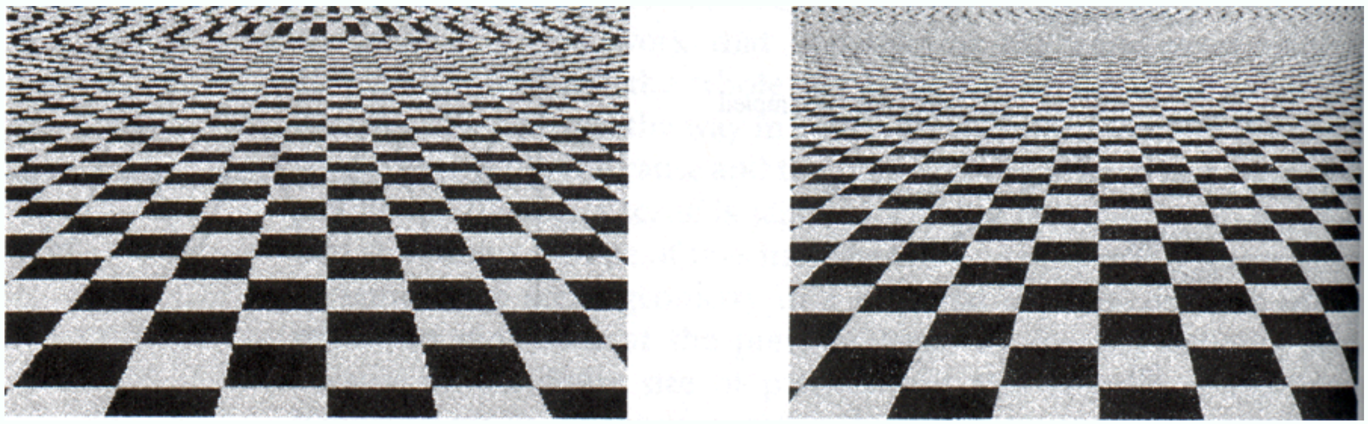
Aliasing and Supersampling

D.A. Forsyth, with much use of John Hart's notes

Aliasing

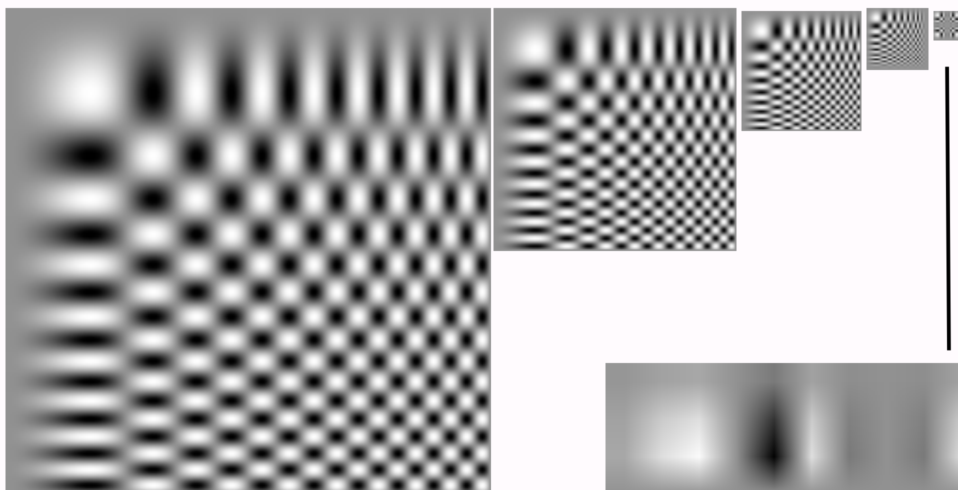


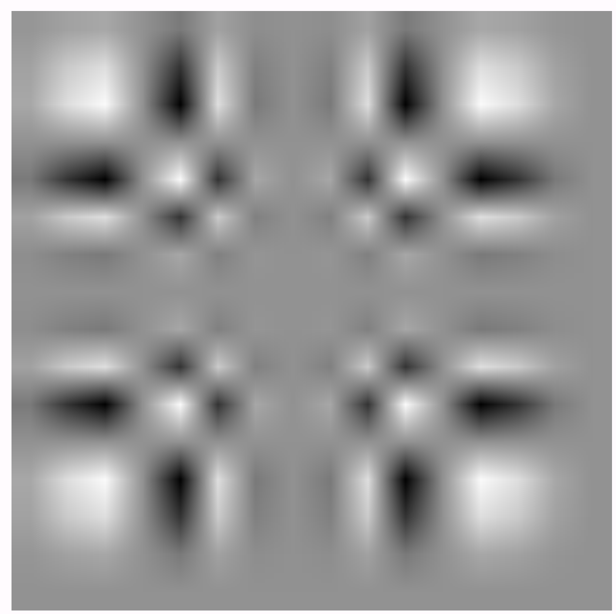
from Watt and Policarpo, The Computer Image

Scaled representations

- Represent one image with many different resolutions
- Why?
    - Efficient texture mapping
        - if the object is tiny in the rendered image, why use a big texture?
    - many other applications will emerge

