


4041R03 SIGNAL PROCESSING ROM

PROGRAMMER'S REFERENCE GUIDE

Copyright © 1983 Tektronix, Inc. All rights reserved. Contents of this publication may not be reproduced in any form without the written permission of Tektronix, Inc.

Products of Tektronix, Inc. and its subsidiaries are covered by U.S. and foreign patents and/or pending patents.

TEKTRONIX, TEK, SCOPE-MOBILE, and  are registered trademarks of Tektronix, Inc. TELEQUIPMENT is a registered trademark of Tektronix U.K. Limited.

Printed in U.S.A. Specification and price change privileges are reserved.

INSTRUMENT SERIAL NUMBERS

Each instrument has a serial number on a panel insert, tag, or stamped on the chassis. The first number or letter designates the country of manufacture. The last five digits of the serial number are assigned sequentially and are unique to each instrument. Those manufactured in the United States have six unique digits. The country of manufacture is identified as follows:

B000000	Tektronix, Inc., Beaverton, Oregon, USA
100000	Tektronix Guernsey, Ltd., Channel Islands
200000	Tektronix United Kingdom, Ltd., London
300000	Sony/Tektronix, Japan
700000	Tektronix Holland, NV, Heerenveen, The Netherlands

4041R03

SIGNAL PROCESSING ROM

RCALL "**AMAX**", numarray, {numvar} , {numvar}
{numarray} {numarray}

500 RCALL "**AMAX**", Datarray, Maxvalue, Maxindex

The array Datarray is searched for its maximum value. The maximum value is returned in numeric variable Maxvalue. The index of the occurrence of the maximum value within Datarray is returned in numeric variable Maxindex.

RCALL "**AMIN**", numarray, {numvar} , {numvar}
{numarray} {numarray}

5000 RCALL "**AMIN**", Datarray, Minvalue, Minindex

The array Datarray is searched for its minimum value. The minimum value is returned in numeric variable Minvalue. The location of the occurrence of the minimum value within Datarray is returned in numeric variable Minindex.

RCALL "**CONVL**", fparray, fparray, fparray

20000 RCALL "**CONVL**", source1, source2, result

A fast convolution is performed on numeric arrays Source1 and Source2. The result is stored in numeric array Result. Arrays Source1 and Source2 are overwritten by intermediate results.

RCALL "**CORR**", fparray, fparray, fparray

15520 RCALL "**CORR**", source1, source2, result

A fast correlation is performed on numeric arrays Source1 and Source2. The result is stored in numeric array Result. Arrays Source1 and Source2 are overwritten by intermediate results.

RCALL "**CROSS**", numarray, numexp, {fpvar} [, numexp]
{fparray}

20 RCALL "**CROSS**", Sinewave, Sinemax, Crosspt, N

The array Sinewave is searched for the Nth crossing of the value given by numeric variable Sinemax. The location within Sinewave of the Nth crossing is returned in numeric variable Crosspt.

RCALL **"DIF2"**,numarray,numarray

2000 RCALL **"DIF2"**, source, dest

A two-point differentiation is performed on numeric array Source. The result is returned in numeric array Dest.

RCALL **"DIF3"**,numarray,numarray

2050 RCALL **"DIF3"**, a1, a2

A three-point differentiation is performed on numeric array A1. The result is returned in numeric array A2.

RCALL **"FFT"**,fparray

2000 RCALL **"FFT"**, signal02

A fast Fourier transform is performed on the data in numeric array Signal02. The results overwrite the original data in the array.

RCALL **"IFT"**, fparray

RCALL **"IFT"**,cplxdata

An inverse Fourier transformation is performed on the data in numeric array Cplxdata. The results overwrite the data in the original array.

RCALL **"INLEAV"**,numarray,numarray,numarray

5025 RCALL **"INLEAV"**,real,imaginary,intrleav

Data from numeric arrays Real and Imaginary are interleaved into numeric array Intrleav.

RCALL **"INTEGRAT"**,fparray,fparray

10000 RCALL **"INTEGRAT"**, source, result

Numeric array Source is integrated, and the result is stored in numeric array Result.

