Error Analysis

E. F. Thacher

and T.S. Whitten

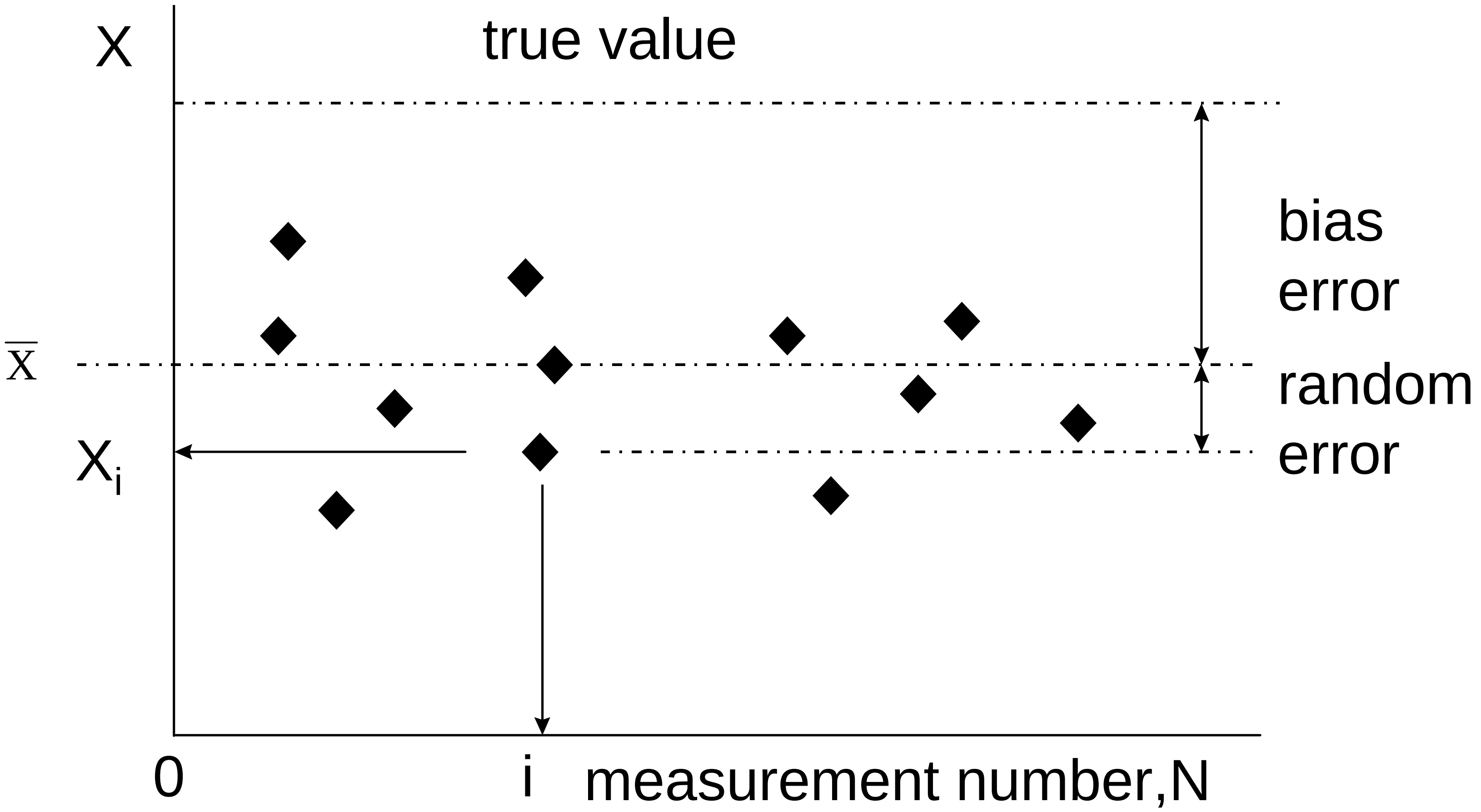
Objectives

- Learn some error analysis
- Demonstrate
  - » Overlay plots
  - » Points-only plotting
  - » Subplots
  - » Histograms
  - » Labeling

Review 1

- Random error (scatter)
  - » Random fluctuations in measurement conditions
  - » Noise introduced by signal processing
- Bias error (constant offset)
  - » Poor calibration, laboratory conditions, etc.
  - » Built into model
- Both propagate through model
-

