

```

1 #include <stdio.h>
2 #include <stdlib.h>
3 #include <math.h>
4 #include <time.h>
5
6 #define twoPI 3.1415926535 * 2
7 #define N 512
8 #define OPT 0 /* OPT = 1 光学的DFT (直流分が中央) */
9 /* OPT = 0 通常のDFT (直流分が左端) */
10 FILE *stream;
11 FILE *fopen();
12
13 int fft1 (double a_rl[], double a_im[], int ex, int inv);
14 void fft1core (double a_rl[], double a_im[], int length,
15 int ex, double sin_tbl[], double cos_tbl[], double buf[]);
16 void cstb (int length, int inv, double sin_tbl[], double cos_tbl[]);
17 void birv (double a[], int length, int ex, double b[]);
18
19 void main ( void )
20 {
21 int t, i;
22 double dataR[N], dataI[N];
23 double start, finish, etime;
24
25 // データの生成
26 for (t=0; t<N; t++) {
27 dataR[t] = 0.5 + 1.0 * cos (2*twoPI*t/N)
28 - 1.0 * cos (3*twoPI*t/N)
29 + 2.0 * sin (4*twoPI*t/N);
30 dataI[t] = 0.0;
31 }
32
33 start=clock();
34 fft1(dataR, dataI, 9, 1);
35 finish=clock();
36
37 stream = fopen( "fft1-spread.txt", "w" );
38
39 for (i=0; i<N; i++) {
40 fprintf( stream, "%4d %14.6f %14.6f %14.6f %n", i,
41 dataR[i], dataI[i], dataR[i]*dataR[i]+dataI[i]*dataI[i] );
42 }
43 etime=(double) (finish-start)/CLOCKS_PER_SEC;
44 printf("Elapse time: %f seconds!\n", etime);
45 printf("Result written to 'fft1-spread.txt.'\n");
46 fclose( stream );
47 }
48
49 /*--- fft1 --- 1次元FFTの実行 -----*/
50 a_rl: データ実数部 (入出力兼用)
51 a_im: データ虚数部 (入出力兼用)
52 ex: データ個数を2のべき乗で与える(データ個数 = 2のex乗個)
53 inv: 1: DFT, -1: 逆DFT
54 -----*/
55 int fft1(double a_rl[], double a_im[], int ex, int inv)
56 {
57 int i, length = 1;
58 double *sin_tbl; /* SIN計算用テーブル */
59 double *cos_tbl; /* COS計算用テーブル */
60 double *buf; /* 作業用バッファ */
61
62 for (i = 0; i < ex; i++) length *= 2; /* データ個数の計算 */
63 /* printf("%d %n", length); */
64
65 sin_tbl = (double *)malloc((size_t) length*sizeof(double));
66 cos_tbl = (double *)malloc((size_t) length*sizeof(double));

```

```

67  buf = (double *)malloc((size_t)length*sizeof(double));
68  if ((sin_tbl == NULL) || (cos_tbl == NULL) || (buf == NULL)) {
69      return(-1);
70  }
71
72  cstb(length, inv, sin_tbl, cos_tbl); /* SIN,COSテーブル作成 */
73  fft1core(a_rl, a_im, length, ex, sin_tbl, cos_tbl, buf);
74
75  free(sin_tbl);
76  free(cos_tbl);
77  return 0;
78 }
79
80 /*--- fft1core --- 1次元FFTの計算の核になる部分 -----*/
81  a_rl:   データ実数部 (入出力兼用)
82  a_im:   データ虚数部 (入出力兼用)
83  ex:     データ個数を2のべき乗で与える(データ個数 = 2のex乗個)
84  sin_tbl: SIN計算用テーブル
85  cos_tbl: COS計算用テーブル
86 -----*/
87 void fft1core(double a_rl[], double a_im[], int length,
88  int ex, double sin_tbl[], double cos_tbl[], double buf[])
89 {
90     int    i, j, k, w, j1, j2;
91     int    numb, lenb, timb;
92     double  xr, xi, yr, yi, nrml;
93
94     if (OPT == 1) {
95         for (i = 1; i < length; i+=2) {
96             a_rl[i] = -a_rl[i];
97             a_im[i] = -a_im[i];
98         }
99     }
100    numb = 1;
101    lenb = length;
102    for (i = 0; i < ex; i++) {
103        lenb /= 2;
104        timb = 0;
105        for (j = 0; j < numb; j++) {
106            w = 0;
107            for (k = 0; k < lenb; k++) {
108                j1 = timb + k;
109                j2 = j1 + lenb;
110                xr = a_rl[j1];
111                xi = a_im[j1];
112                yr = a_rl[j2];
113                yi = a_im[j2];
114                a_rl[j1] = xr + yr;
115                a_im[j1] = xi + yi;
116                xr = xr - yr;
117                xi = xi - yi;
118                a_rl[j2] = xr*cos_tbl[w] - xi*sin_tbl[w];
119                a_im[j2] = xr*sin_tbl[w] + xi*cos_tbl[w];
120                w += numb;
121            }
122            timb += (2*lenb);
123        }
124        numb *= 2;
125    }
126    birv(a_rl, length, ex, buf); /* 実数データの並べ換え */
127    birv(a_im, length, ex, buf); /* 虚数データの並べ換え */
128    if (OPT == 1) {
129        for (i = 1; i < length; i+=2) {
130            a_rl[i] = -a_rl[i];
131            a_im[i] = -a_im[i];
132        }

```

```

133     }
134     nrml = (double)(1.0 / sqrt((double) length));
135     for (i = 0; i < length; i++) {
136         a_rl[i] *= nrml;
137         a_im[i] *= nrml;
138     }
139 }
140
141 /*--- cstb --- SIN, COSテーブル作成 -----*/
142     length:   データ個数
143     inv:      1: DFT, -1: 逆DFT
144     sin_tbl:  SIN計算用テーブル
145     cos_tbl:  COS計算用テーブル
146 -----*/
147 void cstb(int length, int inv, double sin_tbl[], double cos_tbl[])
148 {
149     int i;
150     double xx, arg;
151
152     xx = (double)(-twoPI / (double) length);
153     if (inv < 0) xx = -xx;
154     for (i = 0; i < length; i++) {
155         arg = (double)i * xx;
156         sin_tbl[i] = (double)sin(arg);
157         cos_tbl[i] = (double)cos(arg);
158     }
159 }
160
161 /*--- birv --- データの並べ換え -----*/
162     a:   データの配列
163     length: データ個数
164     ex:   データ個数を2のべき乗で与える(length = 2のex乗個)
165     b:   作業用バッファ
166 -----*/
167 void birv(double a[], int length, int ex, double b[])
168 {
169     int i, ii, k, bit;
170
171     for (i = 0; i < length; i++) {
172         for (k = 0, ii=i, bit=0; ; bit<<=1, ii>>=1) {
173             bit = (ii & 1) | bit;
174             if (++k == ex) break;
175         }
176         b[i] = a[bit];
177     }
178     for (i = 0; i < length; i++)
179         a[i] = b[i];
180 }

```