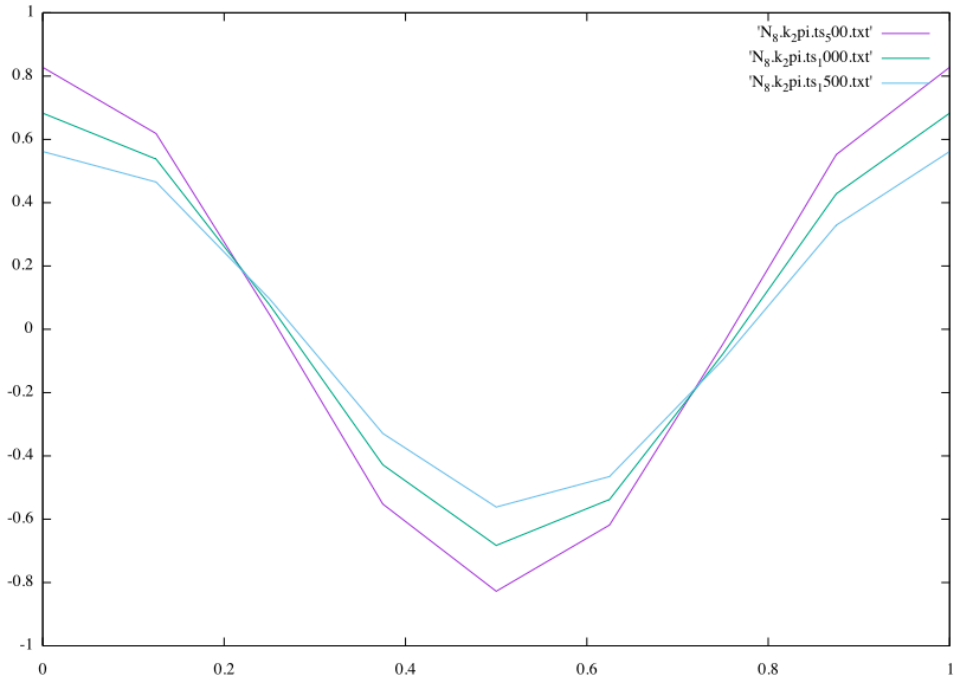
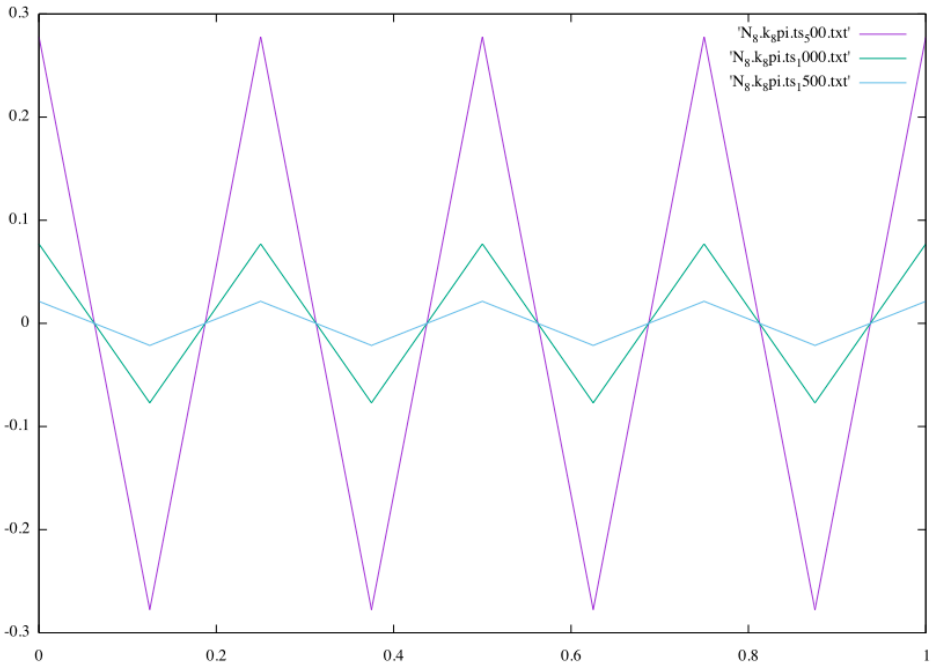
08-153031
教養学部統合自然科学科統合生命コース３年　木戸口　航

