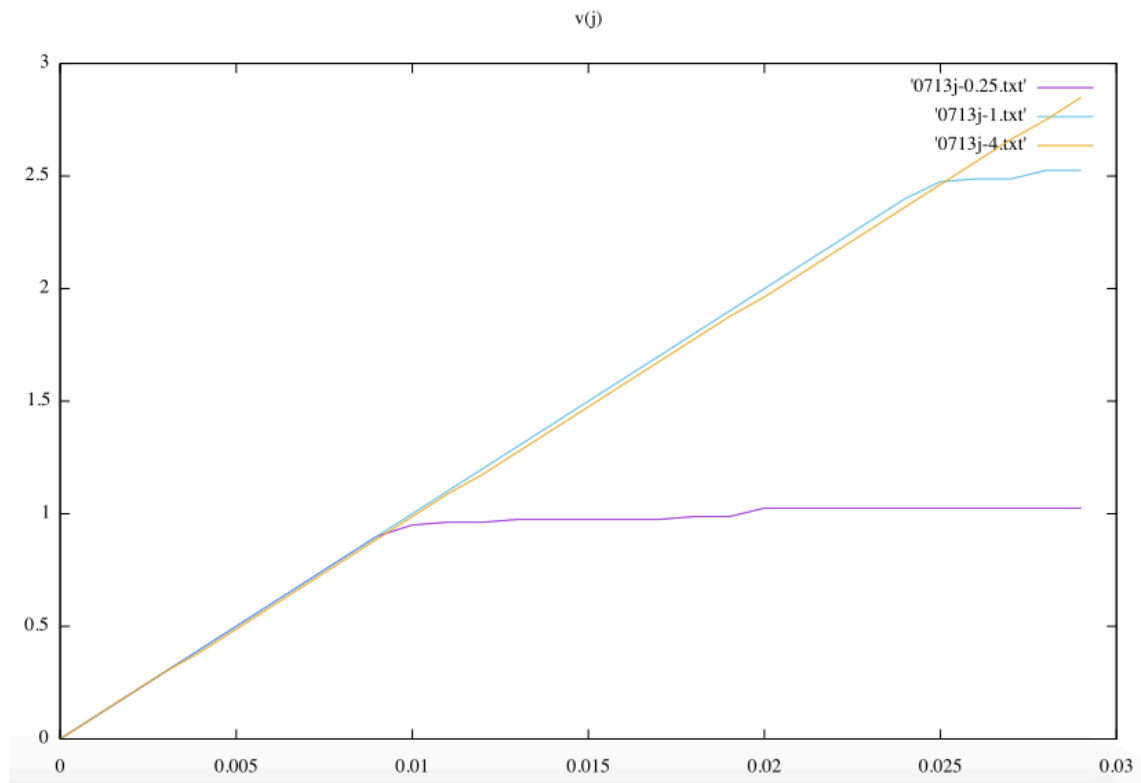
数理基礎レポート

08-153031

木戸口 航

結果

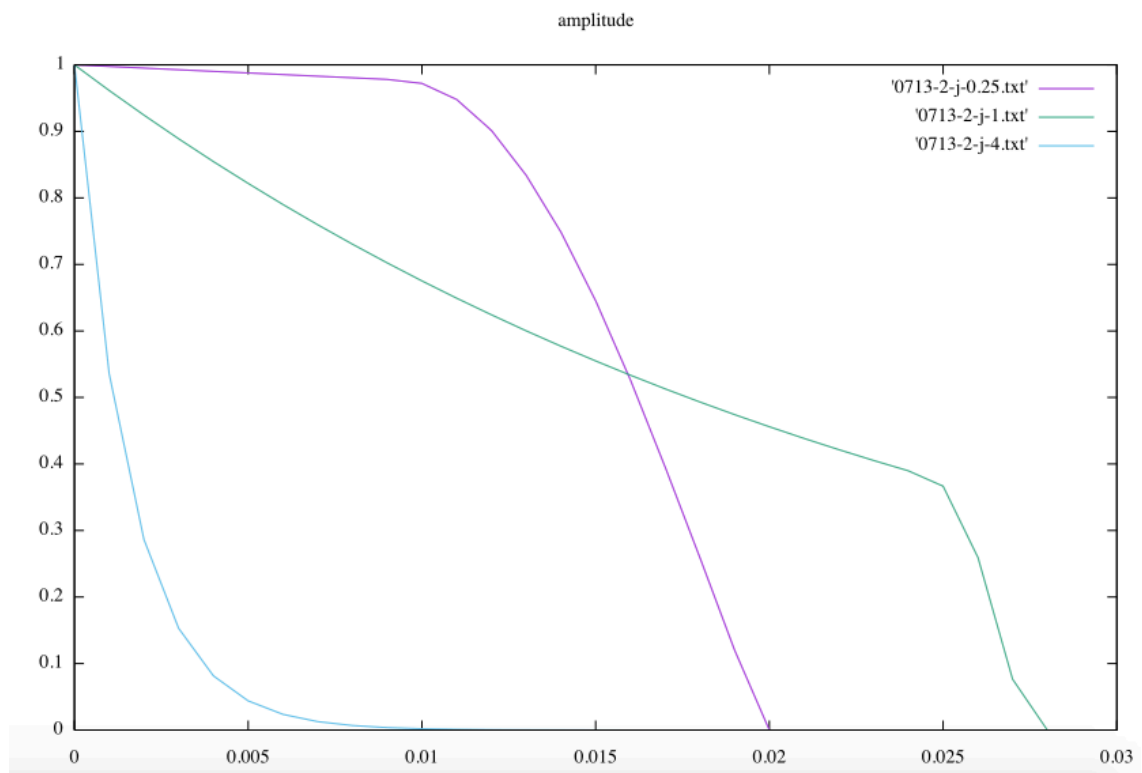
波の進行速度



オレンジ、水色、紫の順に $j = 0.25, 1, 4$ 。

横ばいになるまでは、周波数を変えても速度が変わらないことが見て取れる(おかしい?)

波の振幅の変化



紫、緑、水色の順に $j = 0.25, 1, 4$
周波数が高いほど減衰が早いことがわかる。

ソースコード

```

#include <stdio.h>
#include <math.h>

#define N 80 //変数
#define L 1.0
#define J 4 //変数
#define u 100.0
#define PI 3.1415926535897932384626
#define A 1.0
#define tstep 30000 //変数
#define k 1

int main() {
    double x[3*N+2],
           f[3*N+2],
           fn[3*N+2],
           dx = (double) 1.0/N,
           dt = 0.000001,
           min;
    int i,
        n,
        minI,
        firstMinI;

    for(i=0; i<=3*N+1; i++) {
        x[i] = (double) dx*i;
    }

    //初期条件の設定
    for (i=0; i<=3*N+1; i++) {
        f[i] = A * cos(2*PI*x[i]/L);
    }

    minI = 0;

    //ステップを進めていく
    for (n=0; n<tstep; n++) {
        min = f[minI];
        for (i=minI; i<=3*N+1; i++) {
            if (f[i+1] < f[i]) {
                min = f[i+1];
                minI = i+1;
                if (n==0) {
                    firstMinI = minI;
                }
            } else {
                break;
            }
        }

        for (i=1; i<=3*N; i++) {
            fn[i] = f[i] + (k / (dx*dx)) * (f[i+1] - 2.0*f[i] + f[i-1]) - u / (2*dx) * (f[i+1] - f[i-1])) * dt;
        }
        fn[0] = f[2*N] + (k / (dx*dx)) * (f[2*N+1] - 2.0*f[2*N] + f[2*N-1]) - u / (2*dx) * (f[2*N+1] - f[2*N-1])) * dt; //fn[2L]

        for (i=0; i<=3*N+1; i++) {
