1 Definitions

N	Bin size in number of samples, also called number of points
k	Bin index
n	Signal index
x	Time-domain signal
X	Frequency-domain signal

2 Mathematical Definitions

Standard discrete-time version:

$$X(e^{i\omega}) = \sum_{n=0}^{N-1} x[n]e^{-i\omega n}.$$

Also known as:

$$X[k] = \sum_{n=0}^{N-1} x[n]e^{-i2\pi \frac{k}{N} n}.$$

Let:

$$W_k = \frac{2\pi k}{N}.$$

Then,

$$X[k] = \sum_{n=0}^{N-1} x[n]e^{-jnW}.$$

To calculate only real parts, use only *cos* part of Euler identity,

$$real(X[k]) = \sum_{n=0}^{N-1} x[n] * -cos(n * W).$$

The imaginary component looks similar,

$$imag(X[k]) = \sum_{n=0}^{N-1} x[n] * sin(n * W).$$

3 Implementation

```
for k = 0 ... N - 1
  calculate Wk
  for n = 0 ... N - 1
    calculate inner part of real(X[k]), add to sum
    calculate inner part of imag(X[k]), add to sum
```

3.1 Simplified Algorithm

For simple sound processing, we usually only need real parts.

```
for k = 0 ... N - 1
  calculate Wk
  for n = 0 ... N - 1
    calculate inner part of real(X[k]), add to sum
```

3.2 C Code

```c
/*
 * This function returns an array of doubles, length N,
 * whose values represent the N frequency components
 * contained in the signal, x.
 */
double* dft(int N, double* x)
{
  double Wk;
  int k, n;
  double *rval;

  rval = malloc(N * sizeof(double));

  for(k = 0 ; k < N - 1 ; k++) {
    Wk = (2 * PI * k) / N;
    for(n = 0 ; n < N - 1 ; n++) {
      rval[k] += x[n] * -1 * cos(n * Wk);
    }
  }

  return rval;
}
```

Note that this implementation does no scaling or order correction on the results. This means that you will get redundant values. Look at the matlab function fftshift to see how to correct this.

4 Notes

To get 1Hz accuracy, $N = f_s$ is required.