RCALL **"INTERP1"**,{numarray,numexp,numexp,numvar}
[,numexp[,numexp[,numexp]]]

500 RCALL **"INTERP1"**, datarray, 27.73, 28, outvar, .25, 0.

The value of numeric array Datarray at index 27.73 is computed, using the 28 elements from numeric array Datarray closest to 27.73 (14 elements preceding and including index 27, and the 14 elements after index 27). The convolution uses a SIN(X)/X function windowed by a 25% trapezoidal window. The resulting value is stored in numeric variable Outvar.

```
RCALL "INTERP1",imagin,1.3,12,intv,tapr,wndw,-1
```

The value of numeric array Imagin at index 1.3 is computed using a 12-point sample. The end-of-array algorithm internally extends the data left of Imagin(1) by five elements of opposite sign to the element values selected by the "bounce" method.

```
RCALL "INTERP1",mag,-1.5,numterms,mirrv,tapr,wndw,1
```

The value of numeric array Mag(-1.5) is computed, using the assumption that the Mag array is symmetrical about its first element.

```
RCALL "MEANSTDV",numarray,{fpvar} [,fpvar]  
                                {fparry} [,fparry]
```

```
1000 RCALL "MEANSTDV",datarray,mean,std
```

The arithmetic mean and standard deviation of numeric array Datarray are computed. The arithmetic mean is returned in numeric variable Mean, and the standard deviation is returned in numeric variable Std.

```
RCALL "POLAR",fparry,fparry,fparry [,numexp]
```

```
2050 RCALL "POLAR",X,M,P
```

FFT data from numeric array X is converted to polar form. Magnitude information is stored in numeric array M, phase information in numeric array P.

```
RCALL "POLAR2",fparry,fparry,fparry,fparry[,numexp]
```

```
2100 RCALL "POLAR2",X,Y,M,P
```

A rectangular-to-polar conversion is performed on numeric arrays X and Y. Magnitude data are returned in numeric array M, and phase data are returned in numeric array P.

RCALL "RECTANG2",farray,farray,farray,farray
[,numexp]

RCALL "RECTANG2",Mag,Phase,Xcoords,Ycoords

A polar-to-rectangular conversion is performed on numeric arrays Mag and Phase. X-coordinate information is returned in numeric array Xcoords, Y-coordinate information in numeric array Ycoords.

RCALL "TAPER",farray[,numexp[,numexp]]

2900 RCALL "TAPER", datarray, .5,1

Numeric array Datarray is tapered, using a taper percentage of 50% and a 1/2-cycle sine taper function.

RCALL "UNLEAV",numarray,numarray,numarray

30620 RCALL "UNLEAV",complex,reals,imaginary

Numeric array Complex is UNLEAV'd into two numeric arrays, Real and Imaginary.

ERROR MESSAGES

1. An array passed to a romcall was not singly dimensioned.
2. An argument passed to a romcall was of an incorrect type, e.g., a floating point variable was expected, and an integer variable was found.
3. An array passed to a romcall was not the correct dimension, e.g., an array of size 24 was passed to the FFT, or an array of fewer than 3 elements was passed to INT.
4. The value of an argument passed to a romcall was out of range, e.g., the value specified for a crossing point was -2.
5. The two input arrays may not be the same array.

**ALLOWABLE TYPE COMBINATIONS
FOR DIF2 AND DIF3 ROMCALLS**
(Allowable combinations are indicated by "**".)

Source array type	Destination array type		
	integer array	short float array	long float array
integer array	* (not allowed for DIF3)	*	*
short float array		*	
long float array		*	*

PHASE CALCULATION TABLE

r	i	phase
<0	$=0$	$= -\pi$
<0	<0	$= \arctangent(i/r) - \pi$
$=0$	<0	$= -\pi/2$
>0	<0	$= \arctangent(i/r)$
>0	$=0$	$=0$
>0	>0	$= \arctangent(i/r)$
$=0$	>0	$= \pi/2$
<0	>0	$= \arctangent(i/r) + \pi$

