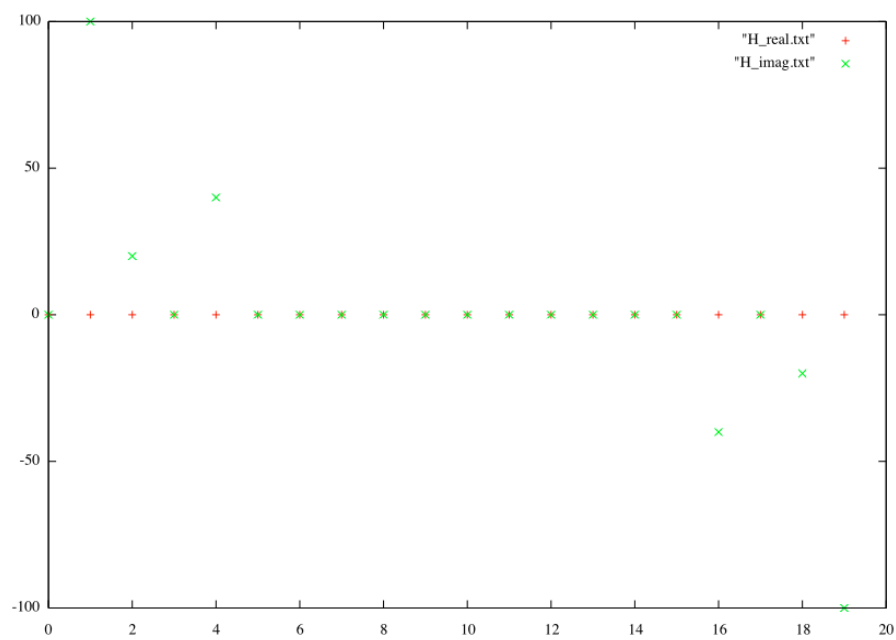


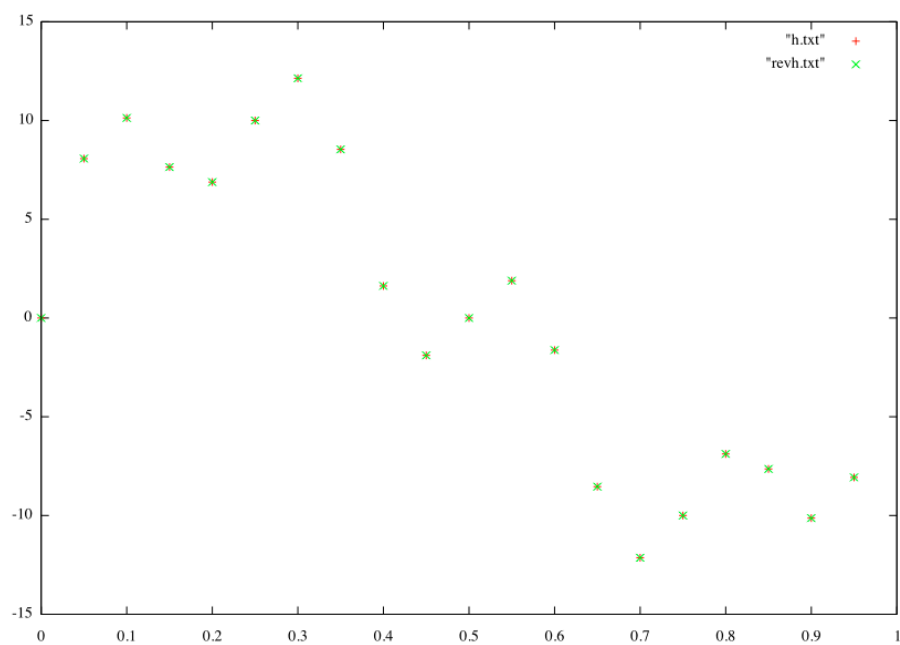
## 結果

①

フーリエ変換

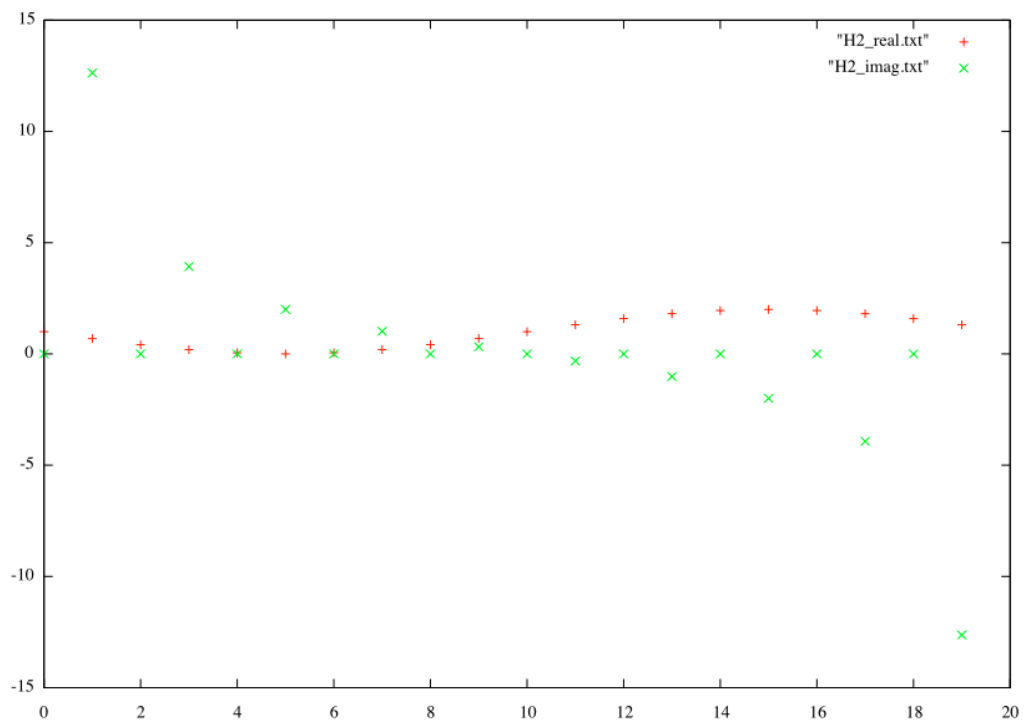


逆フーリエ変換との比較

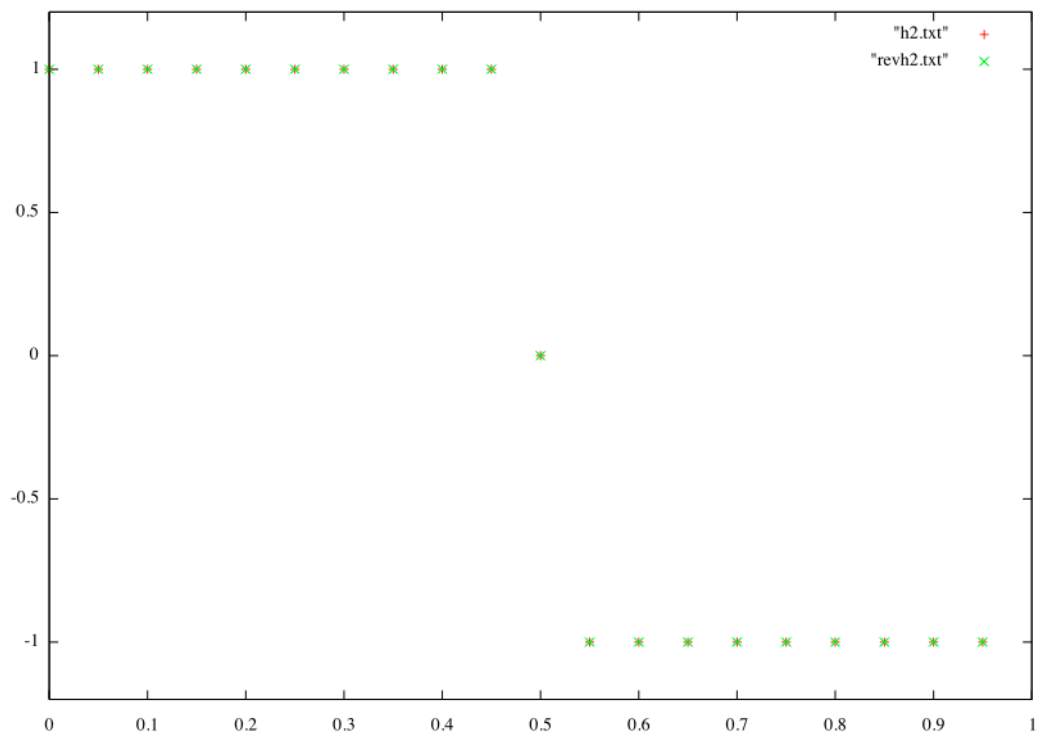


②

フーリエ変換



逆フーリエ変換との比較



## ソースコード

①

```
#include <stdio.h>
#include <math.h>
#include <complex.h>
#define PI 3.141592653589793238

double f (double t) {
    double result = 10 * sin(2*PI*t) + 2 * sin(4*PI*t) + 4 * sin(8*PI*t);
    return result;
}

int main(void) {
    double N = 20;
    double time = 1;
    double dlt = time / N;

    double t = 0;
    double n = 0;

    double _Complex c[20] = {0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0};
    double _Complex rev[20] = {0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0};

    double _Complex tmp = cexp(I*2*PI/N);

    for (int i=0; i<N; i++){
        for (int k=0; k<N; k++) {
            c[i] += f(t) * cpow(tmp, i*k);
            t += dlt;
        }
    }

    for (int i=0; i<N; i++) {