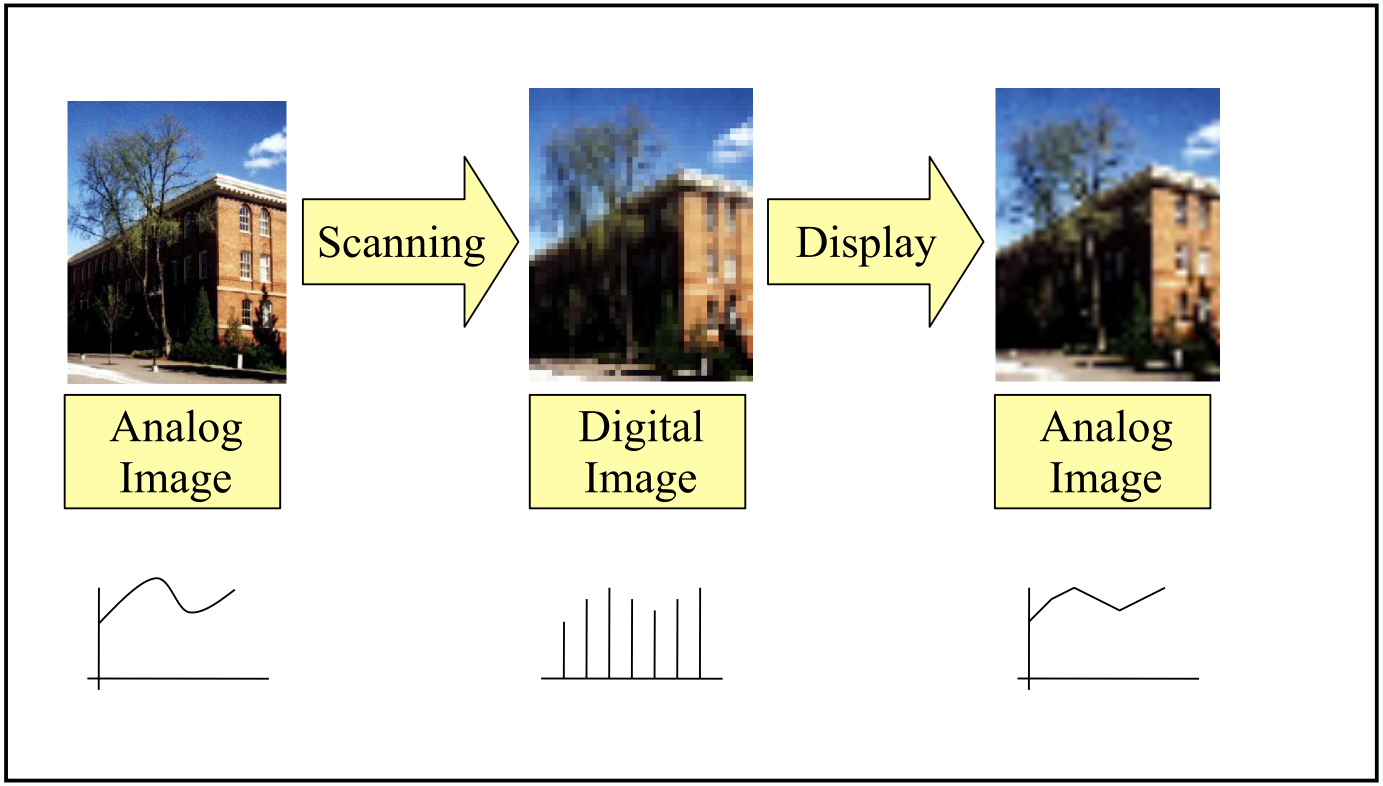
Carelessness causes aliasing



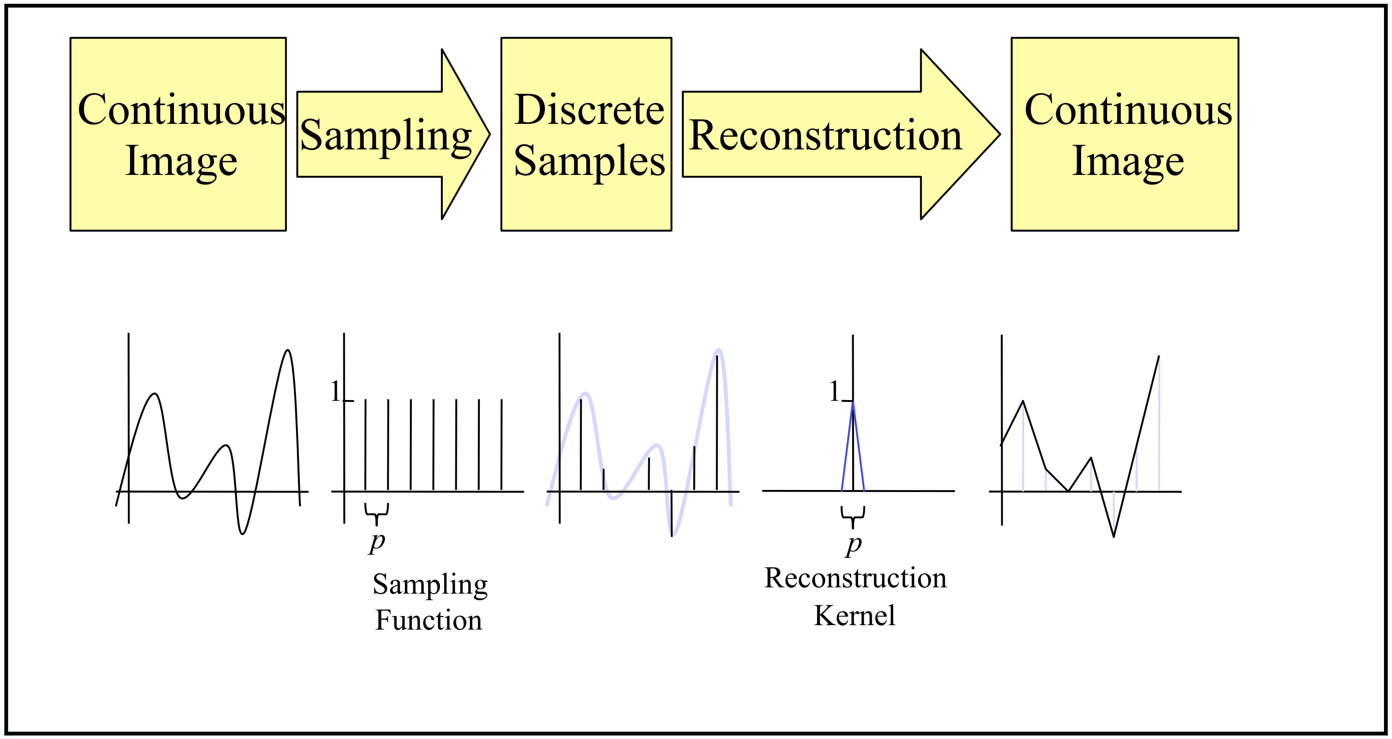


Obtained pyramid of images by subsampling

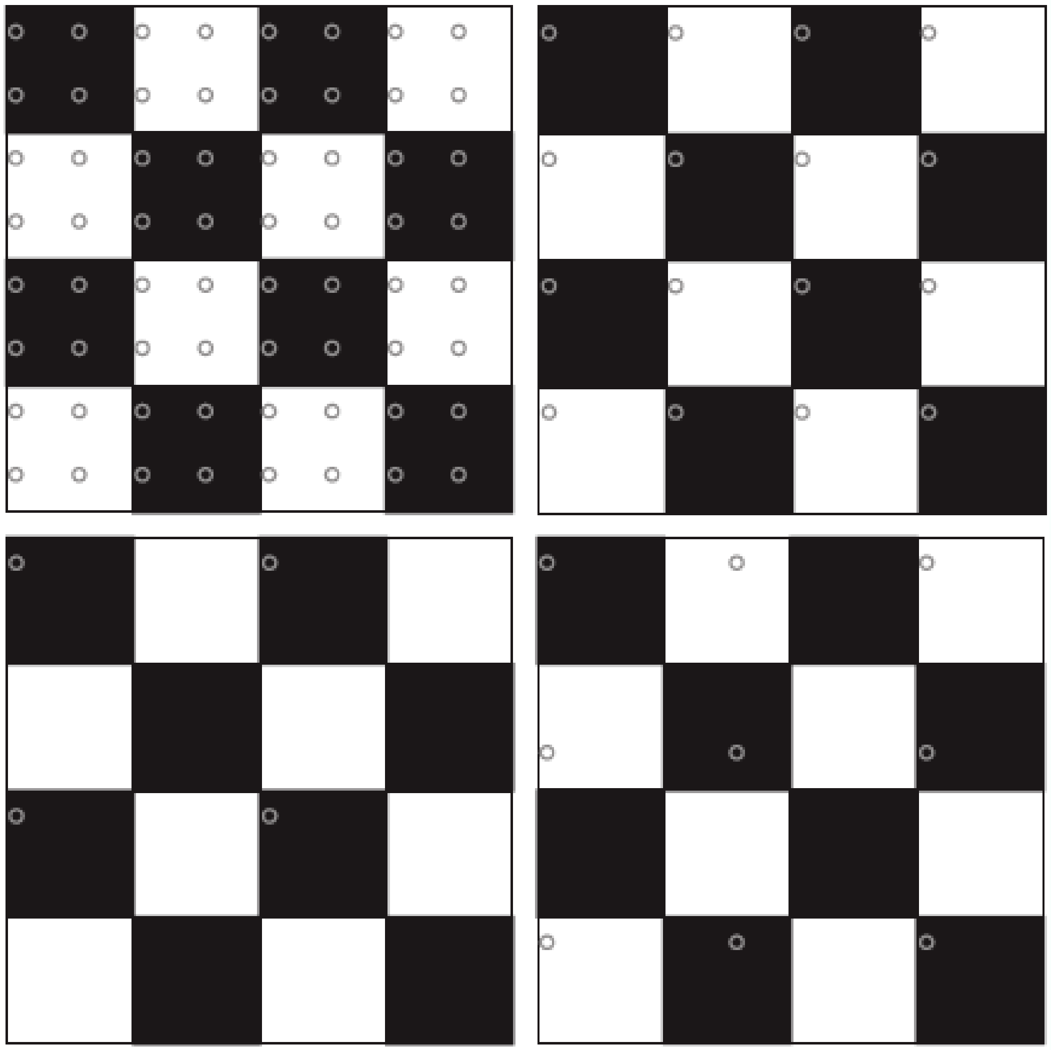
Aliasing is a product of sampling



Sampling and reconstruction

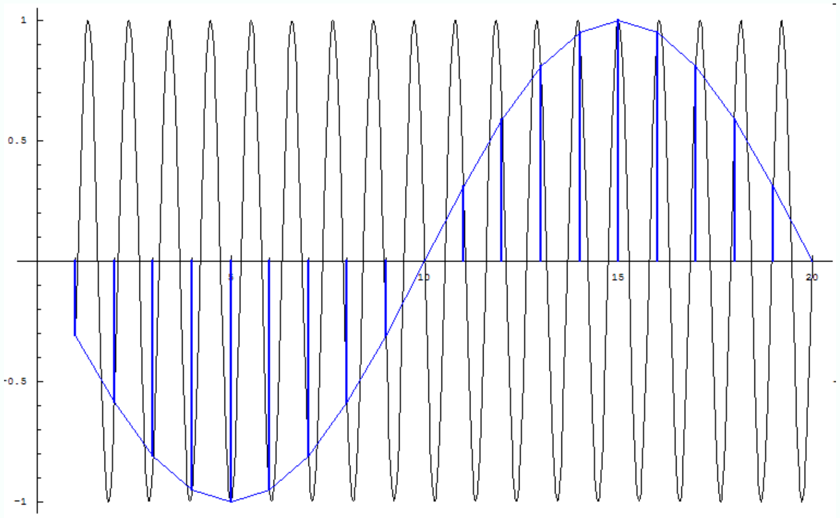


Aliasing and fast changing signals



More aliasing examples

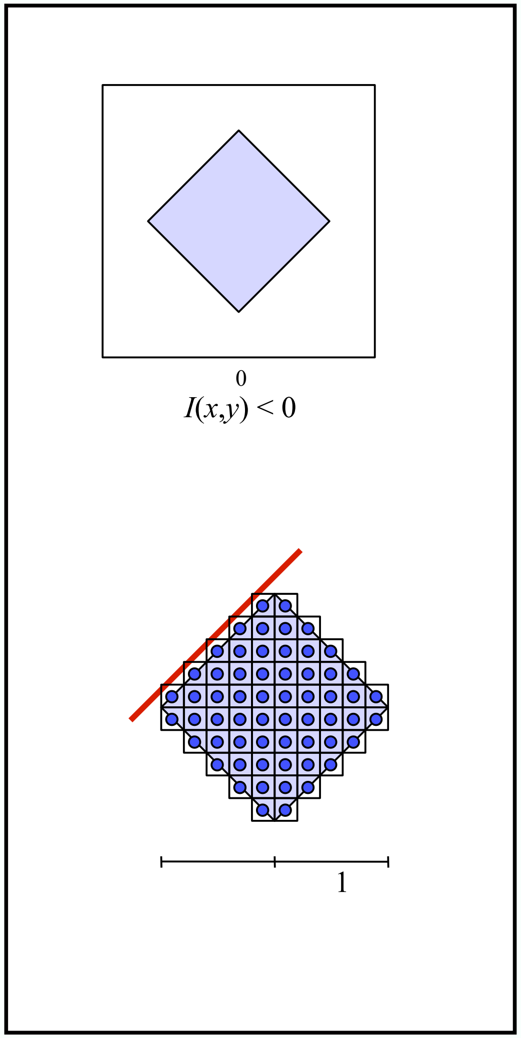
- Undersampled sine wave ->



- Color shimmering on striped shirts on TV
- Wheels going backwards in movies
  - temporal aliasing

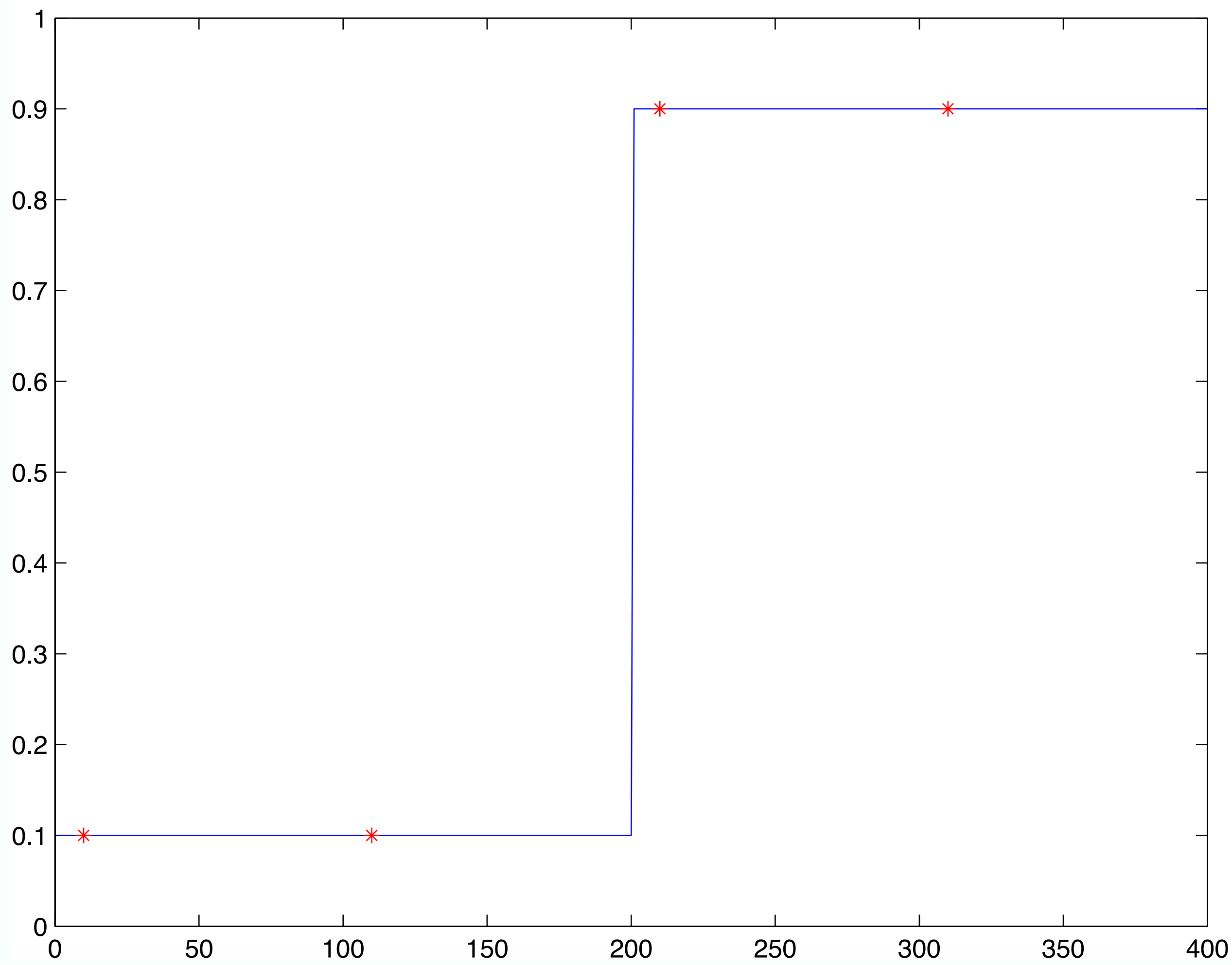
Another aliasing example

- Continuous function at top
- Sampled version at bottom
  - Plot one if $I(x,y) \le 0$, otherwise zero