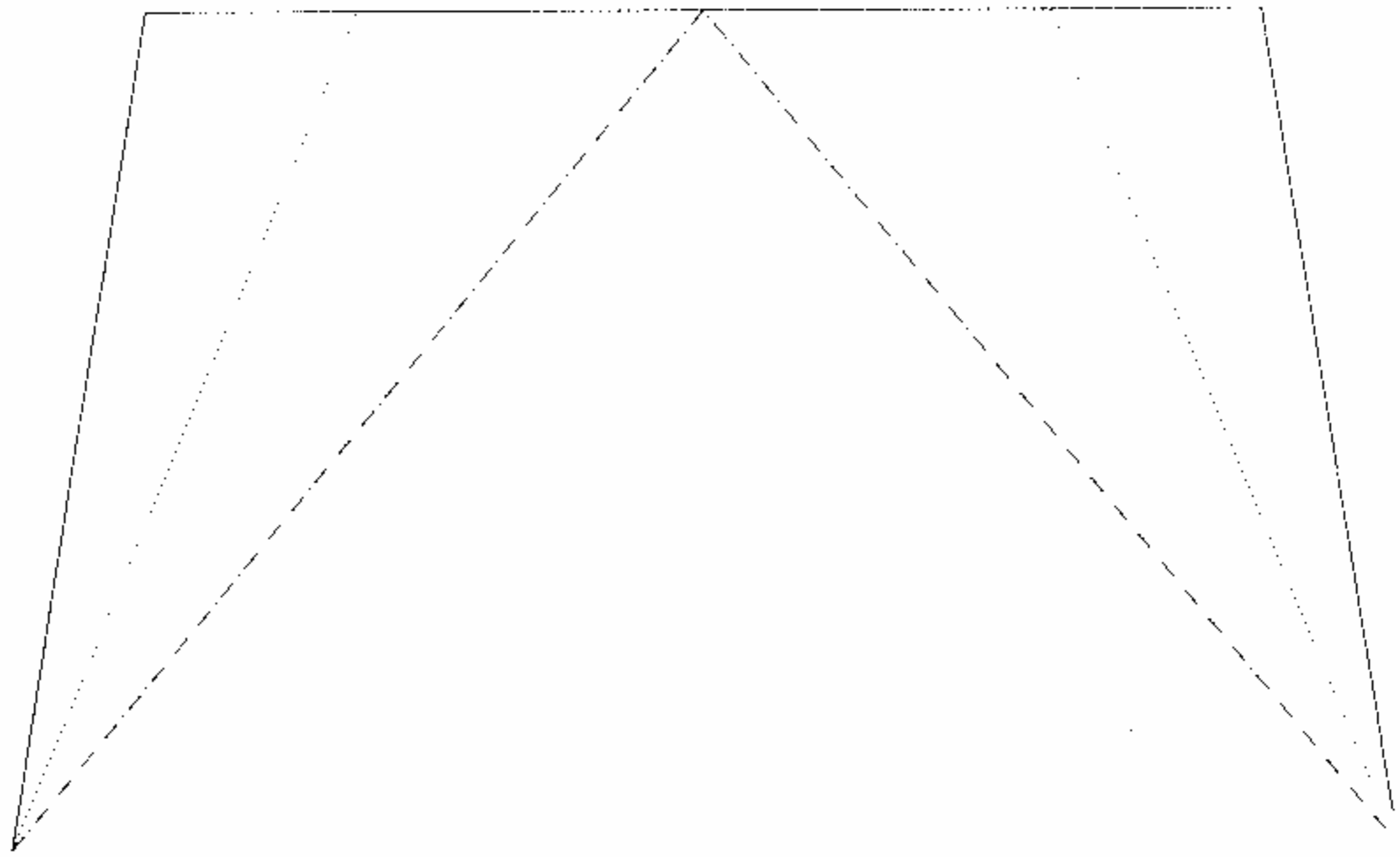
SUGGESTED FILTER LENGTHS, TAPER PERCENTAGES, AND TAPER WINDOW TYPES

Filter Length Type	Taper Percentage	Taper Window	Usage
8	.25	0	Frequency-domain pulse height measurement
24	.25 to .5	2	Pulse width, rise- and fall-time measurement
28	.25 to .5	2,3,4	Pulse-step ringing measurement

