

## ☰ Documentation

[Accelerate](#) / [Fast Fourier transforms](#) / In-Place Functions for 1D Multiple-Signal Real FFT

API Collection

# In-Place Functions for 1D Multiple-Signal Real FFT

Perform fast Fourier transforms in place on multiple-signal 1D real data.

## Overview

The functions in this group use the following operation for a forward real-to-complex transform:

```
N = 1 << Log2N;

// Repeat M times:
for (m = 0; m < M; ++m)
{
    scale = 2;

    // Define a real vector, h:
    for (j = 0; j < N/2; ++j)
    {
        h[2*j + 0] = C->realp[m*IM + j*IC];
        h[2*j + 1] = C->imagp[m*IM + j*IC];
    }

    // Perform Discrete Fourier Transform.
    for (k = 0; k < N; ++k)
        H[k] = scale *
            sum(h[j] * e**(-Direction*2*pi*i*j*k/N), 0 <= j < N);

    // Pack DC and Nyquist components into initial elements.
```

```

C->realp[m*IM + 0*IC] = Re(H[ 0 ]);
C->imagp[m*IM + 0*IC] = Re(H[N/2]).

// Store regular components:
for (k = 1; k < N/2; ++k)
{
    C->realp[m*IM + k*IC] = Re(H[k]);
    C->imagp[m*IM + k*IC] = Im(H[k]);
}
}

```

The functions in this group use the following operation for an inverse complex-to-real transform:

```

N = 1 << Log2N;

scale = 1./N;

// Repeat M times:
for (m = 0; m < M; ++m)
{
    // Define a complex vector, h:
    h[ 0 ] = C->realp[m*IM + 0*IC];
    h[N/2] = C->imagp[m*IM + 0*IC];
    for (j = 1; j < N/2; ++j)
    {
        h[ j ] = C->realp[m*IM + j*IC] + i * C->imagp[m*IM + j*IC];
        h[N-j] = conj(h[j]);
    }

    // Perform Discrete Fourier Transform.
    for (k = 0; k < N; ++k)
        H[k] = scale *
            sum(h[j] * e**(-Direction*2*pi*i*j*k/N), 0 <= j < N);

    // Coerce real results into complex structure:
    for (k = 0; k < N/2; ++k)
    {
        C->realp[m*IM + k*IC] = H[2*k+0];
        C->imagp[m*IM + k*IC] = H[2*k+1];
    }
}

```

The temporary buffer versions perform the same operation but use a temporary buffer for improved performance.

---

## Topics

### In-Place FFT Functions

`vDSP_fftm_zrip`

Computes a forward or inverse in-place, single-precision real FFT on multiple signals.

`vDSP_fftm_zrippD`

Computes a forward or inverse in-place, double-precision real FFT on multiple signals.

### In-Place FFT Functions with Temporary Buffer

`vDSP_fftm_zript`

Computes a forward or inverse in-place, single-precision real FFT on multiple signals using a temporary buffer.

`vDSP_fftm_zriptD`

Computes a forward or inverse in-place, double-precision real FFT on multiple signals using a temporary buffer.

---

## See Also

### Functions for 1D Multiple-Signal Real FFT

≡ Out-of-Place Functions for 1D Multiple-Signal Real FFT

Perform fast Fourier transforms out of place on multiple-signal 1D real data.