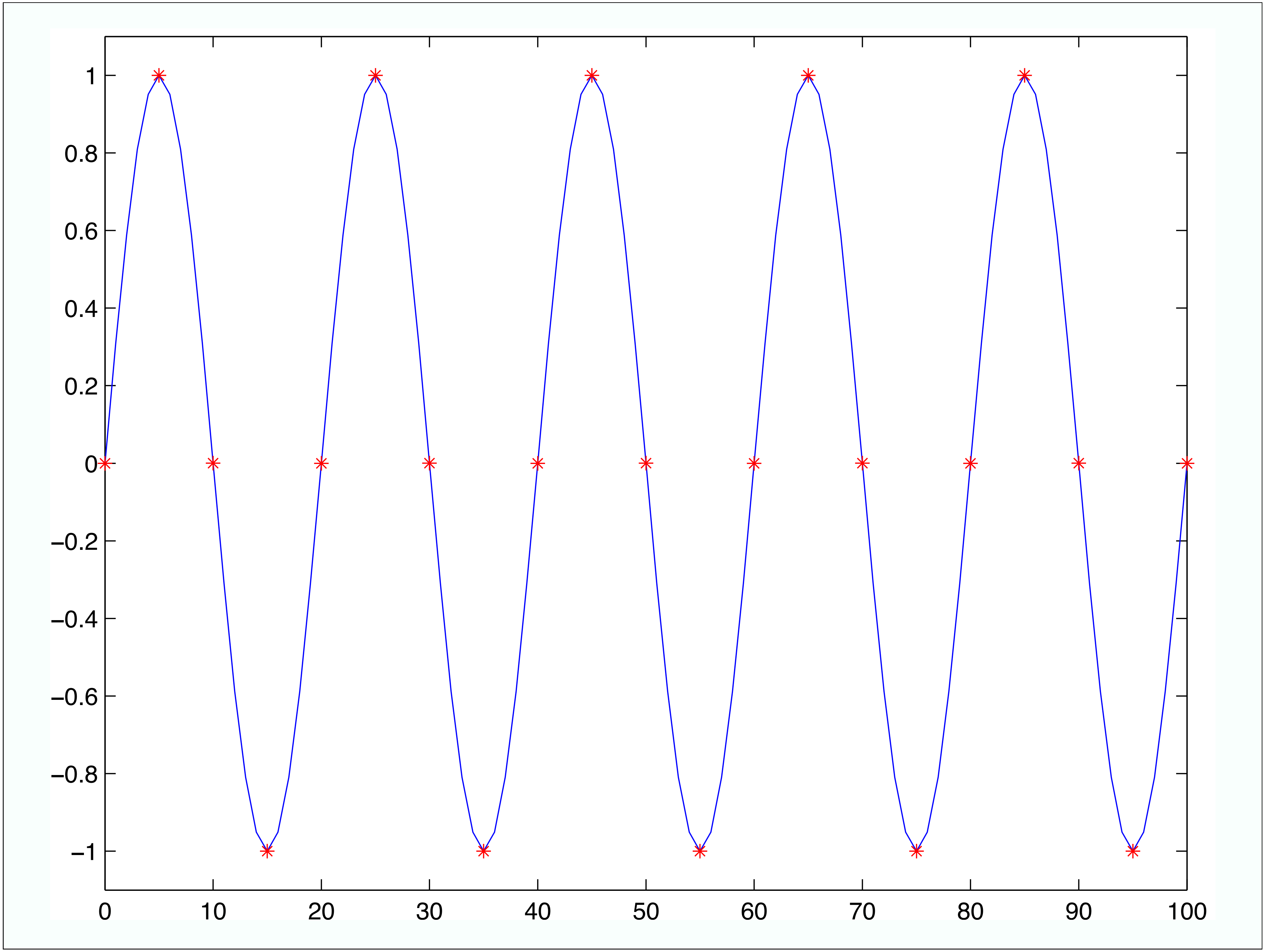
Another aliasing example

- location of a sharp change is known poorly

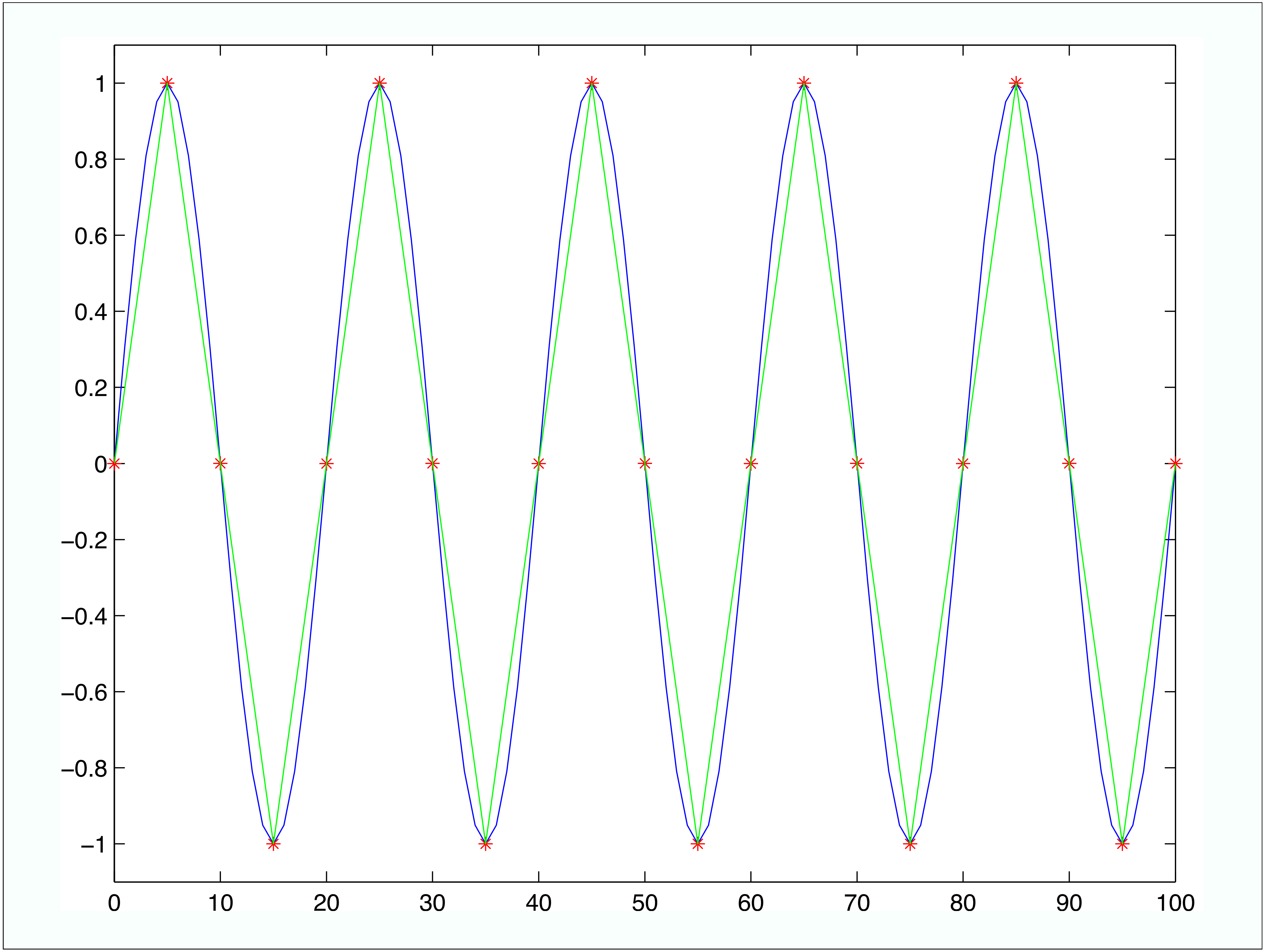


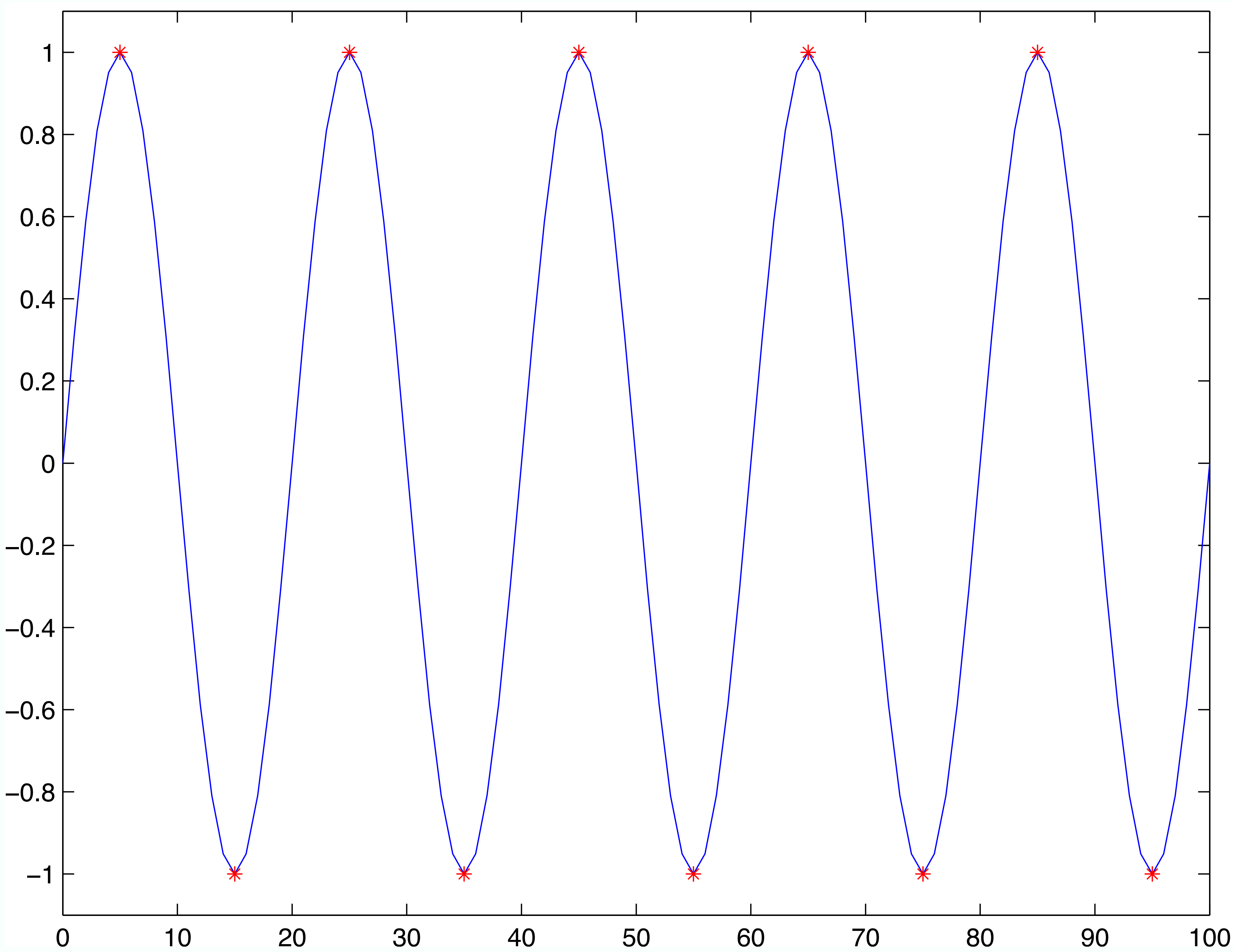
Fundamental facts

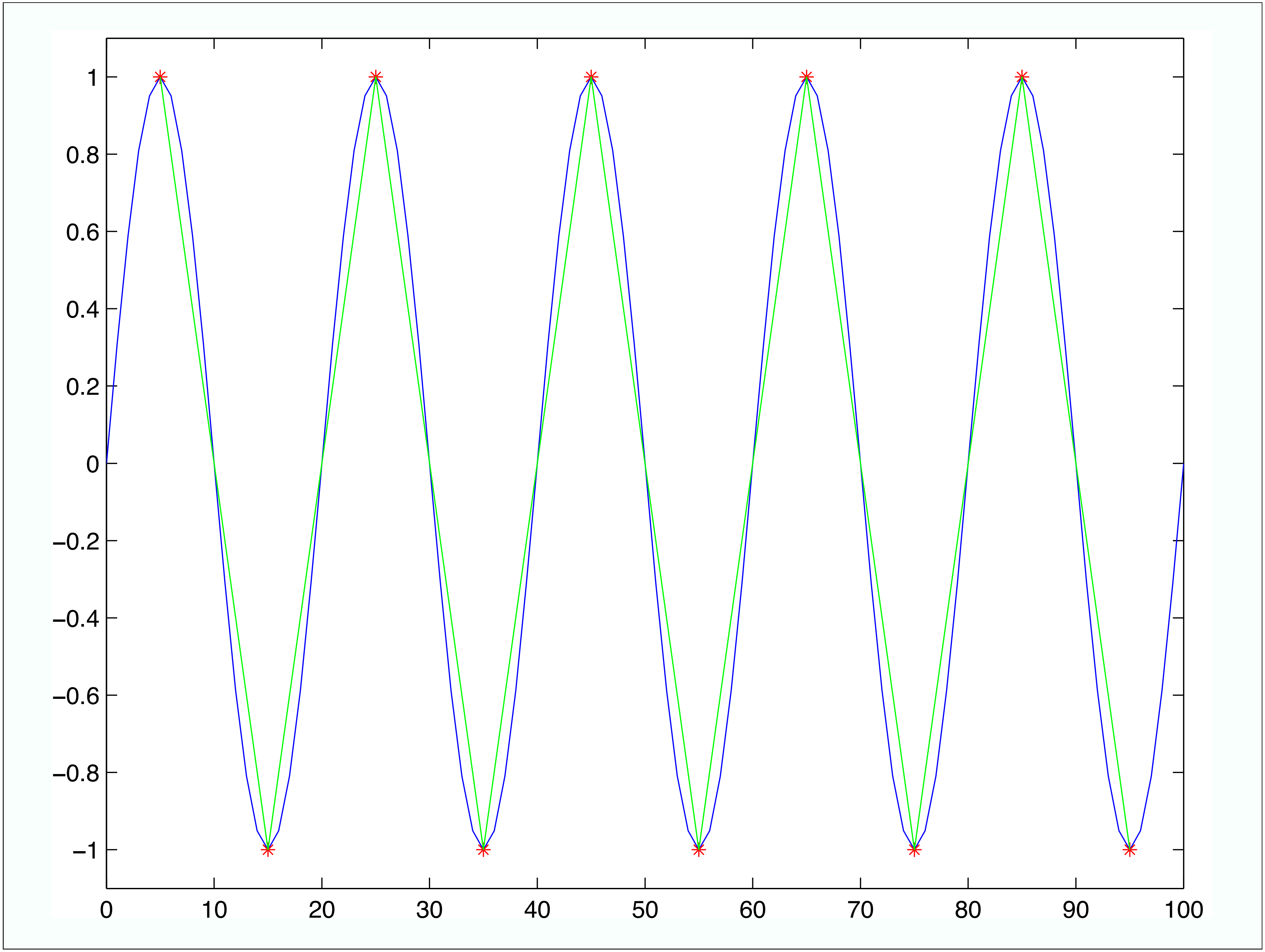
- A sine wave will alias if sampled less often than twice per period

