# hamm_windowx(), hammd_windowx()

■ **Create Hamming Window**

$$C_n = 0.54 - 0.46\cos\frac{2\pi n}{N} \qquad n = \{0, N-1\}$$

## Synopsis

```
void hamm_windowx(
float *C,     /* output vector */
int N,        /* real element count, N >= 16 */
int flag,     /* flags; specify SAL_HALF_WINDOW to */
                        /* create only first (N+1)/2 pts */
int eflag     /* ESAL flag */
);
```

```
void hammd_windowx(
double *c,    /* output vector */
int n,        /* double-precision element count, N >= 16 */
int flag,     /* flags; specify SAL_HALF_WINDOW to */
                        /* create only first (N+1)/2 pts */
int eflag     /* ESAL flag */
);
```

## Description

Functions `hamm_windowx()` and `hammd_windowx()` create a Hamming window function, which can be multiplied by a vector using `vmulx()` or `vmuldx()`.

Specify the `SAL_HALF_WINDOW` flag to create only the first (*N* + 1)/2 points.

## Reference

Harris, Fredric J., "On the Use of Windows for Harmonic Analysis with the Discrete Fourier Transform," *Proceedings of the IEEE*, vol. 64, no. 1, January 1978.

## See Also

`vmulx(), vmuldx(), vmulix()` on page 510.

# hann_windowx(), hannd_windowx()

■ **Create Hanning Window**

$$C_n = W\left(1.0 - \cos\frac{2\pi n}{N}\right) \qquad n = \{0, N-1\}$$

Where: W = 0.5 for a denormalized window (default)
W = 0.8165 for a normalized window

## Synopsis

```
void hann_windowx(
float *C,     /* output vector */
int N,        /* real element count */
int flag,     /* window flags */
int eflag);   /* ESAL flag */
```

```
void hannd_windowx(
double *C,    /* output vector */
int N,        /* double-precision element count */
int flag,     /* window flags */
int eflag);   /* ESAL flag */);
```

## Description

Functions `hann_windowx()` and `hannd_windowx()` create a Hanning window function, which can be multiplied by a vector using `vmulx()` or `vmuldx()`. The window flag parameter can have the following values:

`SAL_HANN_DENORM` DEFAULT

Creates a denormalized window.

`SAL_HANN_NORM`

Creates a normalized window.

`SAL_HALF_WINDOW`

Creates only the first (*N* + 1)/2 points.

## See Also

`vmulx(), vmuldx(), vmulix()` on page 510.