END-OF-ARRAY ALGORITHMS

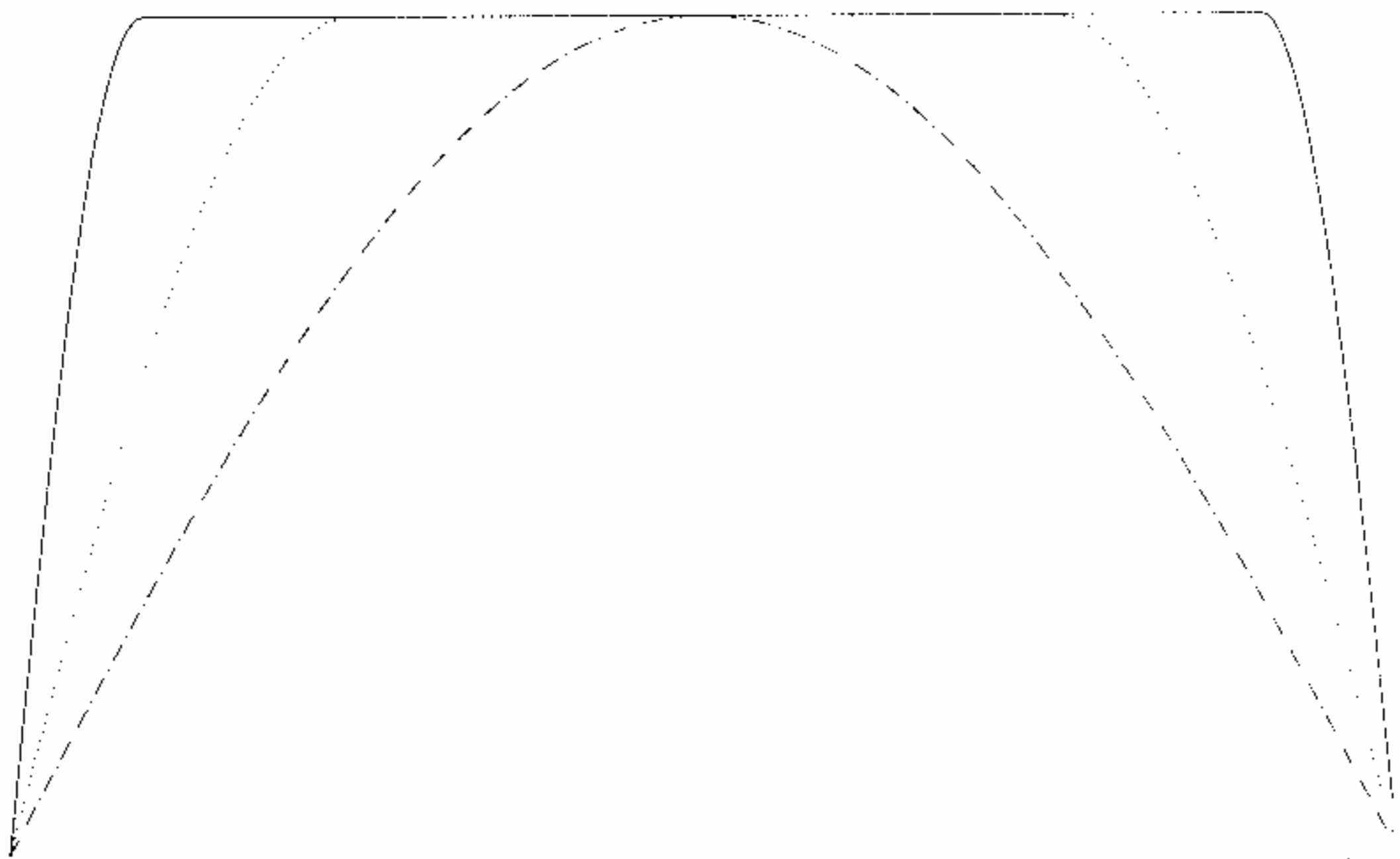
End-of-array algorithm #	Algorithm
0	Repeat end element value (default)
1	"Bounce" off array ends to pick replacement element value.
-1	"Bounce" off array ends (as in algorithm number 1) and change sign of selected element value.
2	"Wrap" around from end of array back to start, or vice versa.