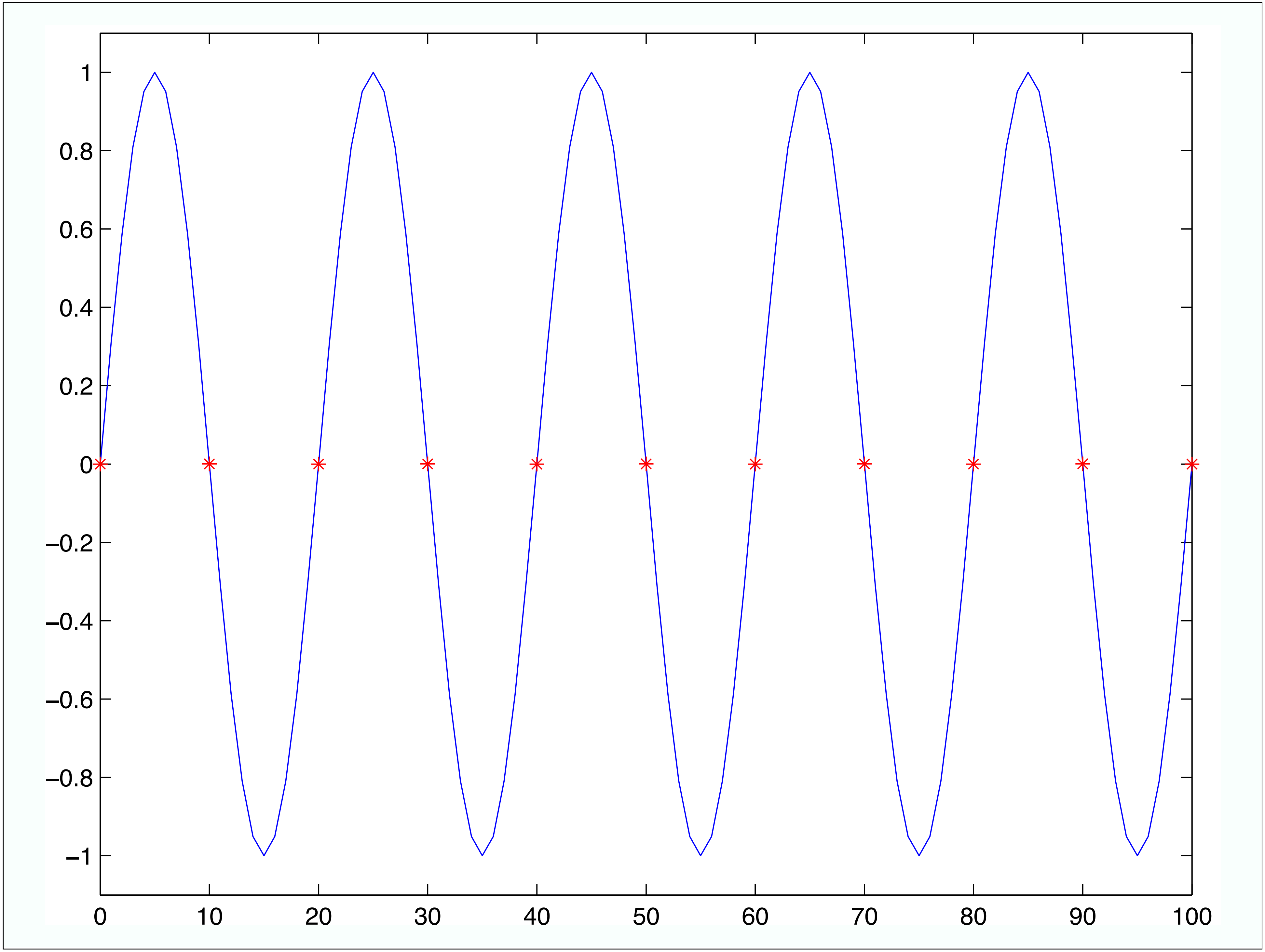


1
0.8
0.6
0.4
0.2
0
−0.2
−0.4
−0.6
−0.8
−1
0
10
20
30
40
50
60
70
80
90
100

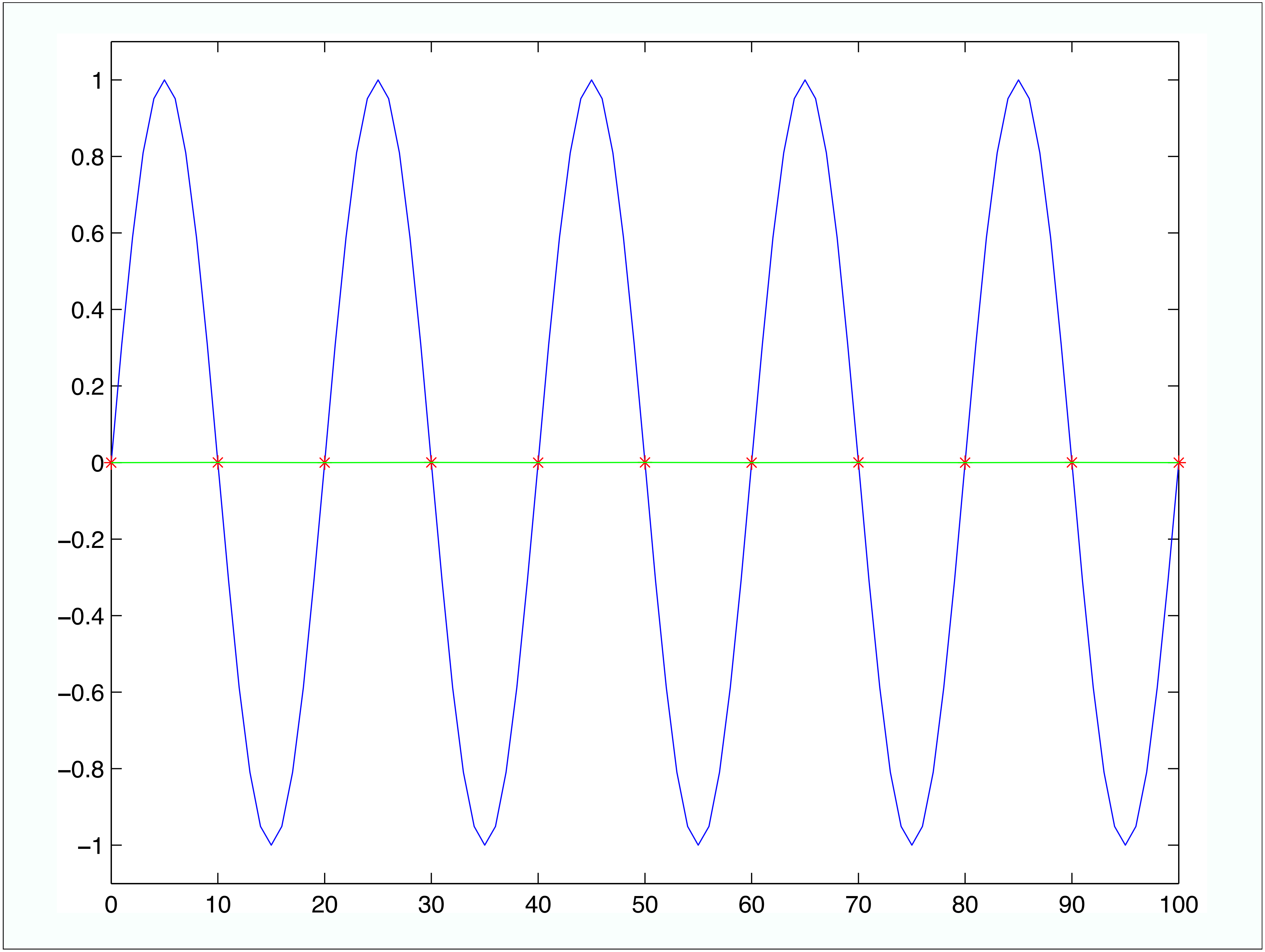

1
0.8
0.6
0.4
0.2
0
−0.2
−0.4
−0.6
−0.8
−1
0
10
20
30
40
50
60
70
80
90
100


1
0.8
0.6
0.4
0.2
0
−0.2
−0.4
−0.6
−0.8
−1
0
10
20
30
40
50
60
70
80
90
100

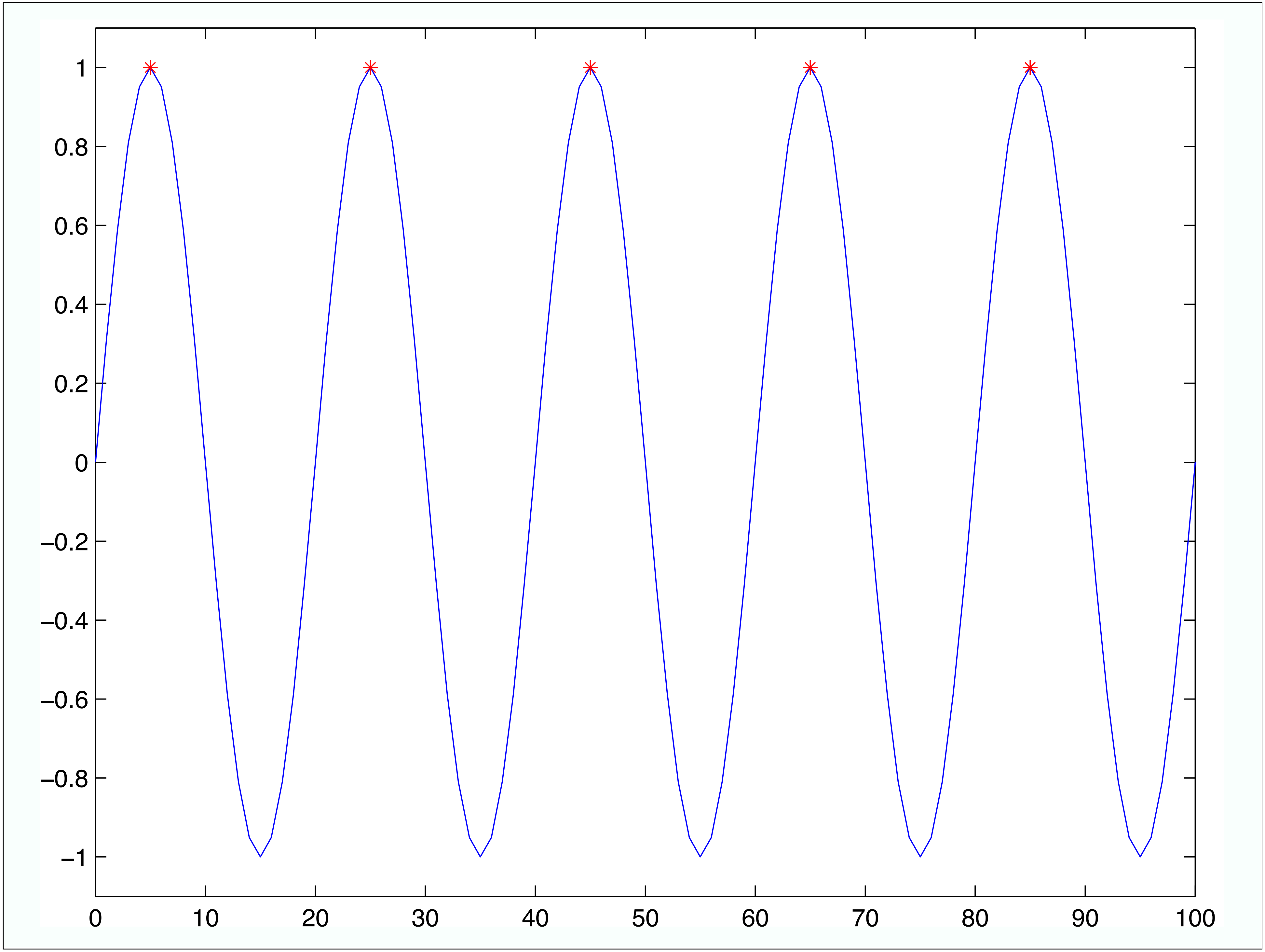

1
0.8
0.6
0.4
0.2
0
−0.2
−0.4
−0.6
−0.8
−1
0
10
20
30
40
50
60
70
80
90
100



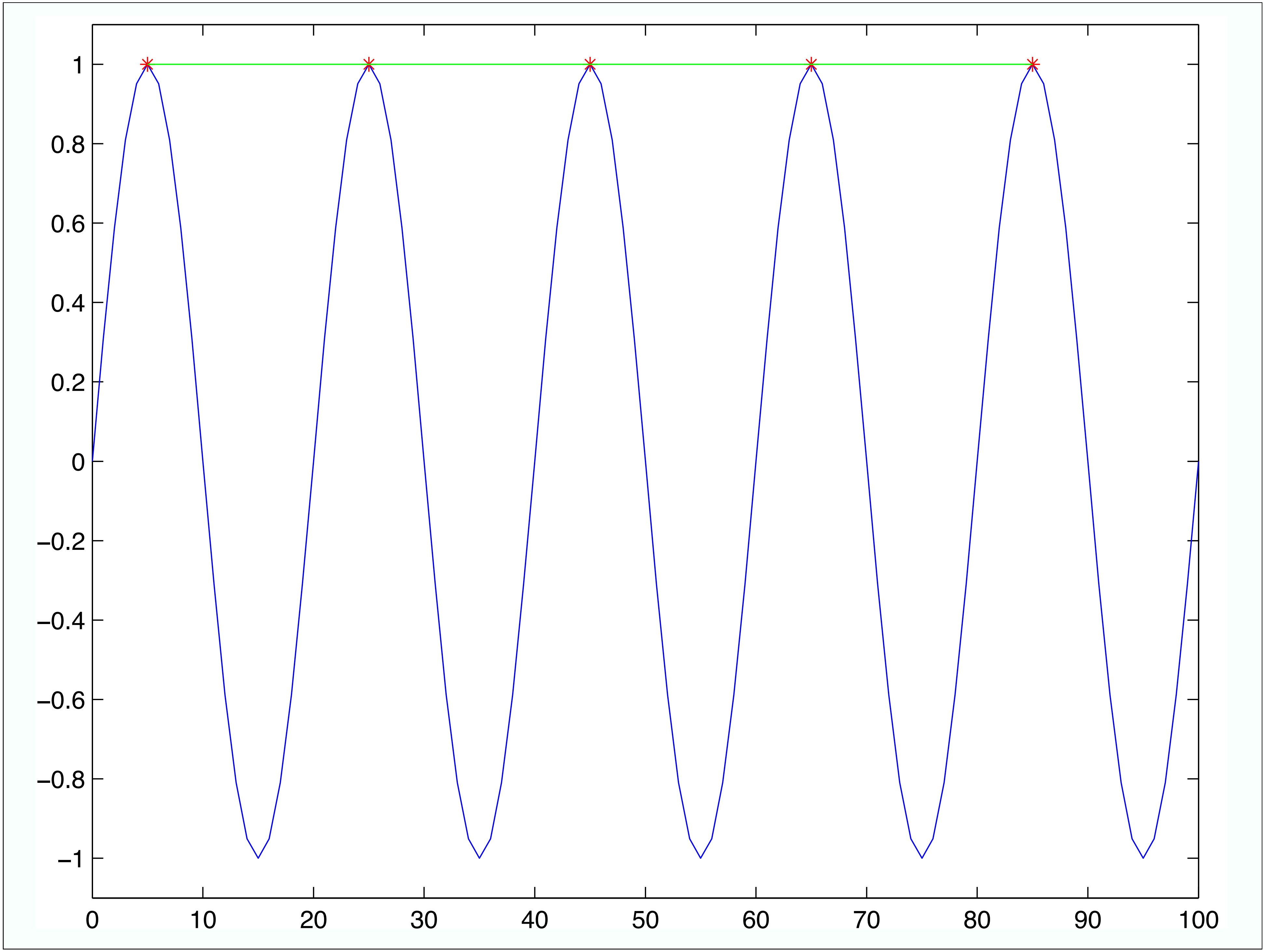
1
0.8
0.6
0.4
0.2
0
−0.2
−0.4
−0.6
−0.8
−1
0
10
20
30
40
50
60
70
80
90
100



