

[Help](#)

```
#include <iostream>
#include <cmath>
#include <cstdlib>

using namespace std;

#include "pnl/pnl_fft.h"
#include "pnl/pnl_vector.h"
#include "fft.h"

/*
  Commentaire temporaire:
  il faut coder une transformée Laplace ...
  pour l'instant je laisse le truc de Peter... c'est à faire
*/

static int real_fourrier_transform(double *a, double *b, int n, int sign)
{
    if (sign == -1)
    {
        int k = 0;
        while (k < n && b[k] == 0)
        {
            k++;
        }
        return (k == n);
    }
    else
    {
        int k = 1;
        while (k < n / 2 && a[k] == a[n - k] && b[k] == -b[n - k])
        {
            k++;
        }
        return (k == n / 2);
    }
}
```

```
}

int fft1d(double *a, double *b, int n, int sign)
{
    int is_real;
    is_real = real_fourrier_transform(a, b, n, sign);

    if (sign == -1 && is_real) pnl_real_fft2(a, b, n);
    else if (sign == 1 && is_real) pnl_real_ifft2(a, b, n);
    else if (sign == -1 && !is_real) pnl_fft2(a, b, n);
    else pnl_ifft2(a, b, n);

    return 0;
}
```