The taper percentage defaults to .1 (10% tapering), the taper window type defaults to 2 (extended Cosine bell), and the end-of-array algorithm number defaults to 0 (repeat end value).



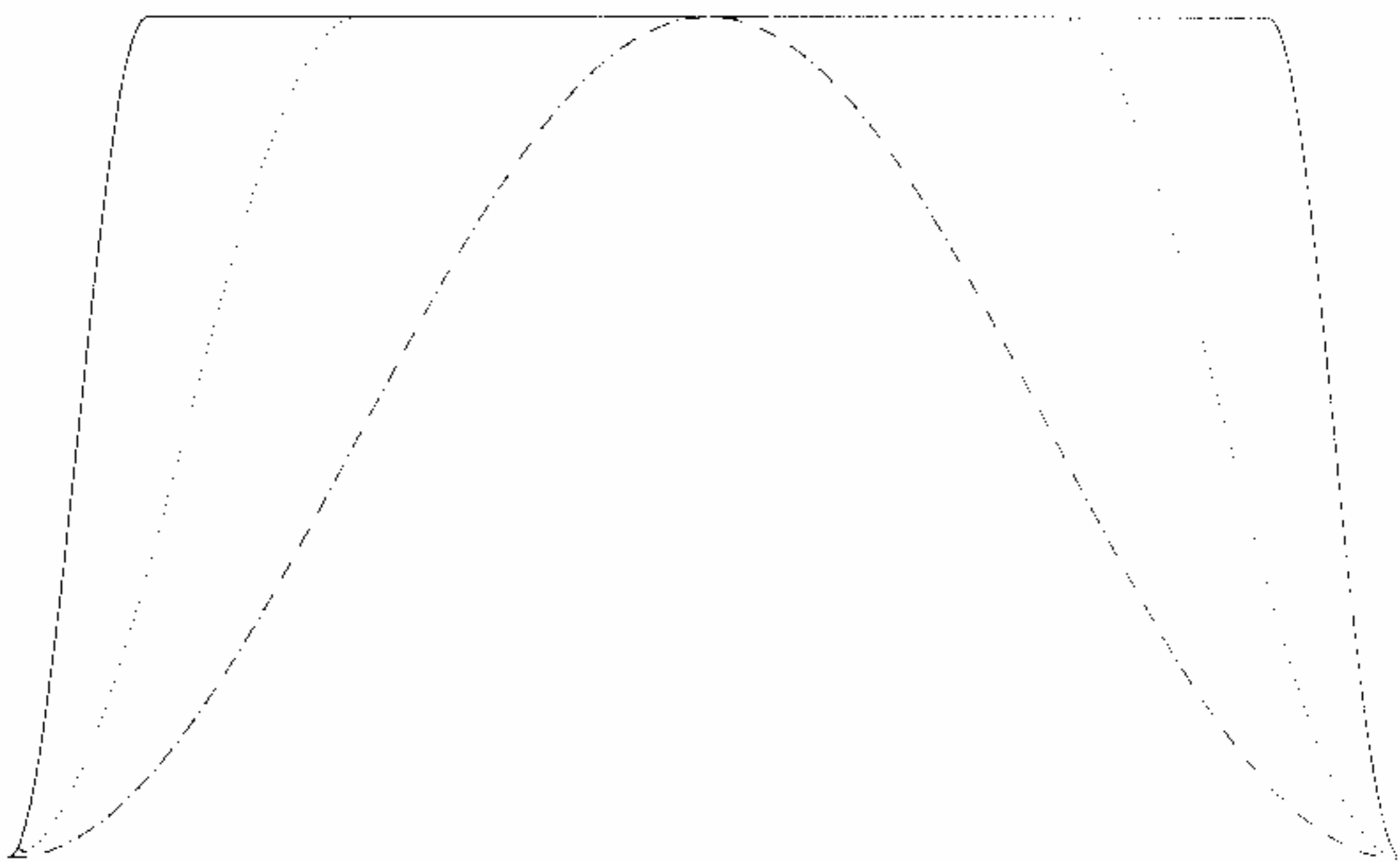
4560-01