1
0.8
0.6
0.4
0.2
0
−0.2
−0.4
−0.6
−0.8
−1
0
10
20
30
40
50
60
70
80
90
100

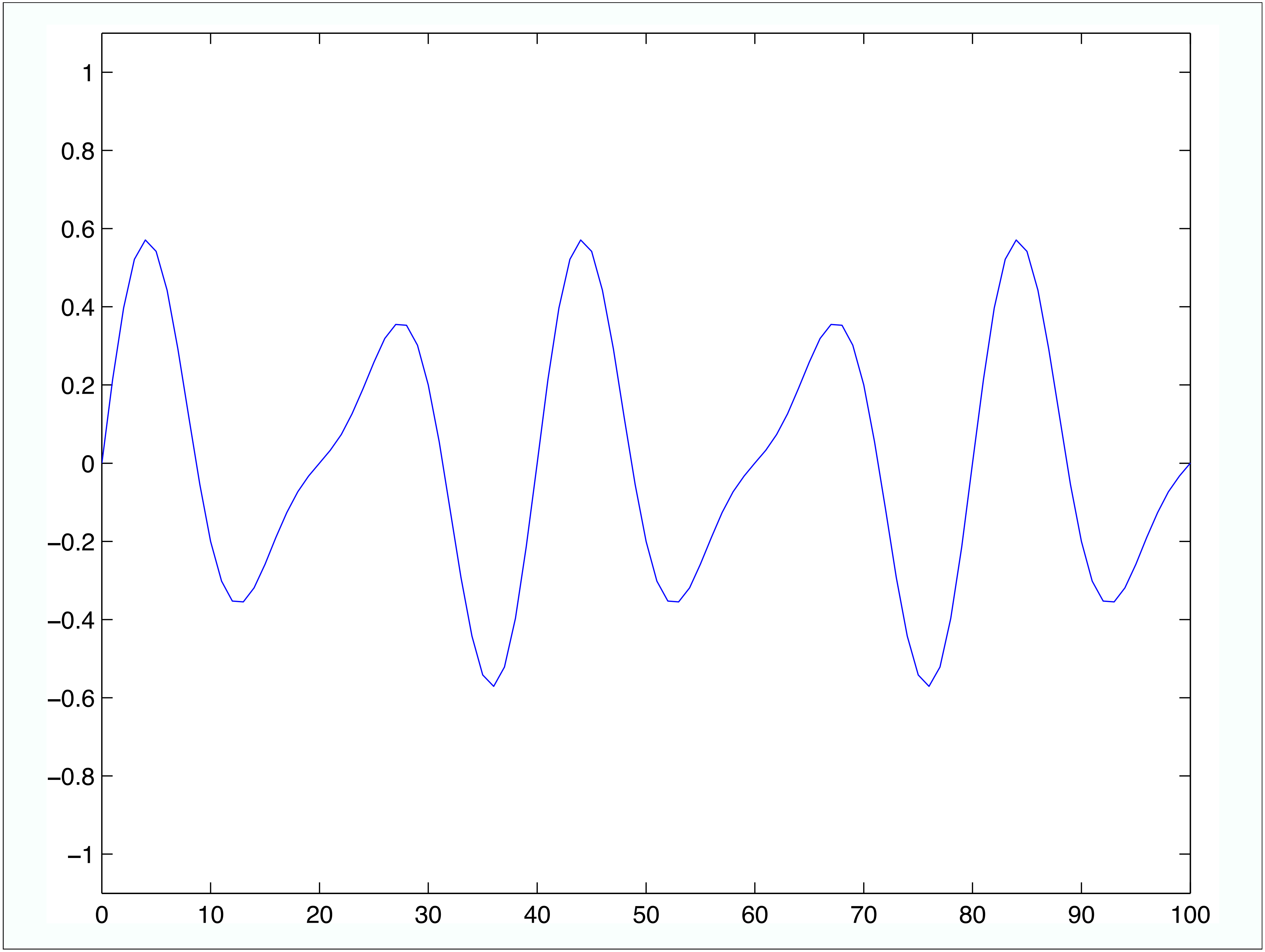


1
0.8
0.6
0.4
0.2
0
−0.2
−0.4
−0.6
−0.8
−1
0
10
20
30
40
50
60
70
80
90
100

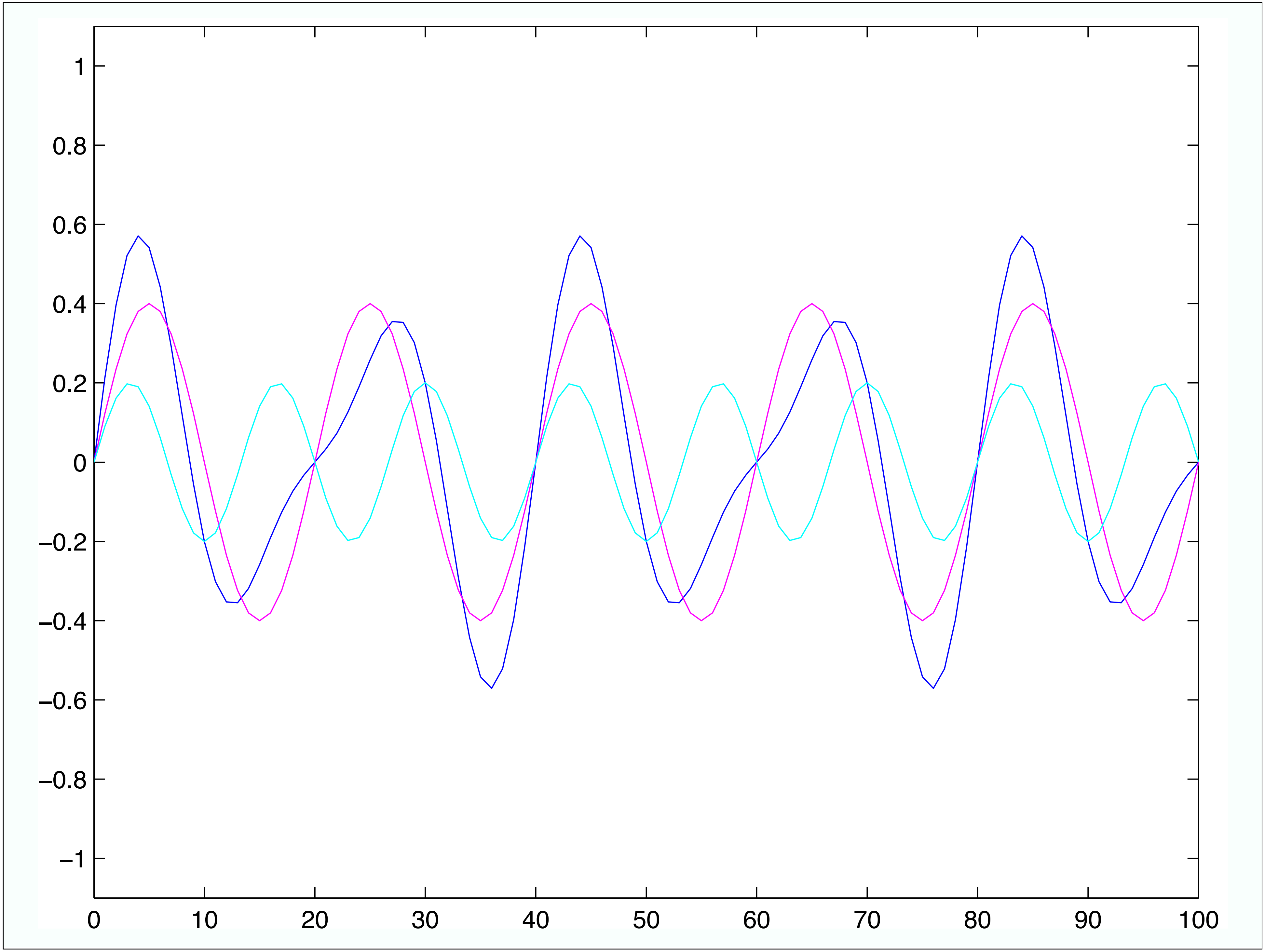

1
0.8
0.6
0.4
0.2
0
−0.2
−0.4
−0.6
−0.8
−1
0
10
20
30
40
50
60
70
80
90
100

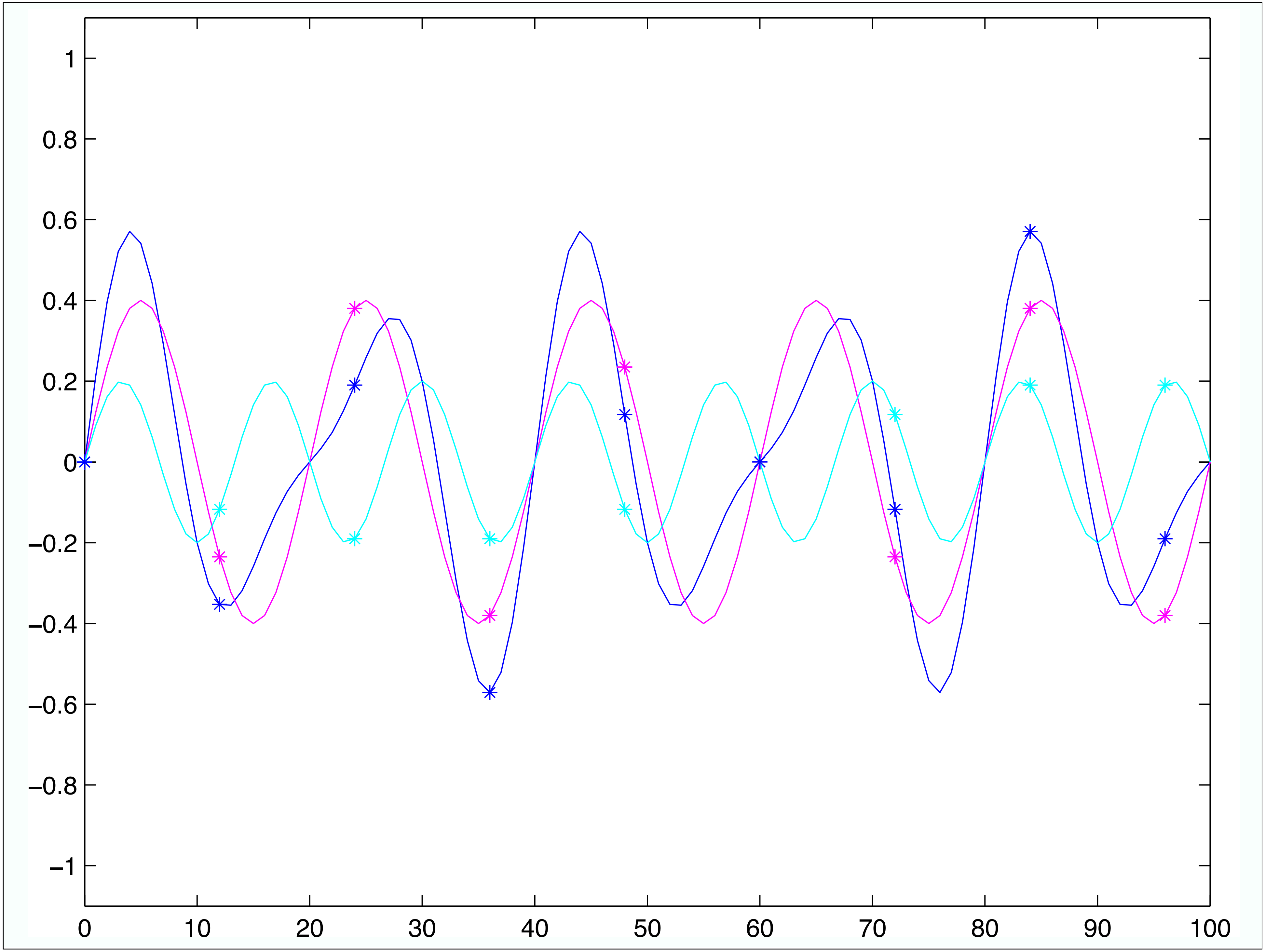
Fundamental facts

- Sample(A+B)=Sample(A)+Sample(B)
  - if a signal contains a high frequency sine wave, it will alias



1
0.8
0.6
0.4
0.2
0
−0.2
−0.4
−0.6
−0.8
−1
0
10
20
30
40
50
60
70
80
90
100


1
0.8
0.6
0.4
0.2
0
−0.2
−0.4
−0.6
−0.8
−1
0
10
20
30
40
50
60
70
80
90
100

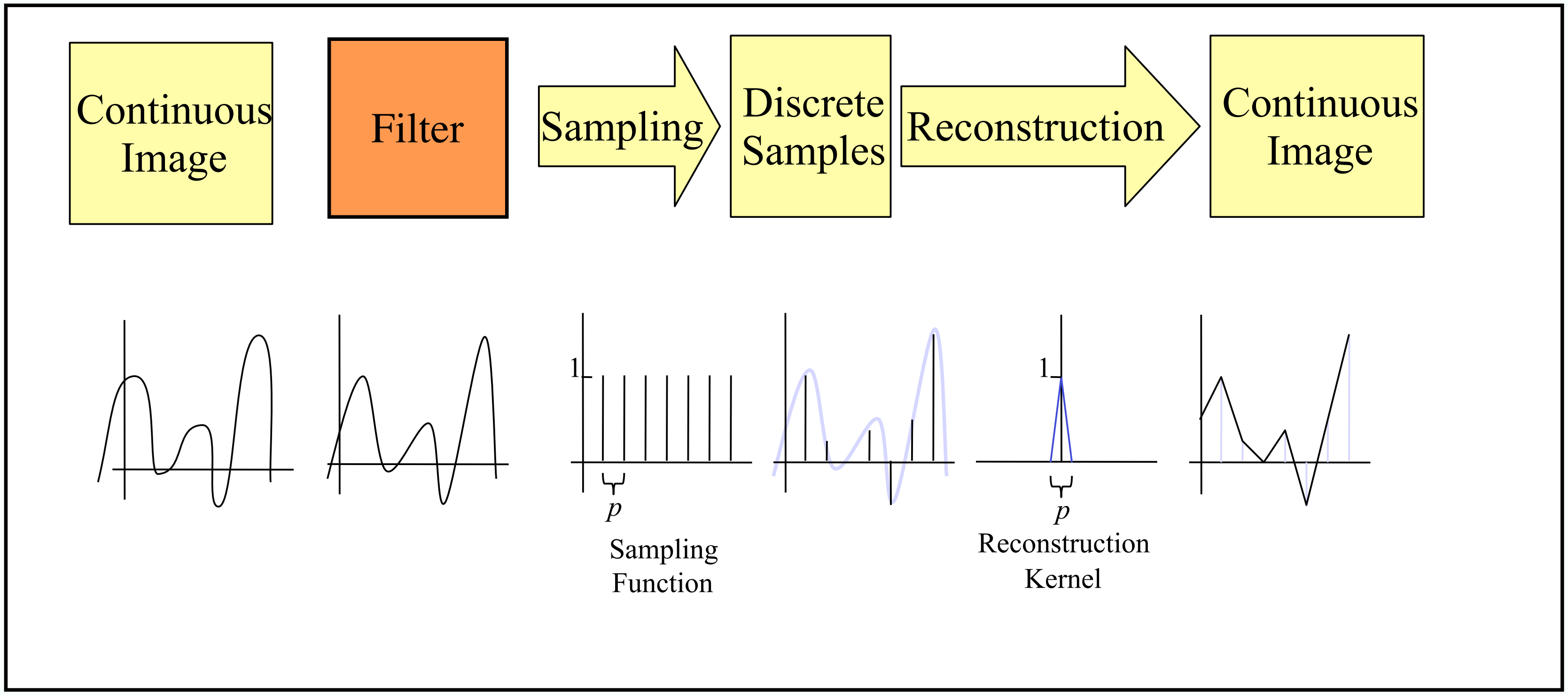


1
0.8
0.6
0.4
0.2
0
−0.2
−0.4
−0.6
−0.8
−1
0
10
20
30
40
50
60
70
80
90
100

Weapons against aliasing

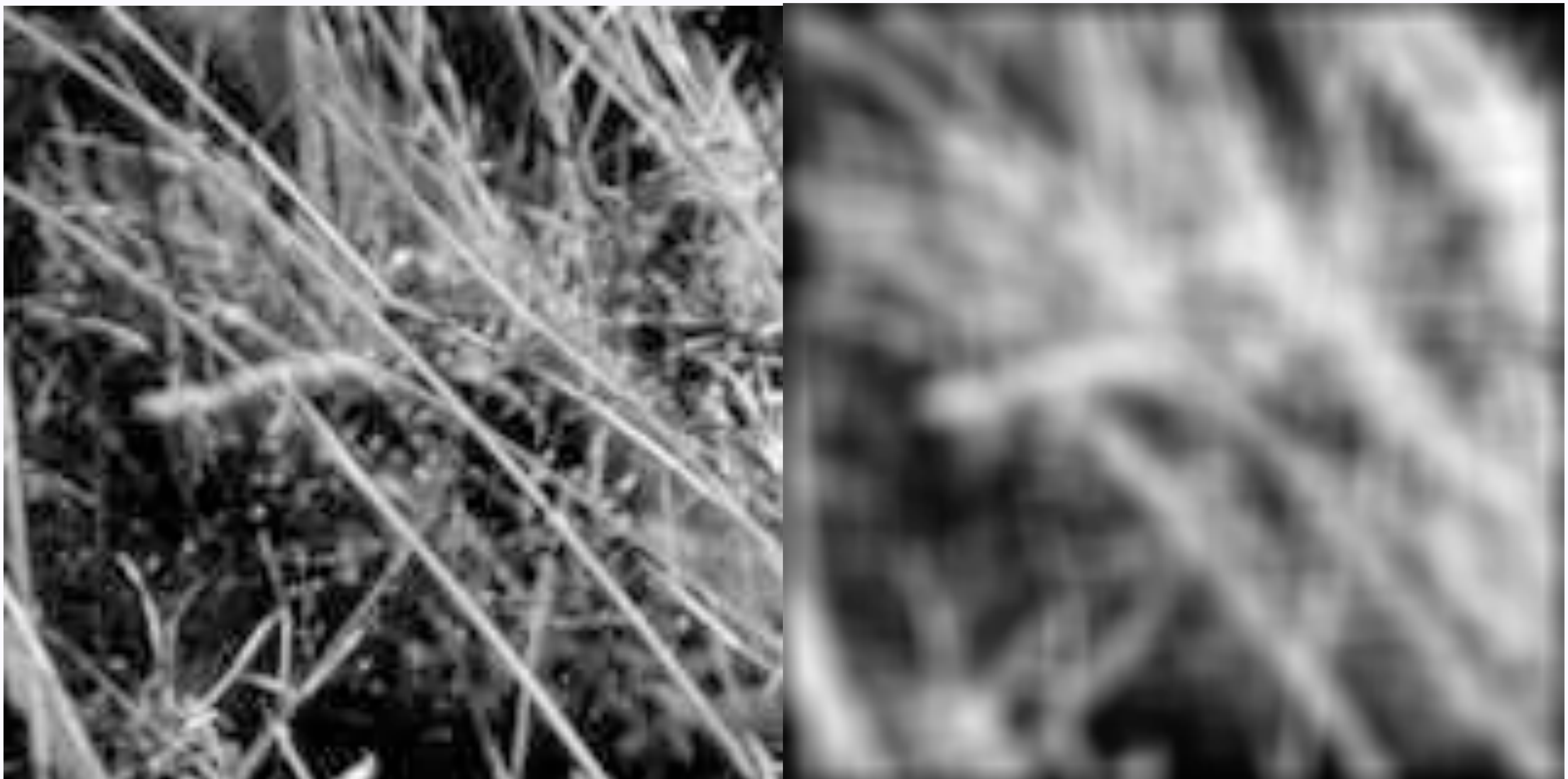
- Filtering
  - or smoothing
  - take the signal, reduce the fast-changing/high-frequency content
  - can do this by weighted local averaging

