Multiplications Rapides

3 décembre 2012

Résumé

Il s'agit d'implanter les algorithmes du cours concernant la multiplication des polynômes à coeffficients complexes.

1 Complexe

Le type `complex` a fait son apparition dans C99, il est maintenant disponible sous `gcc`. Le programme (1) est une nouvelle façon d'imprimer zéro. Expliquer pourquoi !

```
COMPLEX(7)                  Linux Programmer's Manual
NAME
       complex - basics of complex mathematics
SYNOPSIS
       #include <complex.h>
DESCRIPTION
  Complex numbers are numbers of the form z = a+b*i, where
a and b are real numbers and i = sqrt(-1), so that i*i = -1.
There are other ways to represent that number. The pair (a,b)
of real numbers may be viewed as a point in the plane, given
by X and Y-coordinates. This same point may also be described
by giving the pair of real numbers (r,phi), where r is the
distance to the origin O, and phi the angle between the X-axis
and the line Oz. Now z=r*exp(i*phi)=r*(cos(phi)+i*sin(phi)).
The basic ...

SEE ALSO
  cabs, carg, cexp, cpow, creal, csqrt, cimg
```

2 Karatsuba

On représente les polynômes par des pointeurs sur des `complex` :

```
typedef double complex scalar;
typedef  scalar * poly;
```

```c
#include <complex.h>
#include <math.h>
#include <stdlib.h>
#include <stdio.h>
int main( int argc, char *argv[] )
{
double complex x, zeta, s;
int i, n = atoi( argv[1] );
zeta =  cexp( 2 * I * M_PI / n );
s =  0;
x =  1;
for( i = 0; i < n; i++){
   s = s + x;
   x = x * zeta;
}
printf("%f + %f * i\n", creal(s), cimag(s));
return 0;
}
```

FIG. 1 – Le type complexe en C

1. implantez l'algorithme de multiplication de Anatolii Alexevich Karatsuba pour écrire une fonction `poly kara( a, b, n)` qui calcule le produit de deux polynômes de taille n.
2. Assurez vous du bon fonctionnement de votre multiplication au moyen d'un test efficace.
3. Précisez le temps de calcul en fonction de n.

3 Fourier

1. Implanter `Fourier( p, n )` qui calcule la transformée de Fourier de p sur place.
2. Implanter l'inversion de Fourier.
3. Vérifier la correction.
4. Préciser le temps de calcul.

4 Multiplication

1. Implanter `tfrmul( r, p, n)` qui calcule $r := rp \pmod{X^n - 1}$ le produit de 2 polynomes modulo $X^n - 1$.
2. Prouver le bon fonctionnement de votre multiplication.
3. Préciser le temps de calcul en fonction de n.

5 FFT

Reprendre les questions de la section précédente avec la transformation de Fourier Rapide.