FIGURE 1
Trapezoidal (Triangular) Window.
Taper Window Type = 0



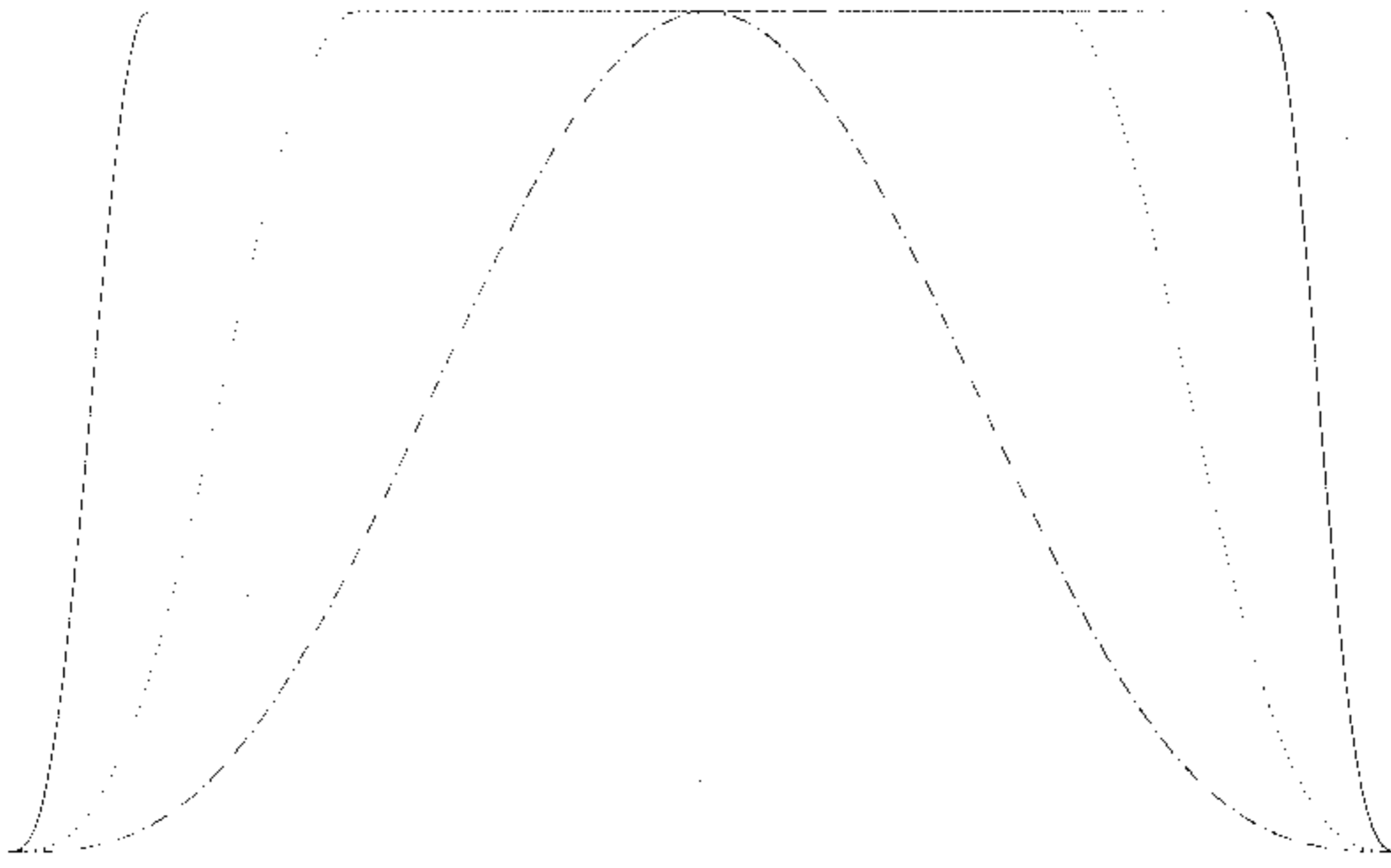
4560-02

FIGURE 2
1/2 Cycle Sine Window.
Taper Window Type = 1



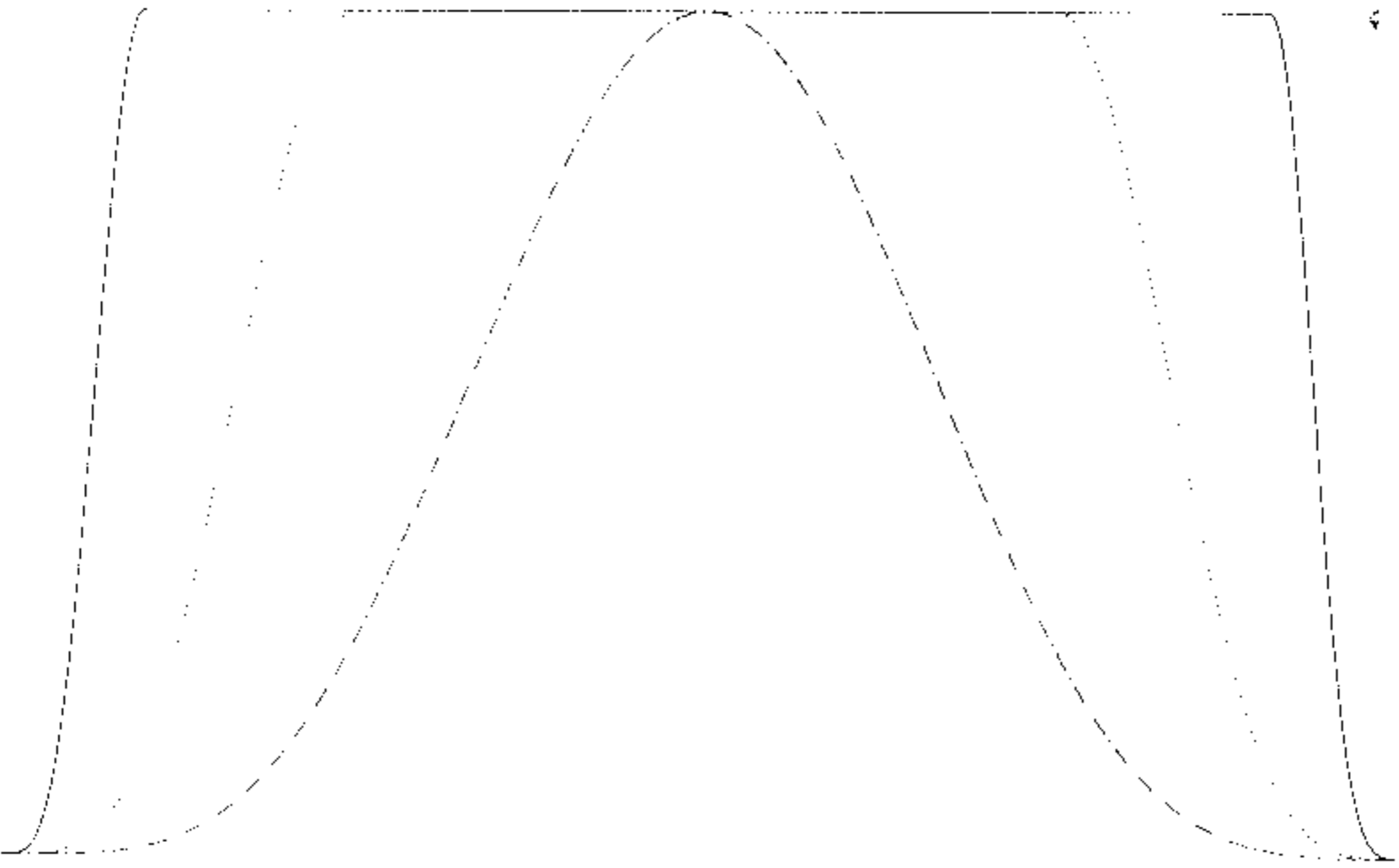
4560-03

FIGURE 3
Extended Cosine Bell Window.
Taper Window Type = 2



4560-04

FIGURE 4
 1/2 Cycle Sine Cubed Window.
 Taper Window Type = 3



4560-05

FIGURE 5
 Cosine Squared Window.
 Taper Window Type = 4

TAPER PERCENTAGE :

- 10% _____
- 25%
- 50% - . - . - .