

FFTW versus NRC FFT

Source: <http://sci.tech-archive.net/Archive/sci.math.num-analysis/2007-01/msg00034.html>

- *From:* "Blacky" <volker.schwarz@xxxxxxxxxxxxxxxx>
 - *Date:* 4 Jan 2007 05:47:06 -0800
-

Hello,

I have a problem with the FFTW algorithm, special the discrete sine transform. Before exchanging the FFT routine known from the NRC (numerical recipes) book by that given from FFTW, I performed some tests. I started with the function $\exp(-t)$ which can be analytically transformed. Applying both codes for that function the FFTW algorithm limits into a function value of one not zero as one would think and what the FFT NRC code does. What is the problem here? The FFT NRC code is checked by using it before on many problems.

Sincerely,

Volker

The code I used:

```
#include<fftw3.h>
#include<math.h>
#include<stdio.h>
#include<stdlib.h>

#define NR_END 1
#define FREE_ARG char*

int main(void)
{
    double *vector(unsigned long start, unsigned long end);
    void free_vector(double *vec, unsigned long start, unsigned long end);
    void Transform_sine(double *function, unsigned long N);

    unsigned int i, N=2048;

    double dr = 0.1;

    double *in_nrc, *out_nrc;

    in_nrc=vector(1, N);
    out_nrc=vector(1, N);
```

FFTW versus NRC FFT

```
double *in, *out;
fftw_plan p;

in=(double*)fftw_malloc(sizeof(double)*N);
out=(double*)fftw_malloc(sizeof(double)*N);
p=fftw_plan_r2r_1d(N,in,out,FFTW_RODFT10,FFTW_ESTIMATE);

for(i=0;i<N;i++) {
in[i]=exp(-dr*i);
in_nrc[i+1]=in[i];
}

Transform_sine(in_nrc, N);

fftw_execute(p);

for(i=0;i<N;i++) {
printf("%e %e %e\n", in[i], out[i], in_nrc[i+1]);
}

fftw_free(in); fftw_free(out);

return(0);
}
```

The NRC routines you will find free online—readable at: www.nr.com

.