

Travaux Pratiques - Langage C - Option
--

## 1 modif 1

```
244a245,246
>         if(phase<0){phase+=M_PI;}
>         else if(phase>2){phase-=M_PI;}
```

## 2 modif 2

```
108c108
<         gain( champ );
---
>         // gain( champ );
110a111,112
>         normalisation(champ);
>
115c117
<         pertes( champ , R2 );
---
>         // pertes( champ , R2 );
116a119,120
>
>         normalisation(champ);
118c122
<         gain( champ );
---
>         // gain( champ );
```

## 3 modif 3

```
169c169
<         z = gsl_complex_rect(0.01*gsl_rng_uniform( r ),0.01*gsl_rng_uniform( r ));
---
>         z = gsl_complex_rect(0.01*(gsl_rng_uniform( r )-0.5),0.01*(gsl_rng_uniform( r )-0.5));
```

## 4 modif 4

```
49a50
> double FWHM( gsl_matrix* matrice );
63c64
```

```

< double DeltaX1 = 2;
---
> double DeltaX1 = 20;
108c109
< // gain( champ );
---
> gain( champ );
111c112
< normalisation(champ);
---
> // normalisation(champ);
117c118
< // pertes( champ , R2 );
---
> pertes( champ , R2 );
120c121
< normalisation(champ);
---
> // normalisation(champ);
122c123
< // gain( champ );
---
> gain( champ );
129a131,134
> /* détermination de la largeur à mi-hauteur de la dernière colonne de module1 */
>
> printf("largeur à mi-hauteur : %.3f mm\n" , FWHM( module1 ) * dx1);
> printf("largeur à mi-hauteur : %.3f mm\n" , FWHM( module2 ) * dx2);
378a384,418
> /*****
> * Largeur à mi hauteur *
> *****/
>
> double FWHM( gsl_matrix* matrice )
> {
> int i;
> int point1=0, point2=0;
> double max;
> gsl_vector *champ = gsl_vector_calloc( N );
>
> /* on place la dernière intération dans le vecteur champ */
>
> for(i=0;i<N;i++)
> gsl_vector_set( champ , i , gsl_matrix_get( matrice , i , ITERATIONS-1 ));
>
> /* puis on recherche le maximum de champ */
>
> max = gsl_vector_max( champ );
>
> /* recherche du premier point à mi-hauteur */
>
> while(gsl_vector_get( champ , point1) < max/2.0)
> point1++;
>
> /* recherche du second point à mi-hauteur */

```

```

>
> point2 = point1+1;
>
> while(gsl_vector_get( champ , point2) > max/2.0)
>     point2++;
>
> return (point2-point1);
> }
>

```

## 5 modif 5

```

30c30
< #define ITERATIONS 60
---
> #define ITERATIONS 30
64c64
< double DeltaX1 = 20;
---
> double DeltaX1 = 2;
131,135d130
< /* détermination de la largeur à mi-hauteur de la dernière colonne de module1 */
<
< printf("largeur à mi-hauteur : %.3f mm\n" , FWHM( module1 ) * dx1);
< printf("largeur à mi-hauteur : %.3f mm\n" , FWHM( module2 ) * dx2);
<
363c358,362
< for(i=1+0.5*(N-1.4*ceil(sqrt(N)));i<0.5*(N+1.4*ceil(sqrt(N)));i++){
---
> for(i=1+ceil(0.43*N);i<1+ceil(0.46*N);i++){
>     z = gsl_complex_polar(1.0,0.0);
>     gsl_vector_complex_set(filtrel,i,z);
> }
> for(i=1+ceil(0.54*N);i<1+ceil(0.57*N);i++){
378c377
< for(i=1+0.5*(N-1.4*ceil(sqrt(N)));i<0.5*(N+1.4*ceil(sqrt(N)));i++){
---
> for(i=0;i<N;i++){
381a381,390
> for(i=0;i<2;i++){
>     z = gsl_complex_polar(0,0.0);
>     gsl_vector_complex_set(filtre2,i,z);
> }
> for(i=N-2;i<N;i++){
>     z = gsl_complex_polar(0,0.0);
>     gsl_vector_complex_set(filtre2,i,z);
> }
> z = gsl_complex_polar(0,0.0);
>     gsl_vector_complex_set(filtre2,N/2,z);

```