

russ_fft

```

/* russ_fft.c 6-jun-05 */
/* from jihn russ fortran code */
/* transform complex data in f, overwrite with F */

void russ_fft(float f[], int N)
{
int i,j,k,l;
int iln,nv2,nml,le,le1,ip;
float ln,pi;
float t[2],u[2],w[2];
float re1,im1,re2,im2,prodr,prodi;

pi=3.14159265358979;
ln=log((float) N)/log(2.0);
iln=(int) (ln+0.5);
nv2=(int) (N/2);
nml=N-1;
j=1;
for(i=1; i<=nml; i++)
{
if(i < j)
{
t[0]=f[j*2 + 0];
t[1]=f[j*2 + 1];
f[j*2 + 0]=f[i*2 + 0];
f[j*2 + 1]=f[i*2 + 1];
f[i*2 + 0]=t[0];
f[i*2 + 1]=t[1];
}
k=nv2;
while(k < j)
{
j=j-k;
k=(int) (k/2);
}
j=j+k;
}
for(l=1; l<=iln; l++)
{
le=pow(2.0,l);
le1=(int) (le/2);
u[0]=1.0;
u[1]=0.0;
w[0]=cos(pi/(float) le1);
w[1]= -sin(pi/(float) le1);
for(j=1; j<=le1; j++)
{
for(i=j; i<=N; i=i+(int) le)
{
ip=i+le1;
/* cplx mutl t=f(ip)*u */
re1=f[ip*2 + 0];
im1=f[ip*2 + 1];
re2=u[0];
im2=u[1];
prodr=re1*re2-im1*im2;
prodi=re1*im2+im1*re2;
t[0]=prodr;
t[1]=prodi;

f[ip*2 + 0]=f[i*2 + 0] - t[0];
f[ip*2 + 1]=f[i*2 + 1] - t[1];
}
}
}
}

```

```

                                russ_fft
    f[i*2 + 0]=f[i*2 + 0] + t[0];
    f[i*2 + 1]=f[i*2 + 1] + t[1];
    }
    /* cplx mutl u=u*w */
    re1=u[0];
    im1=u[1];
    re2=w[0];
    im2=w[1];
    prodr=re1*re2-im1*im2;
    prodi=re1*im2+im1*re2;
    u[0]=prodr;
    u[1]=prodi;
    }
}

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

```