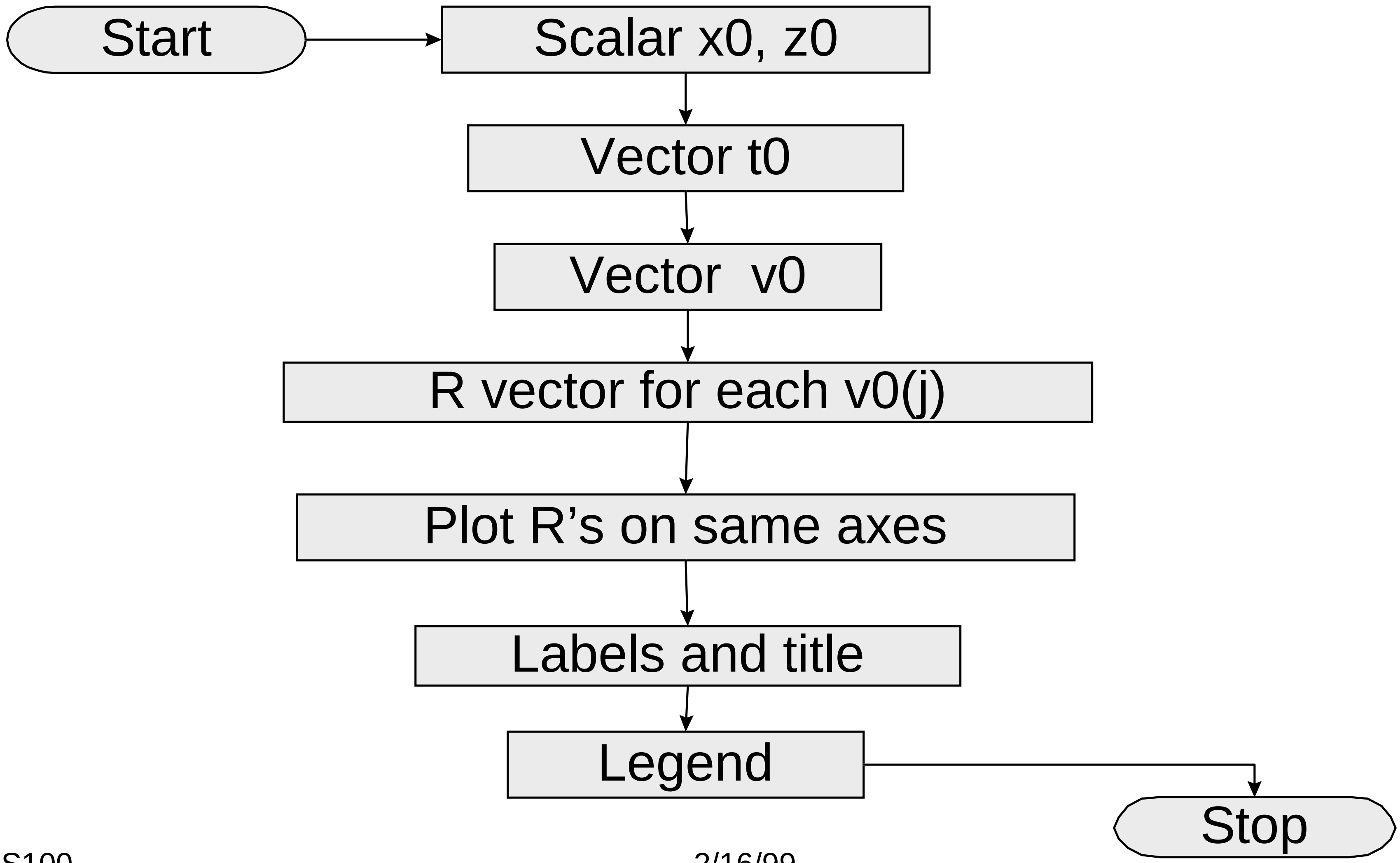
Review 2


X
true value
bias
error
$\overline{x}$
random
error
X_i
0
i
measurement number,N

Plotting Task Statement

- Write a script file called overlay1.m
- Assign initial coordinates x_0 , z_0
- Calculate the range for three different initial velocities, v_{01} , v_{02} , v_{03}
- Plot the range vs. launch angle for each initial velocity on the same figure using circles, x's and stars

Flow Chart for overlay1.m


Start

Scalar x0, z0

Vector t0

Vector v0

R vector for each v0(j)

Plot R’s on same axes

Labels and title

Legend

Stop

overlay1.m Statements

- X=[start:increment:stop]
- plot(x1,y1,['symbol1'],x2, y2,['symbol2'],…)
- xlabel,ylabel,title
- R=range0(v0,t0,x0,z0)
  - » Function to calculate range using vectors v0, t0

overlay1.m DEMO

- run overlay1.m from MATLAB command window

overlay2.m

- How do you connect the symbols?