Prefiltering (Ideal case)



Usually can't do the ideal case

- Supersample
  - take samples at a finer grid than required
  - then filter those samples
  - then subsample

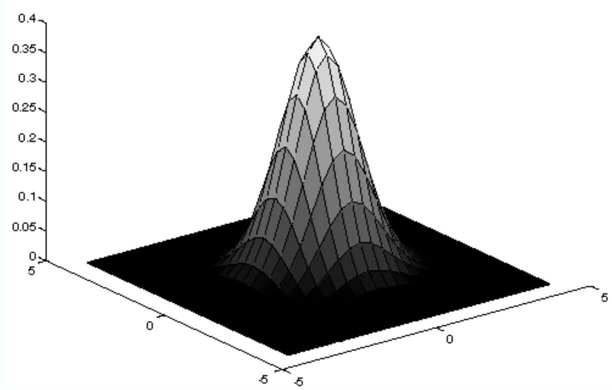


$$N_{ij} = \frac{1}{N}\Sigma_{uv}O_{i+u,j+v}$$

where u, v, is a window of N pixels in total centered at 0, 0

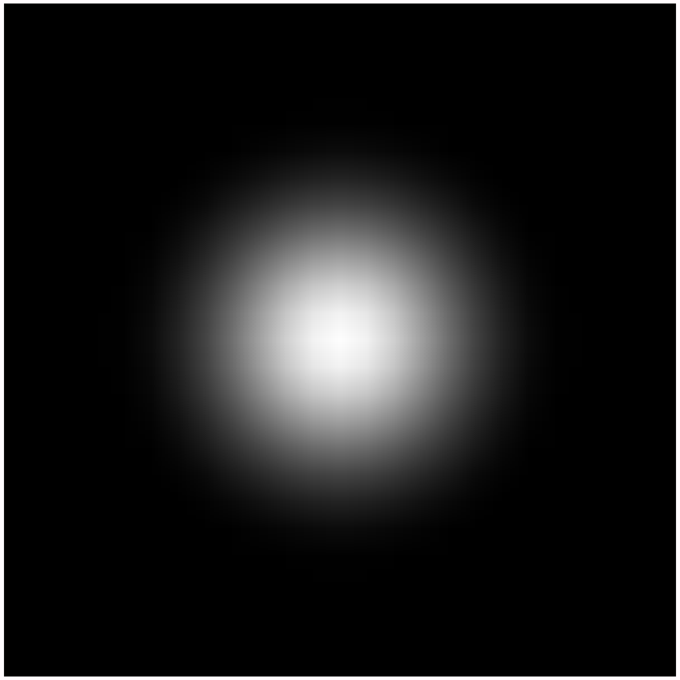
Smoothing with a Gaussian

- Notice "ringing"
  - apparently, a grid is superimposed
- Smoothing with an average actually doesn't compare at all well with a defocussed lens
  - what does a point of light produce?



• A Gaussian gives a good model of a fuzzy blob

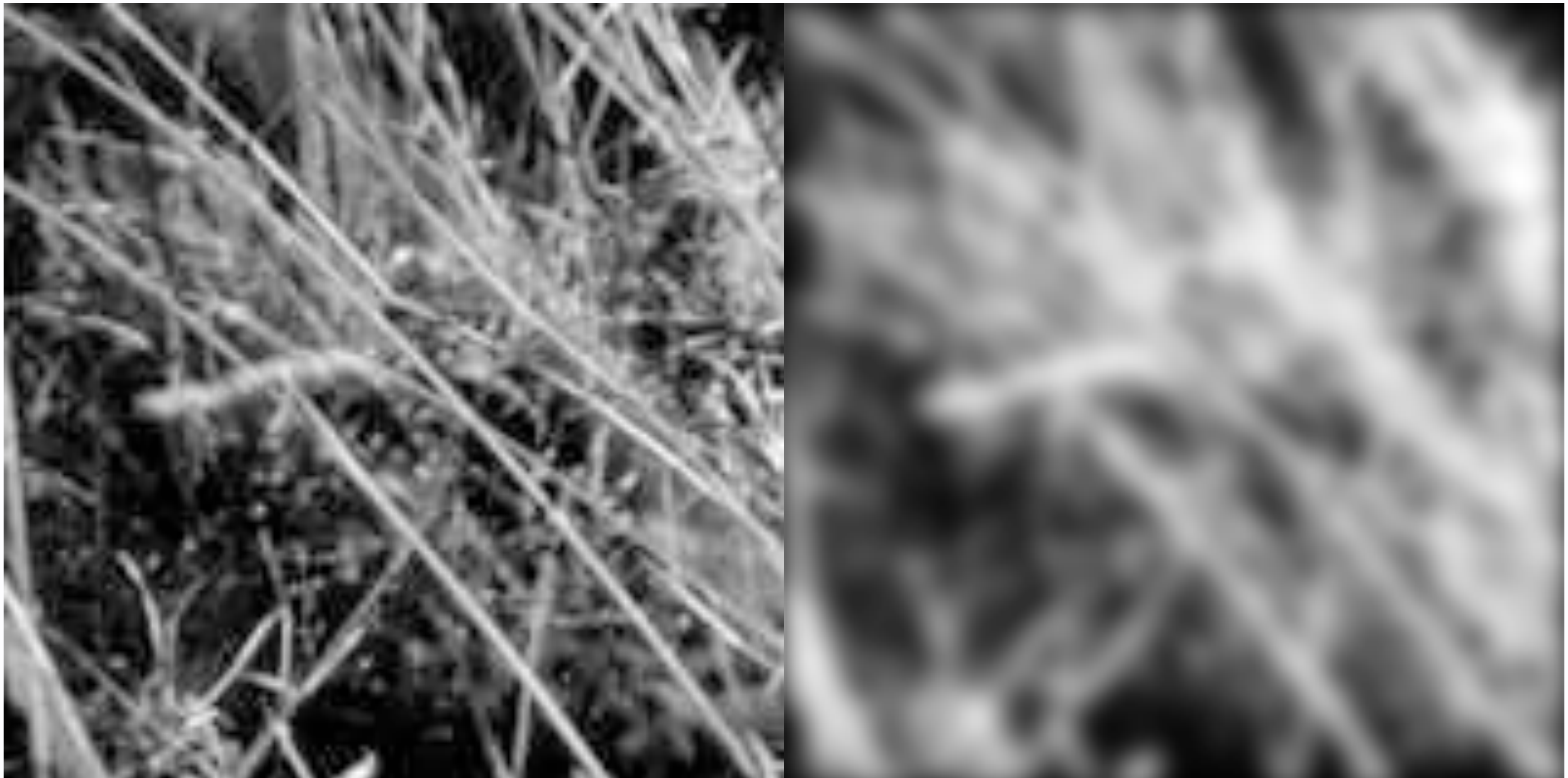
Gaussian filter kernel



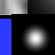
$$K_{uv} = \left(\frac{1}{2\pi\sigma^2}\right)\exp\left(\frac{-\left[u^2+v^2\right]}{2\sigma^2}\right)$$

We're assuming the index can take negative values

Smoothing with a Gaussian



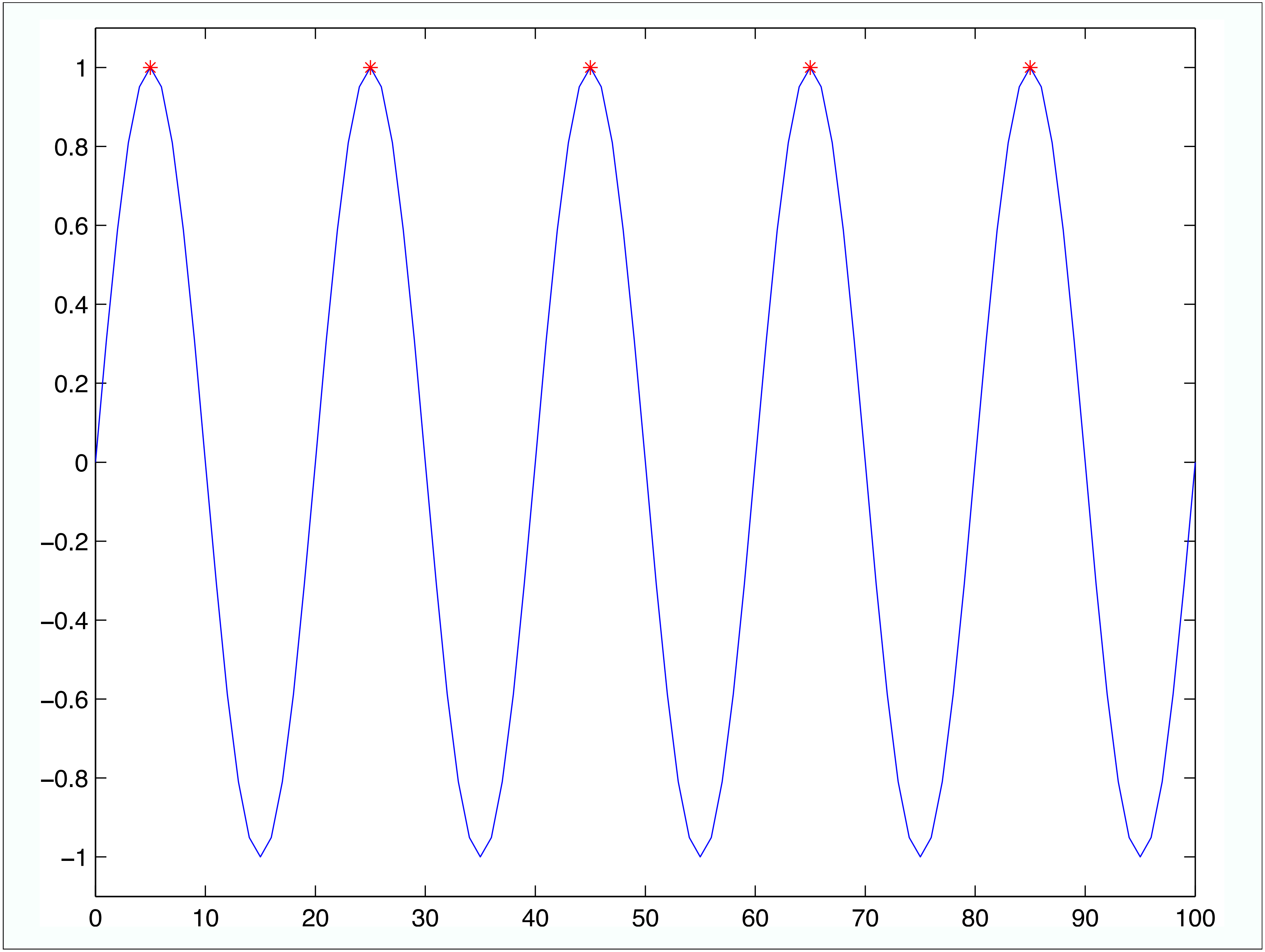
$$N_{ij} = \sum_{uv} O_{i-u,j-v} K_{uv}$$