結果

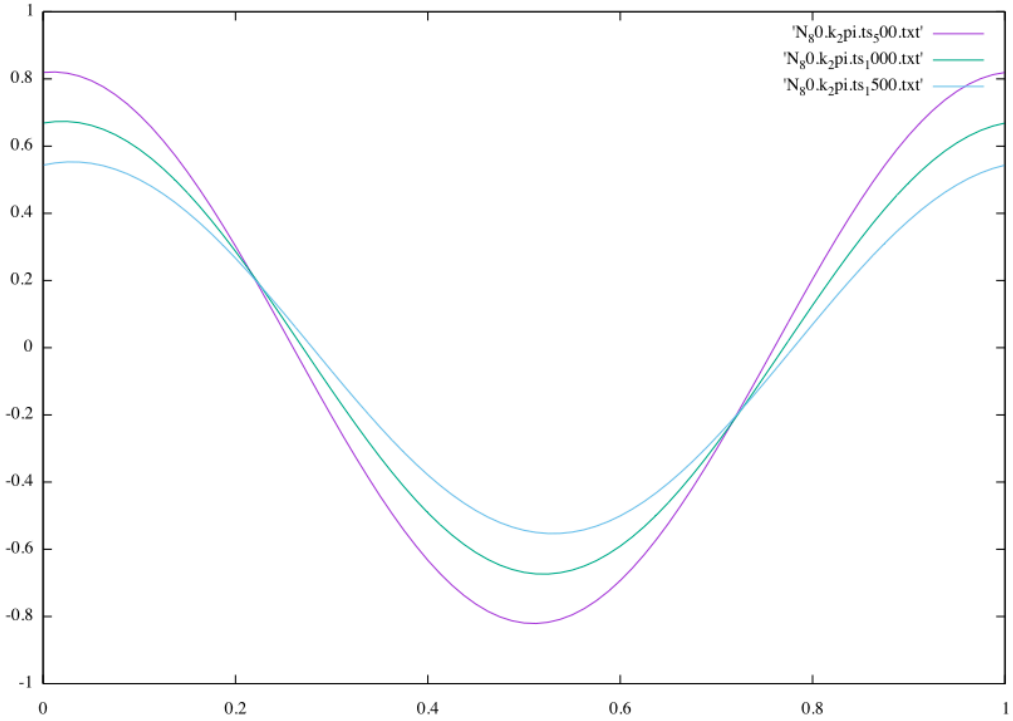
x=0.125, k=2π



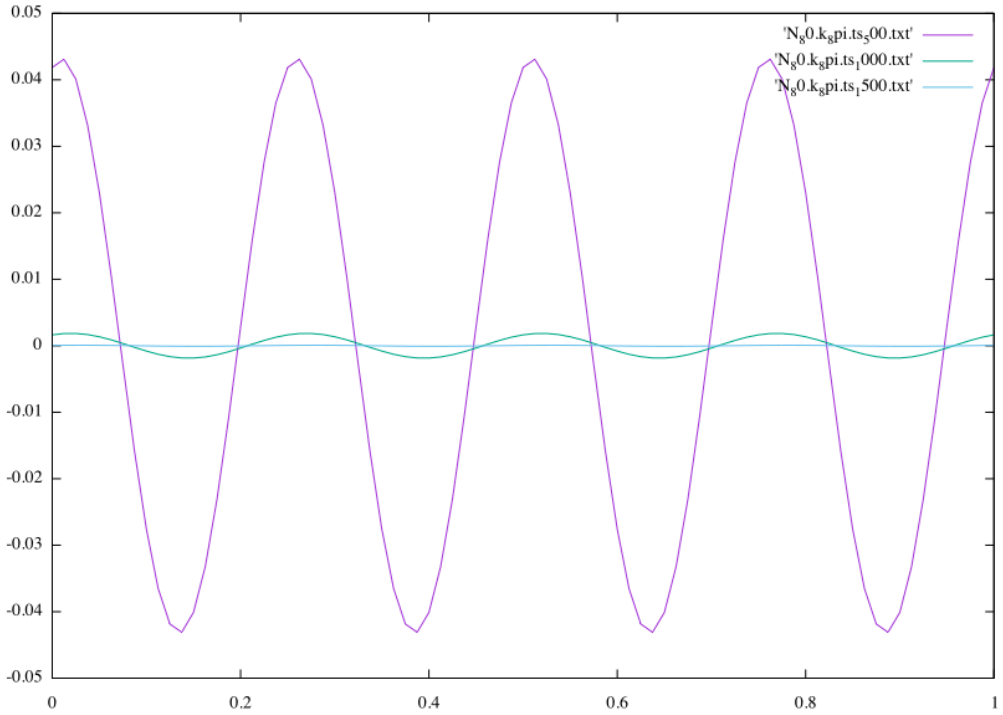
x=0.125, k=8π



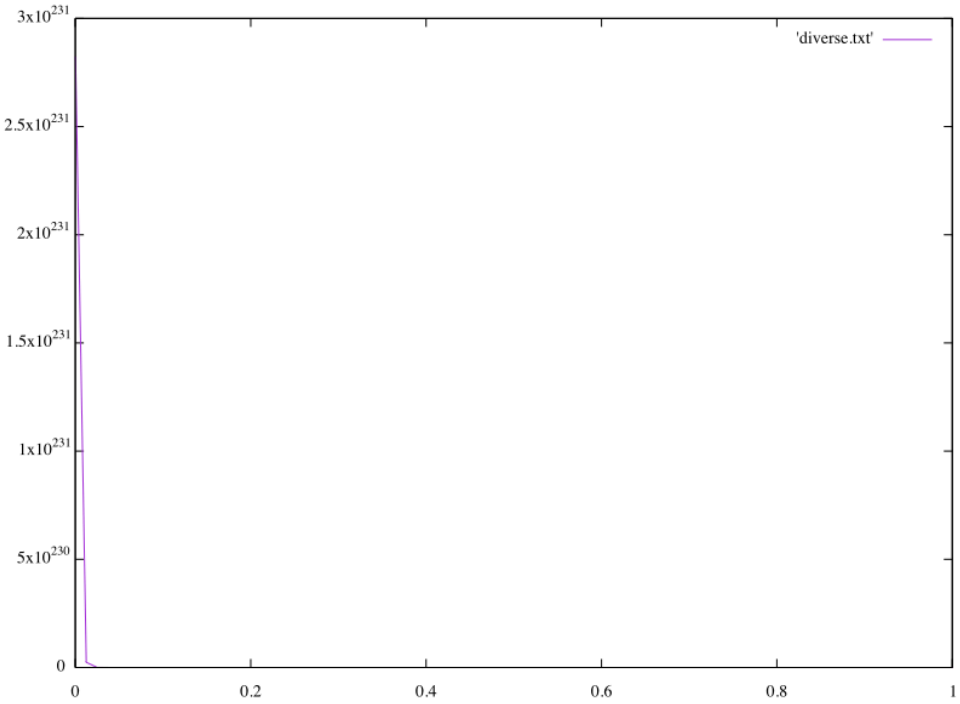
x=0.0125, k=2 π



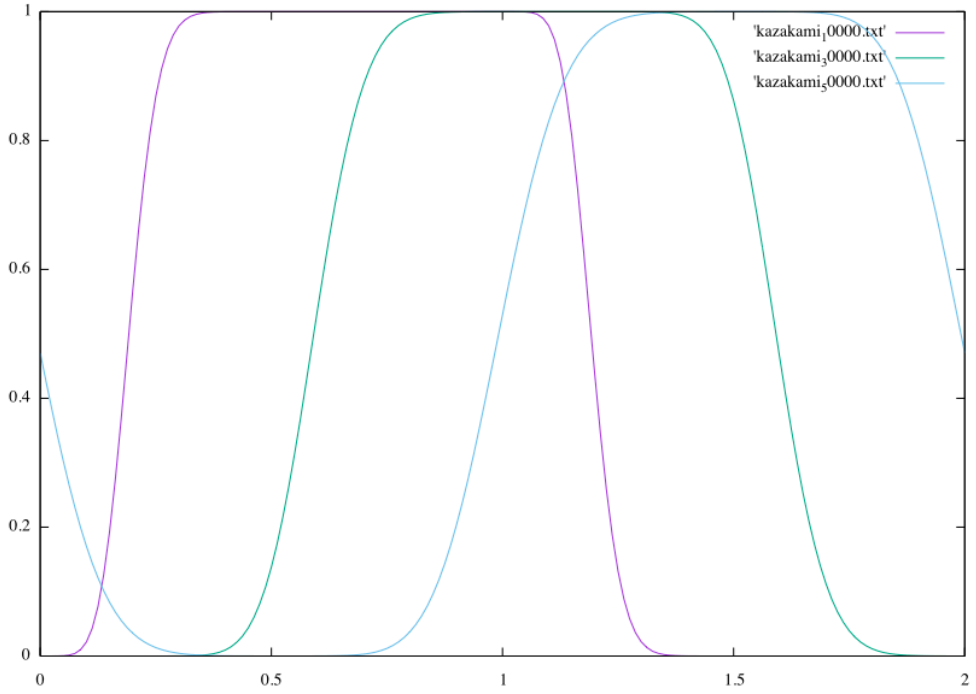
x=0.0125, k=8 π



x=0.0125, k=2 π , κ =0, 中央差分



x=0.0125, k=2 π , κ =0, 風上



考察

メッシュが粗い、または波長が短いと移流が観察しづらい

ソースコード

(1)

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

#define N 80
#define L 1.0
#define j 1.0
#define u 2.0
#define PI 3.14159265358979323384626
#define A 1.0
#define tstep 50000
#define k 0

int main() {
    double  x[3*N+2],
            f[3*N+2],
            fn[3*N+2],
            dx = (double) 1.0/N,
            dt = 0.00001;
    int     i,
            n;


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

    //初期条件の設定
    for (i=0; i<N; i++) {
