```
/*
 * $Id: fft.h,v 1.2 2005/07/11 11:31:39 cperciva Exp $
 */

#ifndef _TRICL_FFT_H_
#define _TRICL_FFT_H_

#include "local.h"
```

The function *tricl_fft_makelut*(LUT, n) generates an FFT lookup table suitable for use in computing FFTs of length up to 2^n. The input n must satisfy $0 \leq n \leq 29$, and LUT must have space to store 2^n doubles (i.e., 2^{n+3} bytes).

```
void tricl_fft_makelut(double *, int);
```

The function *tricl_fft_fft*(DAT, n, LUT) computes a length 2^n in-place FFT on the values z_k where $z_k = \mathrm{DAT}_{2k} + \mathrm{DAT}_{2k+1} i$, using the precomputed lookup table LUT, leaving the output in a wacky order. The input n must satisfy $0 \leq n \leq 29$, DAT must be an array of 2^n complex values (2^{n+1} doubles), and LUT must be as created by *tricl_fft_makelut*(LUT, m) for some $m \geq n$.

```
void tricl_fft_fft(double * __restrict, int, double * __restrict);
```

The function *tricl_fft_ifft*(DAT, n, LUT) computes an inverse FFT corresponding to *tricl_fft_fft*; it takes its input in the wacky order from the output of that function, and leaves its output in normal order.

```
void tricl_fft_ifft(double * __restrict, int, double * __restrict);

#endif /* !_TRICL_FFT_H */
```

A Copyright