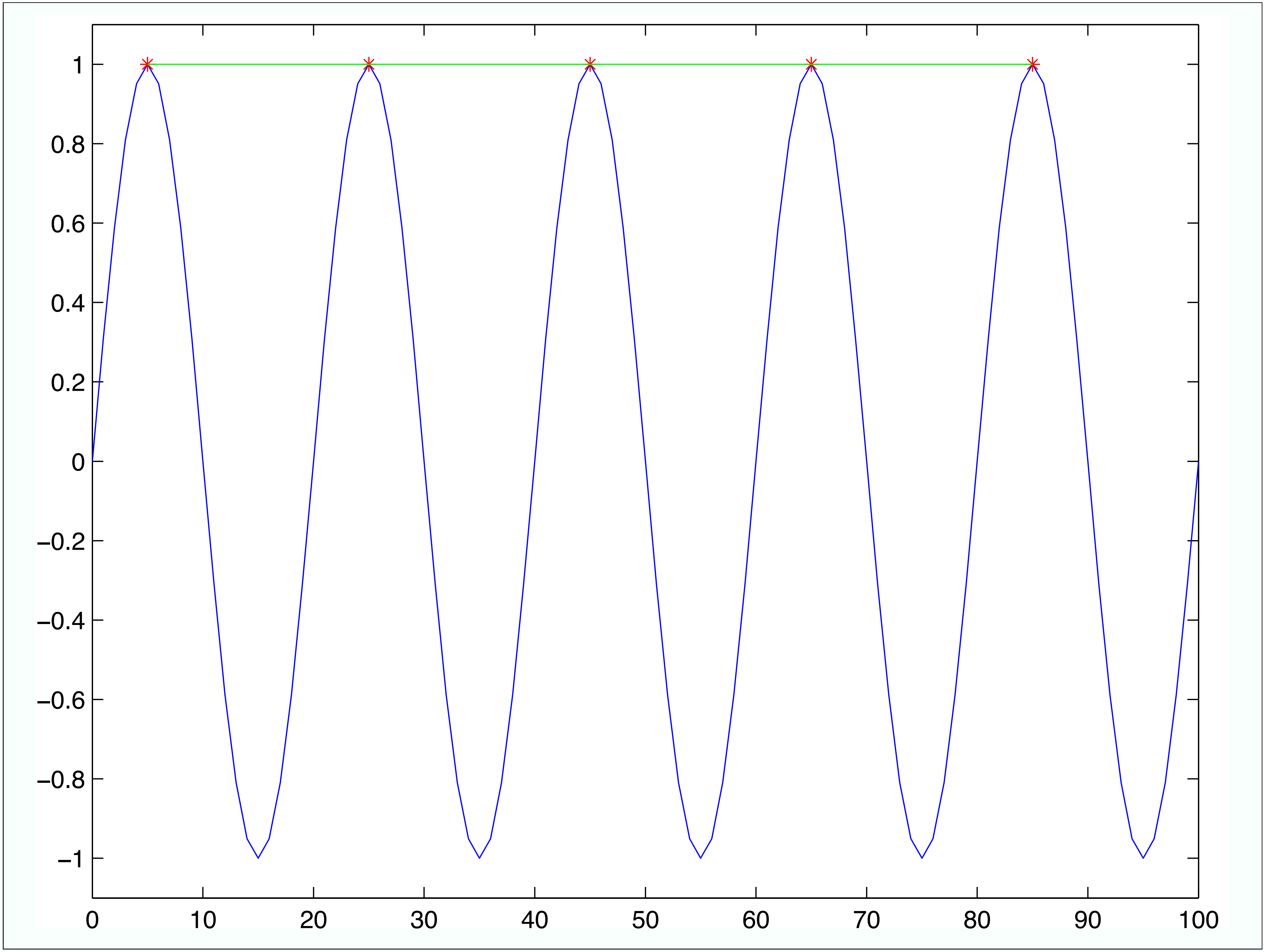
Notice the curious looking form

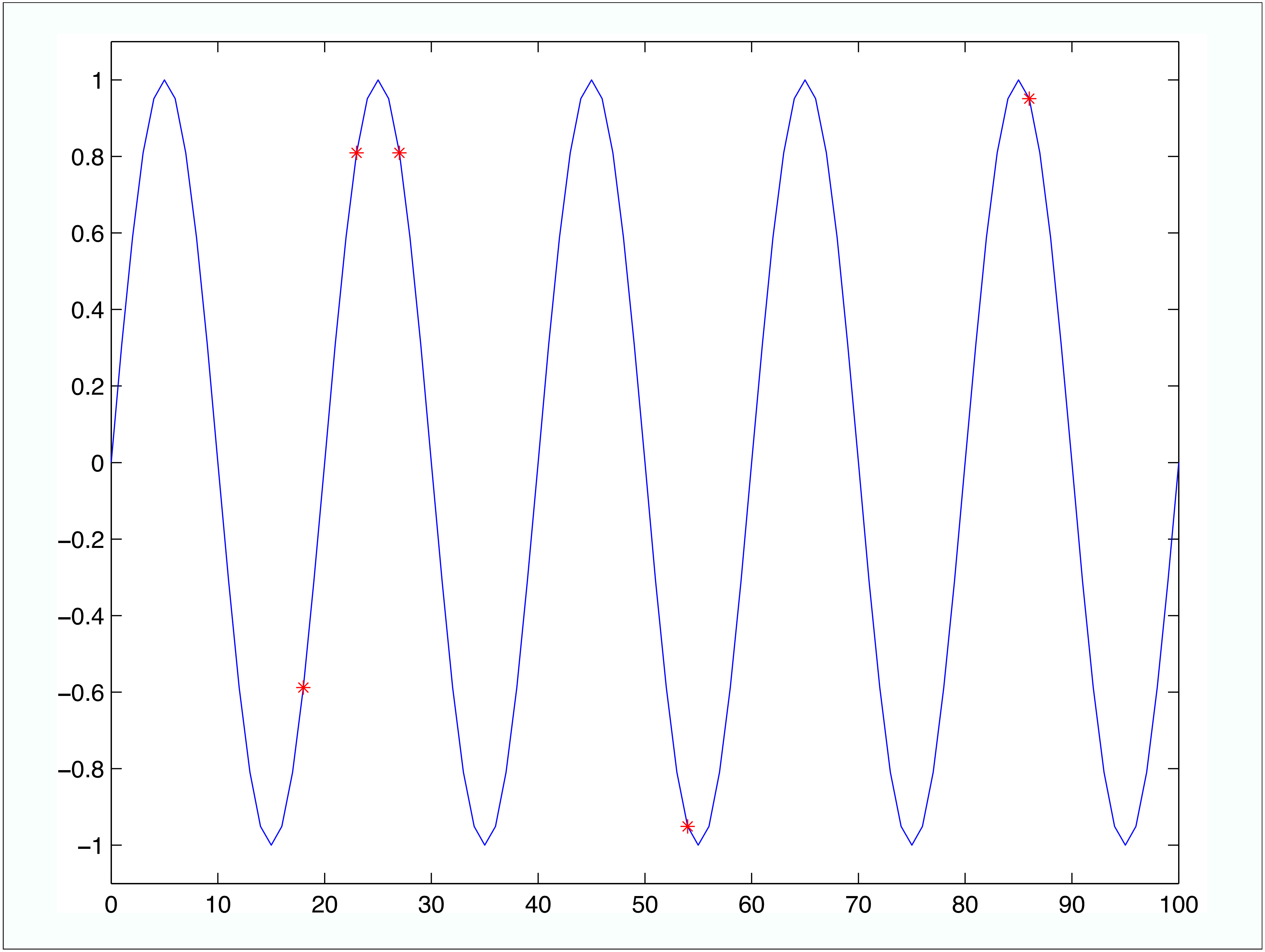
Weapons against aliasing

- Random sample locations
  - signal still aliases, but error looks like noise
    - rather than low frequency signal
    - much less offensive

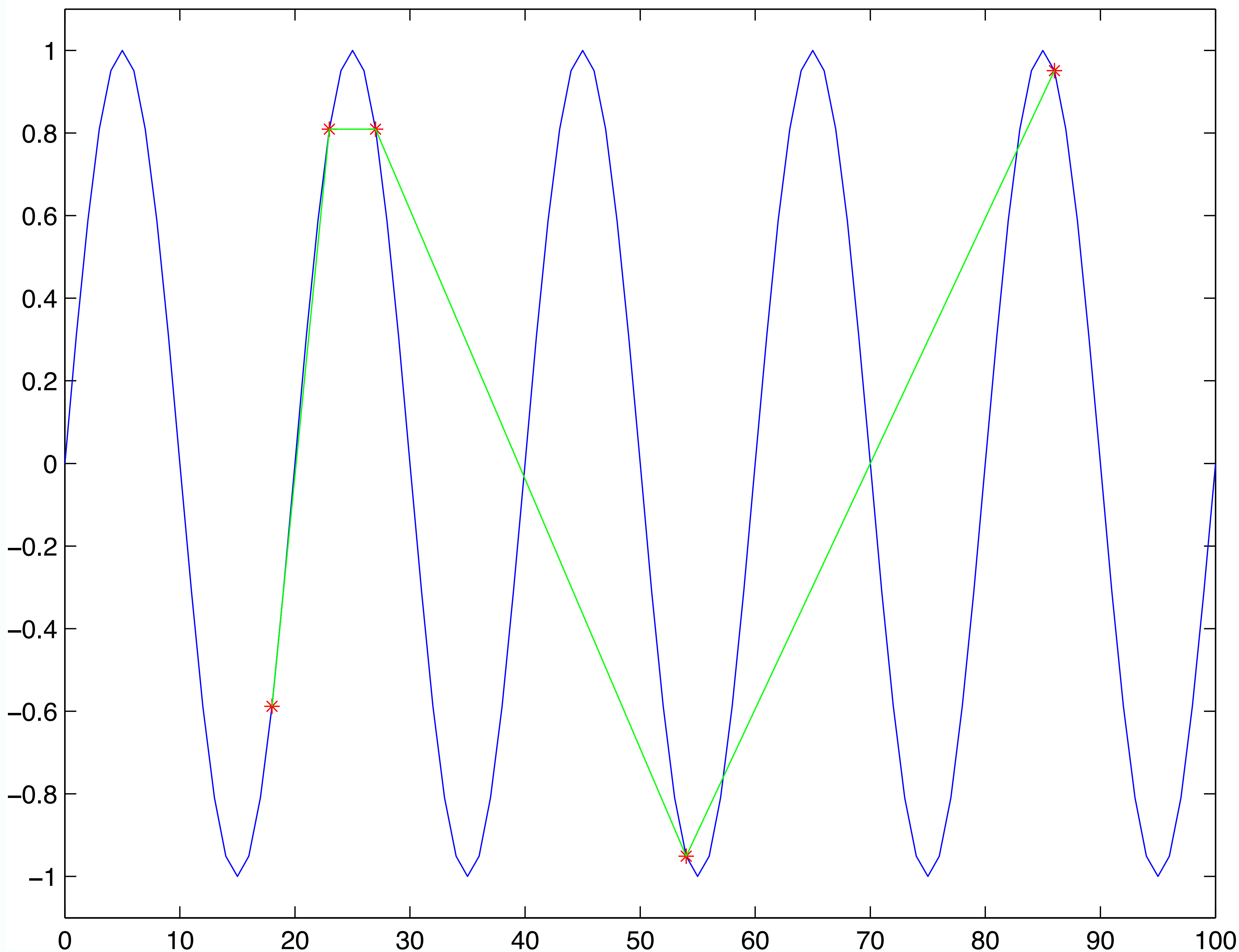


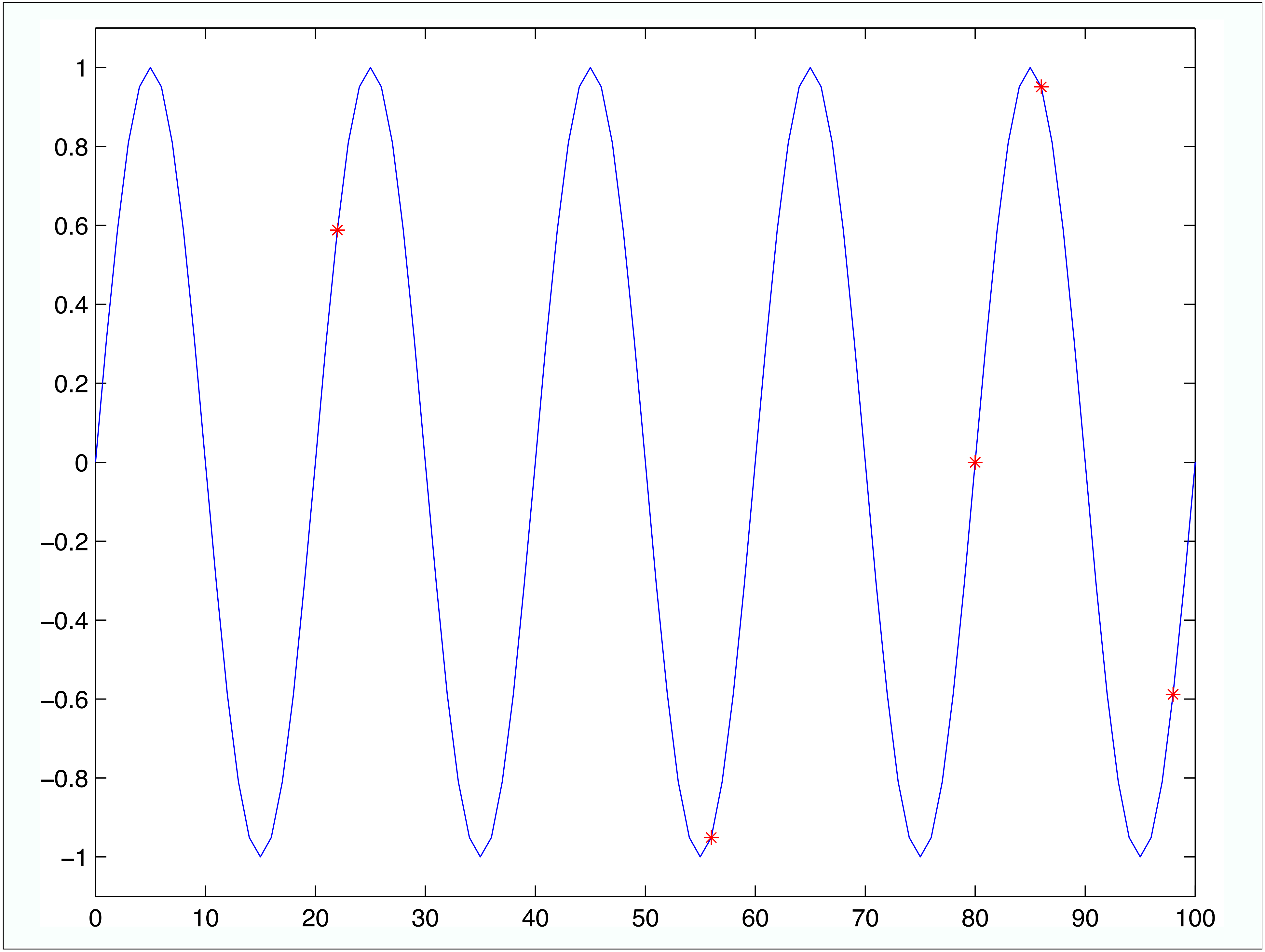
1
0.8
0.6
0.4
0.2
0
−0.2
−0.4
−0.6
−0.8
−1
0
10
20
30
40
50
60
70
80
90
100

