

```
/* FD1D_1.1.c. 1D FDTD simulation in free space */
```

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

```
#include <stdlib.h>
```

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

```
#define KE 200
```

```
main ()
```

```
{
```

```
float ex [KE], hy [KE];
```

```
int n, k, kc, ke, NSTEPS;
```

```
float T;
```

```
float t0, spread, pulse;
```

```
FILE *fp, *fopen();
```

```
/* Initialize */
```

```
for (k=1; k < KE; k++)
```

```
{   ex [k] = 0;
```

```
    hy [k]= 0. }
```

```
kc = KE/2;
```

```
t0 = 40.0;
```

```
spread = 12;
```

```
T = 0;
```

```
NSTEPS = 1;
```

```
while (NSTEPS > 0 ) {
```

```
printf("NSTEPS - ->");
```

```
scanf("%d", &NSTEPS) ;
```

```
printf("%d"\n", NSTEPS) ;
```

```
n = 0;
```

```
for (n=1; n <=NSTEPS ; n++)
```

```
{
```

```
T = T + 1;
```

```
/* Main FDTD Loop */
```

```
/* Center of the problem space */
```

```

/* Center of the incident pulse */
/* Width of the incident pulse */
/*KE is the number of cells to be used */
/* Calculate the Ex field */
For ( k=1; k < KE; k++ )
{ ex [k] = ex [k] + .5*( hy [k-1] - hy [k] ) ; }
/* Put a Gaussian pulse in the middle */
/* NSTEPS is the number of times the */
/* main loop has executed */
/* T keeps track of the total number */
/* of times the main loop is executed */

Pulse = exp(-0.5* (pow ( (t0-T) /spread, 2.0) ) );
Ex[kc] = pulse;
printf("%5.1f %6.2f\n",t0-T, ex [kc]);
/* Calculate the Hy field */
For (k=0; k < KE-1; k++ )
{ hy [k]
}
/* End of the Main FDTD Loop */
/* At the end of the calculation, print out the Ex and Hy fields */
For (k=1; k <= KE; k++ )
{ printf("%3d  %6.2f  %6.2f\n", k, ex [k], hy [k]);
/* Write the E field out to a file "Ex" */
Fp fopen("Ex", "w");
for ( k=1; k <= KE; k++ )
{ fprintf (fp," %6.2 f \n", ex [k]); }
Fclose(fb);
/* Write the H field out to a file "Hy" */
Fb=fopen( "Hy", "w");
for ( k=1; k <= KE; k++ )

```

```
{ fprintf (fp," %6.2 f \n", hy [k]); }  
fclose(fp);  
printf ( "T = %5.0f\n",T) ;  
}  
}
```