- NOTE: symbols are usually reserved for raw data while continuous lines are used for analytical curves

overlay2.m

- How do you connect the symbols?

```
plot(t0,R1,t0,R1,'o',t0,R2,t0,R2,'x',t0,
R3,t0,R3,'Pentagram')
```

- NOTE: symbols are usually reserved for raw data while continuous lines are used for analytical curves

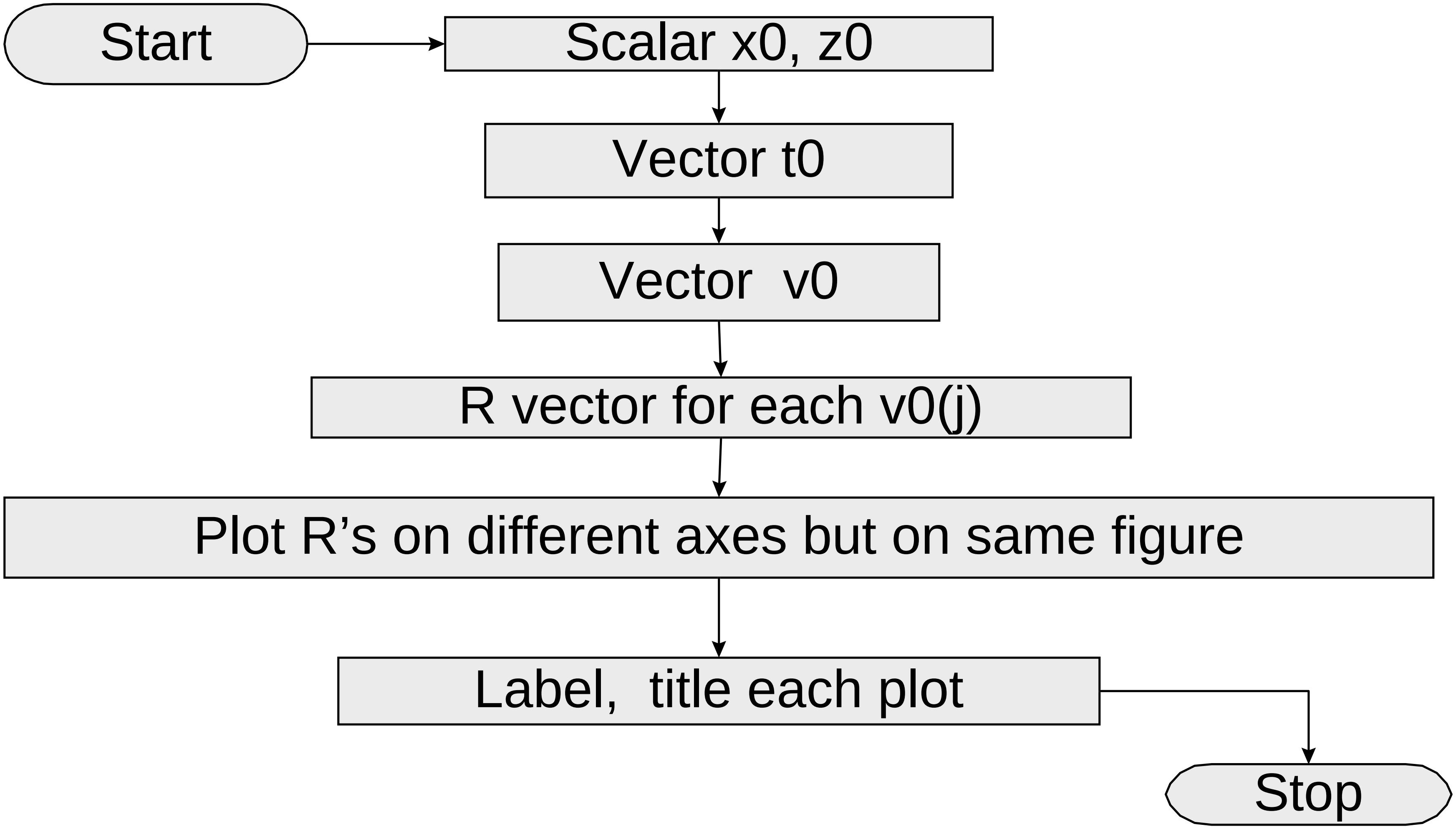
overlay2.m Demo

- run overlay2.m from MATLAB command window

Plotting Multiple Graphs In One Figure

- Write a script file called triplot.m that produces the same curves as overlay.m but uses the subplot command to split the output onto three separate graphs in one figure window

