

```

//Programme qui integre une equation de schrodinger
//dans un puit 1D quelconque (etats liés)

#include<stdio.h>
#include<math.h>

#define nx_max 1000    //nombre de points
#define nn_max 2       //nombre d'etats a calculer

int main()
{
    FILE *fich; //fonctions d'onde et potentiel
    FILE *fcom; //commandes gnuplot
    fich=fopen("fct_onde.dat","w");
    fcom=fopen("graph.gnu","w");

    int i,j,k,node;
    int n; //indice le nombre quantique principal
    double dx, x[nx_max],x_min,x_max; //pas d'integration et liste des points

    double u[nx_max],psi[nx_max][nn_max],sum; //fonction d'onde pour chaque n
    double E[nn_max], E_test,E_min,E_max; //spectre du systeme et Énergie
d'essai
    double V[nx_max]; //potentiel
    double psi_min,psi_max;

    //-----
    //          Initialisation des variables
    //-----

    x_min=-1.5;
    x_max=1.5;
    dx=(x_max-x_min)/nx_max;

    //-----
    //          Definition du potentiel
    //-----

    for(i=0;i<nx_max;i++)
    {
        x[i]=x_min+dx*i;

        //Puits lisse
        //if (x[i]>=-2.5 && x[i]<=2.5) {
        //    V[i]=4-(4/6.25)*x[i]*x[i];
        //}

        //Marche de potentiel
        //if (x[i]>=0) {
        //    V[i]=7;}

        //Barriere de potentiel
        if (x[i]>=-1 && x[i]<=1) {

            V[i]=15;}

        //double puit avec barriere gaussienne
        //V[i]=-26+x[i]*x[i]+50*exp(-10*x[i]*x[i]);

        //puits quartique

```

```

//V[i]=-10+0.4*x[i]*x[i]*(-5+0.3*x[i]*x[i]);

//double puit carre
//if((i>nx_max/2.5)&&(i<2*nx_max/3.5)) V[i]=-1;
//else V[i]=-10;

//Oscillateur harmonique inversé
// if((i<750)&&(i>250)) V[i]=-9.1*(1+(x[i]*x[i])/0.5625);
// else V[i]=-10;

//puit harmonique
//V[i]=0.5*x[i]*x[i];

//puit infini
//if((i==0)|| (i==nx_max-1)) V[i]=100;
//else V[i]=0;

//puit carre
//if((i>nx_max/3)&&(i<2*nx_max/3)) V[i]=-100;
//else V[i]=-0.0001;

//puit gaussien
//V[i]=-10*exp(-0.2*x[i]*x[i]);

//puit atroce
//V[i]=-x[i]*(1-x[i])*sin(x[i])+exp(-cosh(x[i]));

//puit bizarre
//V[i]=(sin(x[i])/x[i])*(sin(x[i])/x[i]);

//puit quasiperiodique
//[i]=cos(20*x[i])+cos(20*0.5*(sqrt(5)-1)*x[i]);

}

//-----
//          Integration de schrodinger
//-----

for(n=0;n<nn_max;n++) //boucle sur les nombres quantiques
{
    //L'energie est trouvée par bisection

    E_min=-50;
    E_max=50;

    for(k=0;k<400;k++) //boucle sur l'Énergie
    {
        //integration

        E_test=0.5*(E_min+E_max);
        u[0]=0.01; //point de départ
        u[1]=0.02;

        node=0;

        for(i=1;i<nx_max-1;i++)
        {
            u[i+1]=2*(1+dx*dx*(V[i]-E_test))*u[i]-u[i-1];
            if(u[i+1]*u[i]<0) node=node+1;
        }
    }
}

```

```

    }
    printf("Nombre de noeux it %d %d\n",k,node);
    if(node>n) E_max=E_test;
    else E_min=E_test;
}

//normalisation de la fonction d'onde
sum=0;
for(i=0;i<nx_max;i++)
{
    sum=sum+u[i]*u[i]*dx;
}
sum=sqrt(sum);

//memorisation de psi_n
for(i=0;i<nx_max;i++)
{
    psi[i][n]=u[i]/sum;
}

//memorisation du spectre
E[n]=E_test;
}

//ecriture des fct d'onde dans le fichier

psi_min=psi[0][0]+E_max;
psi_max=psi[0][0]+E_min;

for(i=0;i<nx_max;i++)
{
    fprintf(fich,"%lg ",x[i]);
    fprintf(fich,"%lg ",V[i]);
    for(n=0;n<nn_max;n++)
    {
        if((psi[i][n]+E[n])>psi_max) psi_max=psi[i][n]+E[n];
        if((psi[i][n]+E[n])<psi_min) psi_min=psi[i][n]+E[n];
        fprintf(fich,"%lg %lg ",psi[i][n]+E[n],E[n]);
    }
    fprintf(fich,"\n");
}

for(n=0;n<nn_max;n++)
{
    printf("energie n %d %lg\n",n,E[n]);
    printf("ecart: %lf\n",E[n]-E[n-1]);
}

//fichier gnuplot
fprintf(fcom,"set nokey\n");
fprintf(fcom,"set yrange[%lg:%lg]\n",psi_min,psi_max);
fprintf(fcom,"set multiplot\n");
fprintf(fcom,"plot 'fct_onde.dat' using 1:2 w l lt 2.\n");
for(n=0;n<nn_max;n++)
{
    fprintf(fcom,"plot 'fct_onde.dat' using 1:%d w l \n",2*n+3);
    fprintf(fcom,"plot 'fct_onde.dat' using 1:%d w l \n",2*n+4);
}
fprintf(fcom,"set nomultiplot\n");
fprintf(fcom,"pause -1\n");

```

```
fclose(fich);fclose(fcom);  
return(0);  
}
```