            f[i] = fn[i];
        }

        if (n % 1000 == 0) {
            printf("%f %f\n", n*dt, (minI-firstMinI)*dx);
        }
    }
}

```

```

#include <stdio.h>
#include <math.h>

#define N 80 //変数
#define L 1.0
#define j 0.25 //変数
#define u 100.0
#define PI 3.1415926535897932384626
#define A 1.0
#define tstep 30000 //変数
#define k 1

int main() {
    double x[3*N+2],
           f[3*N+2],
           fn[3*N+2],
           dx = (double) 1.0/N,
           dt = 0.000001,
           min;
    int i,
        n,
        minI,
        firstMinI;

    for(i=0; i<=3*N+1; i++) {
        x[i] = (double) dx*i;
    }

    //初期条件の設定
    for (i=0; i<=3*N+1; i++) {
        f[i] = A * cos(2*PI*i*x[i]/L);
    }

    minI = 0;

    //ステップを進めていく
    for (n=0; n<tstep; n++) {
        min = f[minI];
        for (i=minI; i<=3*N+1; i++) {
            if (f[i+1] < f[i]) {
                min = f[i+1];
                minI = i+1;
                if (n==0) {
                    firstMinI = minI;
                }
            } else {
                break;
            }
        }

        for (i=1; i<=3*N; i++) {
            fn[i] = f[i] + (k / (dx*dx)) * (f[i+1] - 2.0*f[i] + f[i-1]) - u / (2*dx) * (f[i+1] - f[i-1]) * dt;
        }
        fn[0] = f[2*N] + (k / (dx*dx)) * (f[2*N+1] - 2.0*f[2*N] + f[2*N-1]) - u / (2*dx) * (f[2*N+1] - f[2*N-1]) * dt; //fn[2L]

        for (i=0; i<=3*N+1; i++) {
            f[i] = fn[i];
        }

        for (i=0; i<=2*N; i++) {
            printf("%f %f\n", i*dx, fn[i]);
        }
    }
}

```