        f[i] = 1;
    }

    for (i=N; i<2*N-1; i++) {
        f[i] = 0;
    }

    for (i=2*N; i<3*N-1; i++) {
        f[i] = 1;
    }

    //ステップを進めていく
    for (n=0; n<tstep; n++) {

        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 / (dx) * (f[i] - 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", x[i], fn[i]);
    }
}
```

(2)

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

#define N 80
#define L 1.0
#define j 1.0
#define u 2.0
#define PI 3.14159265358979323846
#define A 1.0
#define tstep 500
#define k 0

int main() {
    double  x[3*N+2],
            f[3*N+2],
            fn[3*N+2],
            dx = (double) 1.0/N,
            dt = 0.00001;
    int     i,
            n;


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

    //初期条件の設定
    for (i=0; i<N-1; i++) {
        f[i] = 1;
    }

    for (i=N-1; i<2*N-1; i++) {
        f[i] = 0;
    }
    f[2*N-1] = 1;

    //ステップを進めていく
    for (n=0; n<tstep; n++) {

        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<=N; i++) {
        printf("%f %f\n", x[i], fn[i]);
    }
}
```

(3)

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

#define N 80  //変数
#define L 1.0
#define j 4.0
#define u 2.0  //変数
#define PI 3.14159265358979323846
#define A 1.0
#define tstep 1500  //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 = 0.00001;
    int     i,
            n;


    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*j*x[i]/L);
    }

    //ステップを進めていく
    for (n=0; n<tstep; n++) {

        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<=N; i++) {
        printf("%f %f\n", x[i], fn[i]);
    }
}
```