        for (int k=0; k<N; k++) {
            rev[i] += c[k] * cpow(tmp, -i*k);
        }
        rev[i] /= 20;
    }

    //元の関数
    t = 0;
    for (int i=0; i<N; i++) {
        printf("%f %f\n", t, f(t));
        t += dlt;
    }

    //フーリエ変換
    for (int i=0; i<N; i++)
    {
        printf("%d %f\n", i, creal(c[i]));
    }
    printf("\n");

    for (int i=0; i<N; i++)
    {
        printf("%d %f\n", i, cimag(c[i]));
    }
    printf("\n");

    //フーリエ逆変換
    t = 0;
    for (int i=0; i<N; i++) {
        printf("%f %f\n", t, creal(rev[i]));
        t += dlt;
    }
}
```

②

```
#include <stdio.h>
#include <math.h>
#include <complex.h>
#define PI 3.141592653589793238

double f (int i) {
    if (0 <= i && i < 10) {
        return 1;
    } else if (i == 1) {
        return 0;
    } else if (i > 10) {
        return -1;
    }
}

int main(void) {
    double N = 20;
    double time = 1;
    double dlt = time / N;
    double t = 0;
    double n = 0;

    double _Complex c[20] = {0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0};
    double _Complex rev[20] = {0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0};

    double _Complex tmp = cexp(I*2*PI/N);

    for (int i=0; i<N; i++){
        for (int k=0; k<N; k++) {
            c[i] += f(k) * cpow(tmp, i*k);
        }
    }

    for (int i=0; i<N; i++) {
        for (int k=0; k<N; k++) {
            rev[i] += c[k] * cpow(tmp, -i*k);
        }
        rev[i] /= 20;
    }

    //元の関数
    t = 0;
    for (int i=0; i<N; i++) {
        printf("%f %f\n", t, f(i));
        t += dlt;
    }

    フーリエ変換
    for (int i=0; i<N; i++)
    {
        printf("%d %f\n", i, creal(c[i]));
    }
    printf("\n");

    for (int i=0; i<N; i++)
    {
        printf("%d %f\n", i, cimag(c[i]));
    }
    printf("\n");

    フーリエ逆変換
    t = 0;
    for (int i=0; i<N; i++) {
        printf("%f %f\n", t, creal(rev[i]));
        t += dlt;
    }
}
```