Flowchart for triplot.m



Start

Scalar x0, z0

Vector t0

Vector v0

R vector for each v0(j)

Plot R’s on different axes but on same figure

Label, title each plot

Stop

triplot.m Demonstration

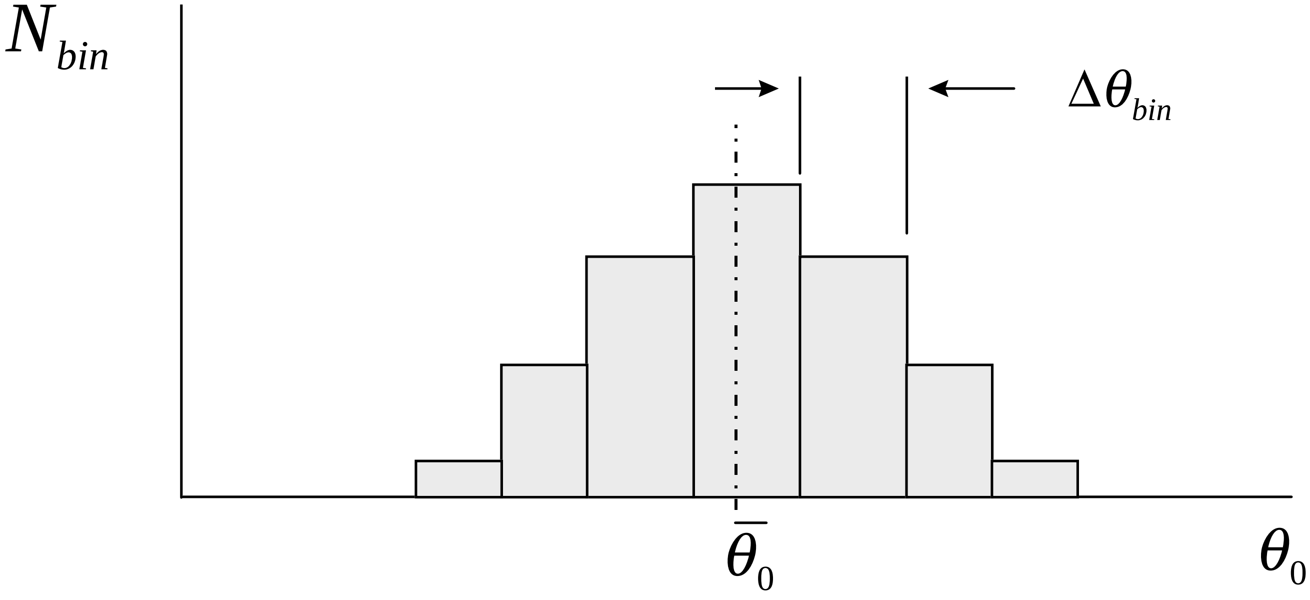
- run triplot.m from MATLAB command window

triplot.m
New Statements

```
%The following statements accomplish the
%flow chart’s objective:
 subplot (1, 3, 1)
 plot (x1, y1, [‘symbol1’] )
 xlabel ...
 ylabel ...
 title...
 subplot (1, 3, 2 )...
 subplot (1, 3, 3 )...
```

