

[Help](#)

```
int fft1d(double *a, double *b, int n, int sign);

// One-dimensional Fast Fourier transform
// a : real part of data
// b : imaginary part of data
// n : length (arbitrary)
// sign : sign of the exponent
// for sign = 1
// (a,b)_j becomes equal to 1/n {sum_{k=0}^{n-1} exp(2 * pi
    * i * j * k / n) (a,b)_k
// for sign = -1
// (a,b)_j becomes equal to {sum_{k=0}^{n-1} exp( - 2 * pi
    * i * j * k / n) (a,b)_k

// This program was f2c-ed from Scilab dfft2.f routine
// originally based on algorithm by R. C. Singleton, Stanford, sept. 1968
// and edited by P. Tankov, March 2005

// Statistics on PIII - 800 computer running MS Windows 2000
// FFT of a random vector of size 65536 --- 0.18 sec
// This algorithm works for arbitrary n, but for primes it
// reverts
// to the standard (not fast) DFT which has complexity n^2
```

## References