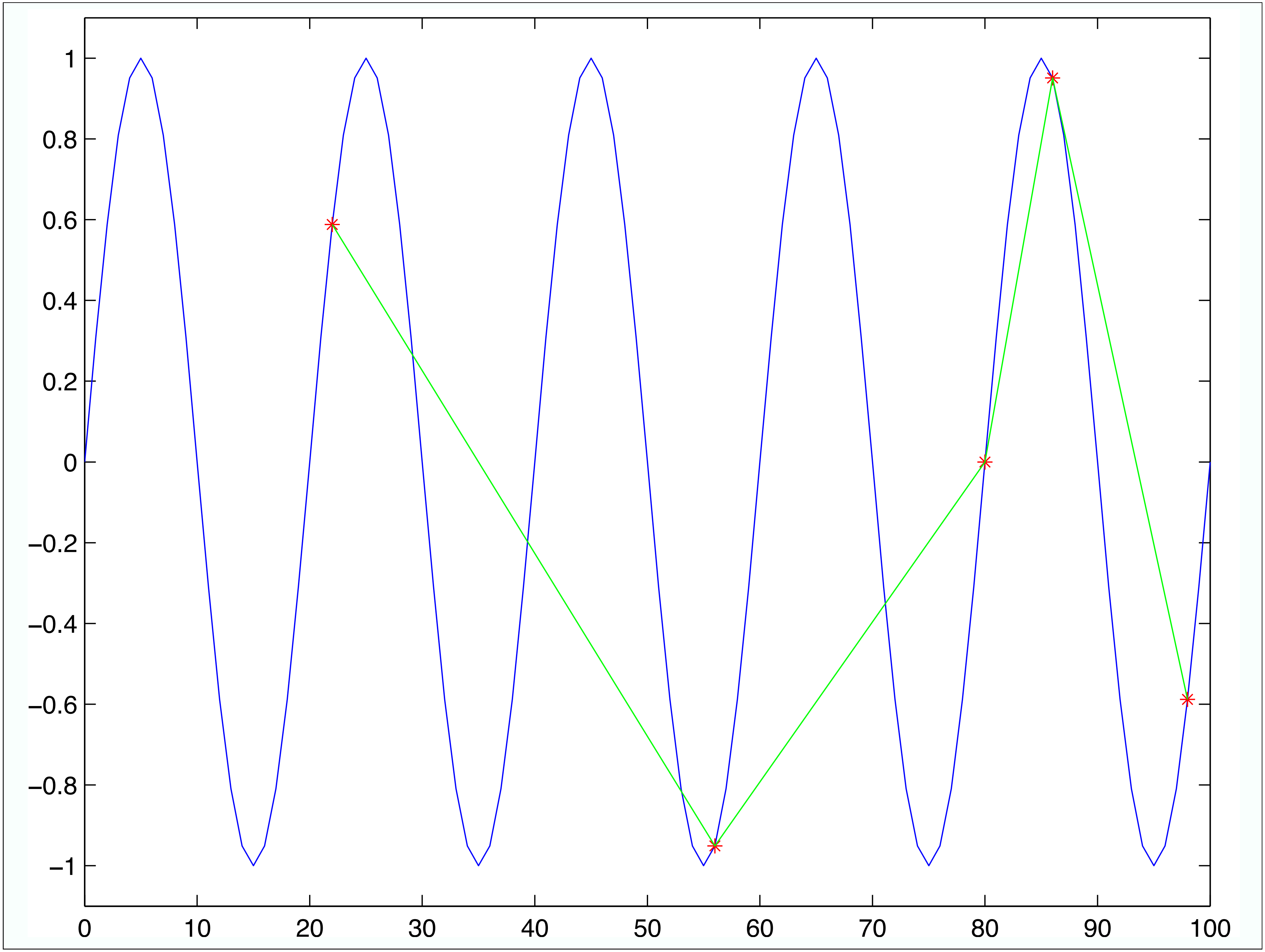

1
0.8
0.6
0.4
0.2
0
−0.2
−0.4
−0.6
−0.8
−1
0
10
20
30
40
50
60
70
80
90
100



1
0.8
0.6
0.4
0.2
0
−0.2
−0.4
−0.6
−0.8
−1
0
10
20
30
40
50
60
70
80
90
100


1
0.8
0.6
0.4
0.2
0
−0.2
−0.4
−0.6
−0.8
−1
0
10
20
30
40
50
60
70
80
90
100


1
0.8
0.6
0.4
0.2
0
−0.2
−0.4
−0.6
−0.8
−1
0
10
20
30
40
50
60
70
80
90
100

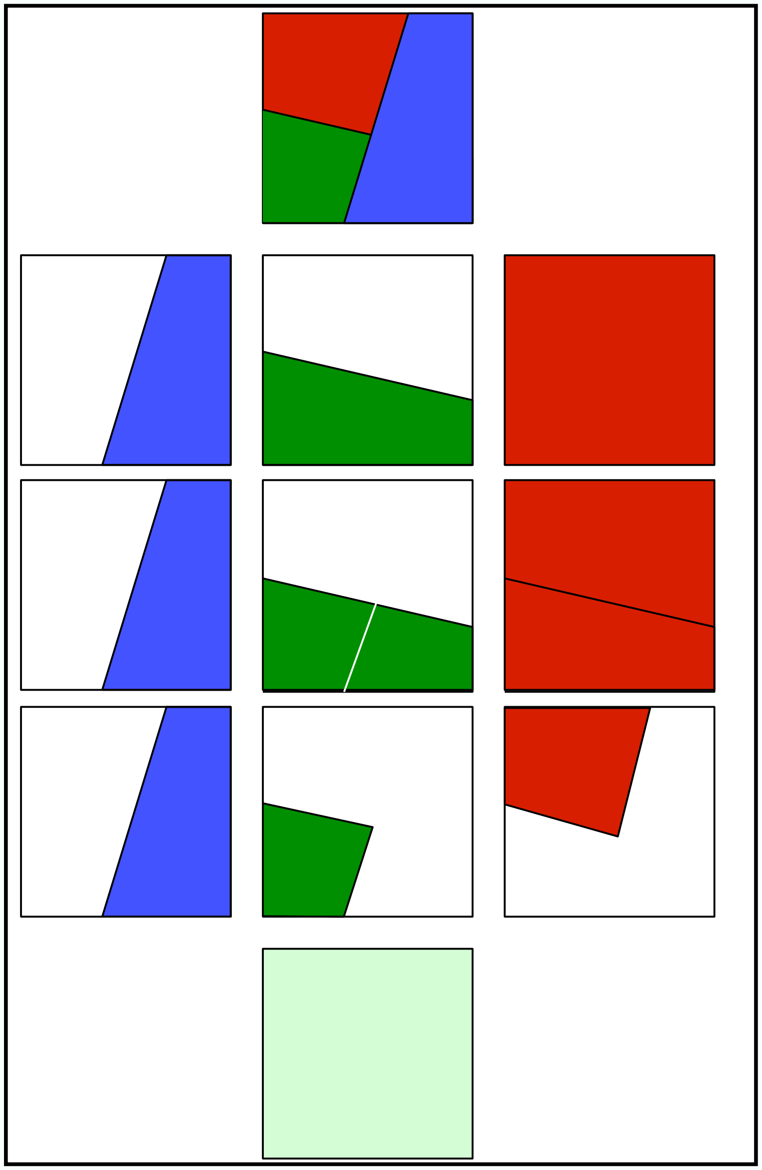

1
0.8
0.6
0.4
0.2
0
–0.2
–0.4
–0.6
–0.8
–1
0
10
20
30
40
50
60
70
80
90
100

Applying filtering

- Pixel value= Sample(smoothed illumination)
  - two ways to compute this
    - compute the smoothed illumination exactly, then sample
      - e.g. determine all fragments of polygon that overlap pixel, average
        - cone tracing
        - beam tracing
        - analytic area sampling
    - compute the smoothed illumination approximately
      - super sample, then smooth the supersampled signal

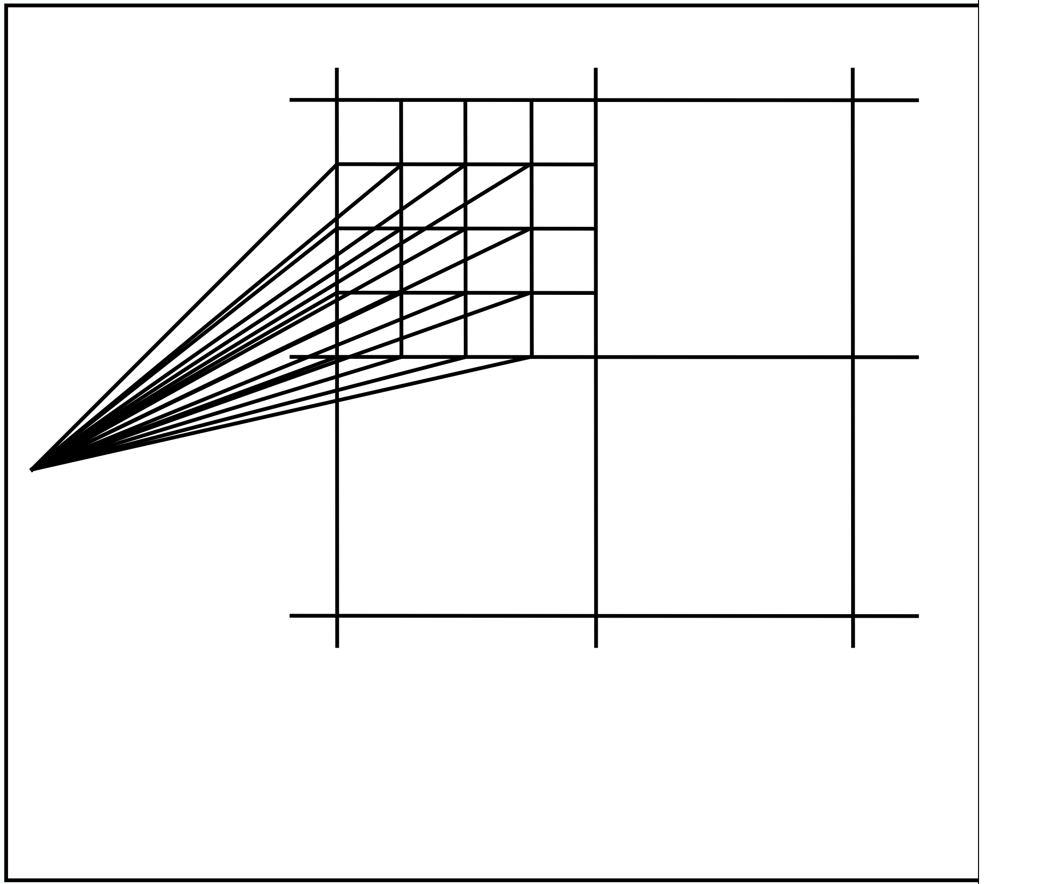
Analytic area sampling

- Ed Catmull, 1978
- Eliminates edge aliases
- Clip polygon to pixel boundary
- Sort fragments by depth
- Clip fragments against each other
- Scale color by visible area
- Sum scaled colors



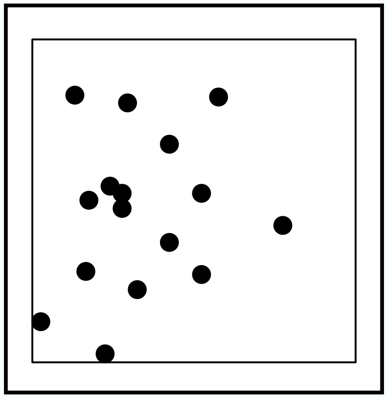
Supersampling

- Trace at higher resolution, average (filter) results
- Adaptive supersampling
  - trace at higher resolution only where necessary
- Problems
  - Does not eliminate aliases (e.g. moire patterns)
  - Makes aliases higher-frequency
  - Due to uniformity of samples



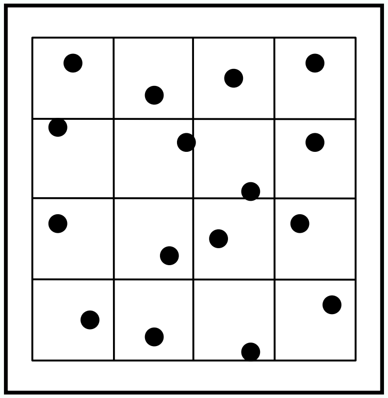
Stochastic sampling

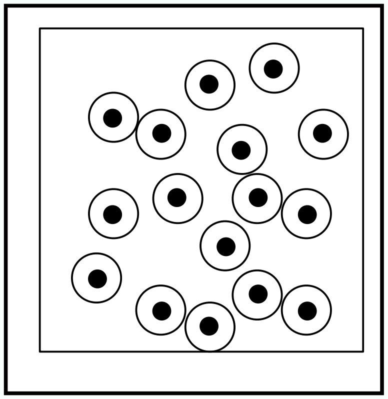
- Cast multiple rays through pixel
  - average/weighted average of results
  - at random locations
- How random?
  - uniformly at random wastes samples



Stochastic sampling

- Stratified sampling
  - uniform jitter; quasi random points
  - cut pixel into even boxes, one sample per box
- Hardcore process
  - Poisson disk
  - Pick n random points
    - k'th may not lie in disks around earlier points
      - rejection sampling
  - Samples can't cluster, but may run out of room
    - i.e. obtaining location of sample point can get hard





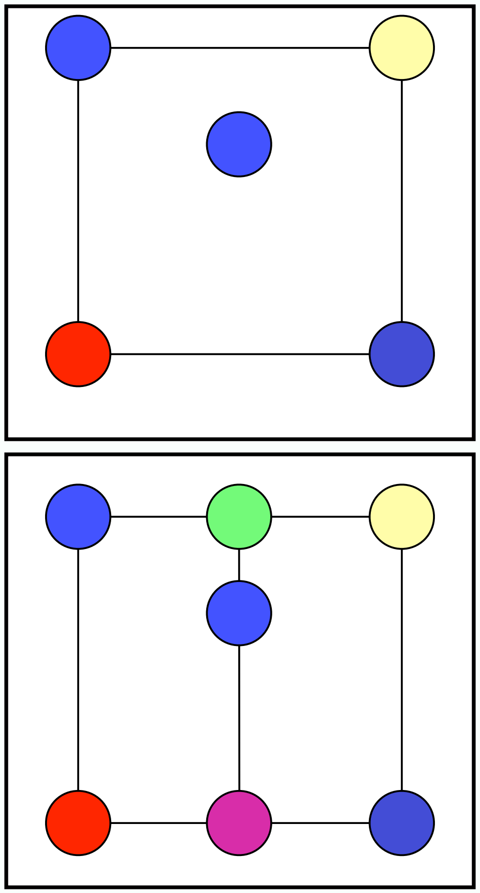
Texture Aliasing

- Image mapped onto polygon
- Occur when screen resolution differs from texture resolution
- Magnification aliasing
  - Screen resolution finer than texture resolution
  - Multiple pixels per texel
- Minification aliasing
  - Screen resolution coarser than texture resolution
  - Multiple texels per pixel

Magnification Filtering

- Nearest neighbor
  - Equivalent to spike filter
  - NEVER EVER DO THIS!

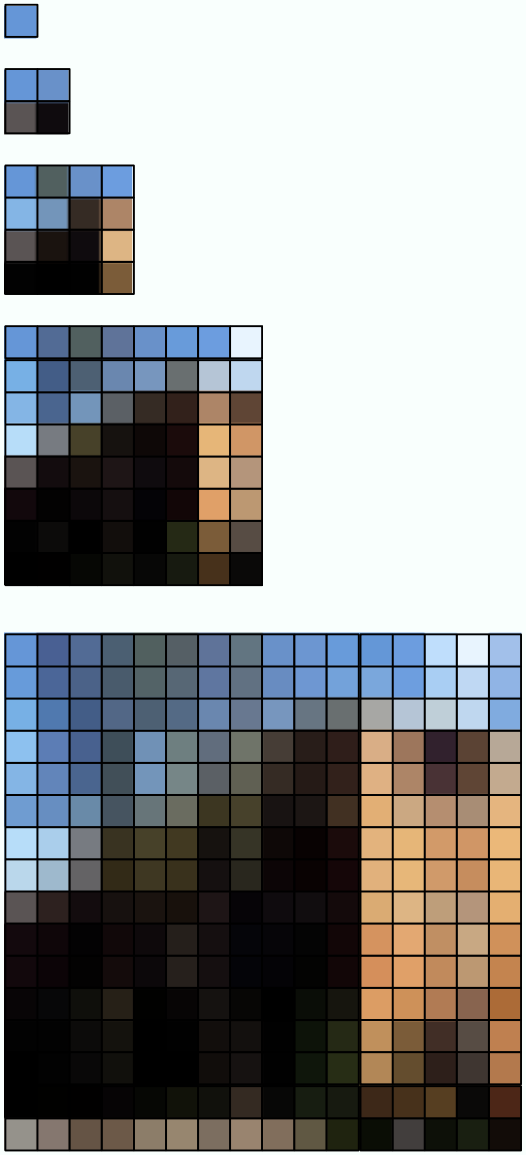
- Linear interpolation
  - Can give OK to good results



Minification Filtering

- Multiple texels per pixel
- Potential for aliasing since texture signal bandwidth greater than framebuffer
- Box filtering requires averaging of texels
- Precomputation
  - MIP Mapping
  - Summed Area Tables

MIP Mapping



- Lance Williams, 1983
- Create a resolution pyramid of textures
  - Repeatedly subsample texture at half resolution
  - Until single pixel
  - Need extra storage space
- Accessing
  - Use texture resolution closest to screen resolution
  - Or interpolate between two closest resolutions