Histogram Review

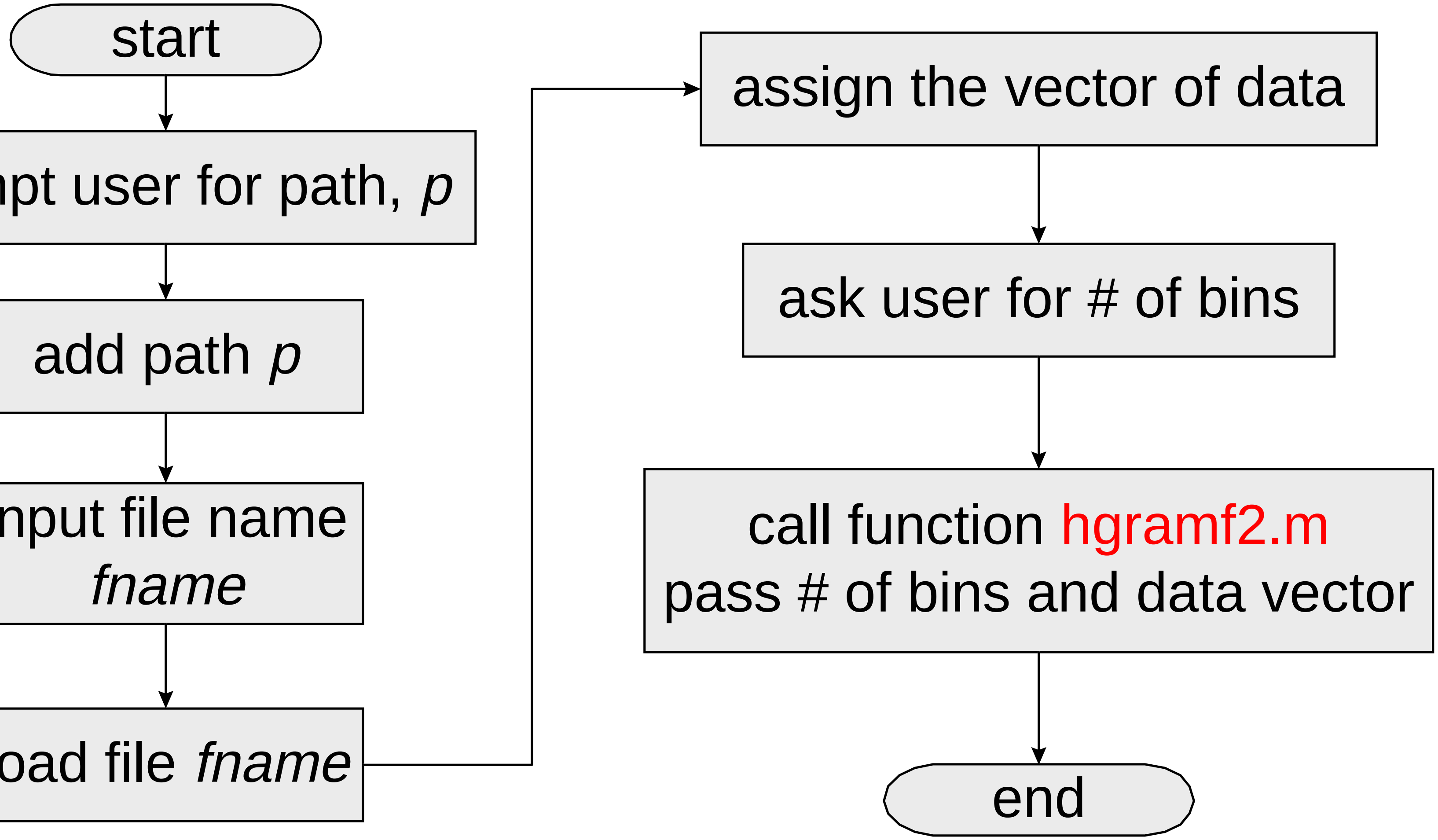
- After N measurements



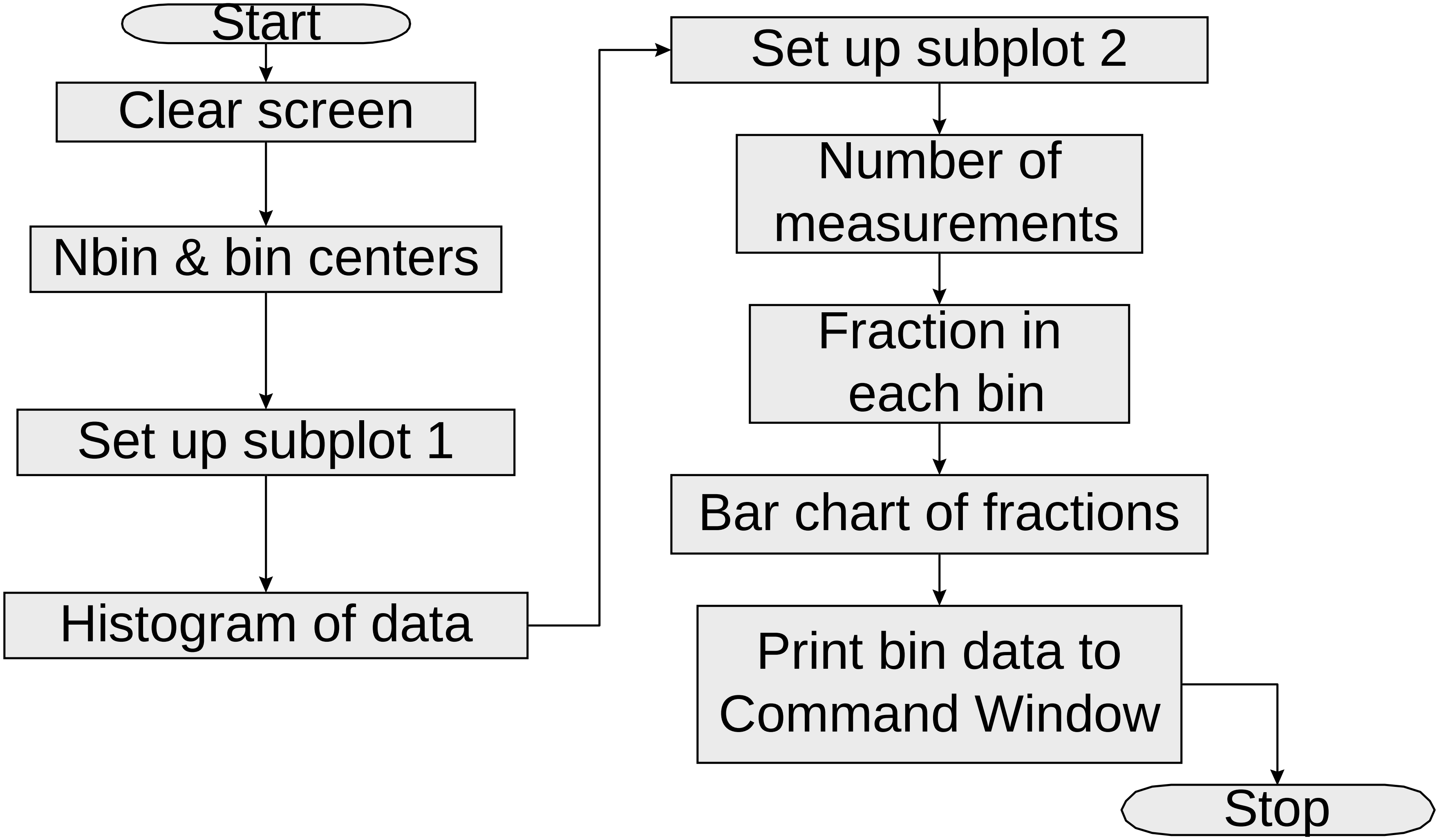
Histogram Task Statement

- Write a script file that prompts the user for a *path*, and a *data file* containing *n* data points to be plotted on a histogram with a user-specified number of bins
  - » assign inputs in command window and pass the data as a vector array and the number of bins to a function to produce a histogram

histplot.m Flowchart



Function hgramf2.m Flowchart



histplot.m New Commands

```
% Input characher string
p=input('Enter the drive path containg the
data file: ','s');
```

```
% break up file name into components in order
% to assign data vector
[path,name,ext]=fileparts(fname);
data=eval(name);
```

histplot.m New Commands

```
% Input characher string
p=input('Enter the drive path containg the
data file: ','s');
```

use this to identify the input as a string

```
% break up file nam                         in order
% to assign data vector
[path,name,ext]=fileparts(fname);
data=eval(name);
```

histplot.m New Commands

```
% Input characher string
p=input('Enter the drive path containg the
data file: ','s');
```

use this to identify the input as a string

```
% break up file nam                       in order
% to assign data vector
[path,name,ext]=fileparts(fname);
data=eval(name);
```

fileparts is a function that returns the name separate from the extension

histplot.m New Commands

```
% Input characher string
p=input('Enter the drive path containg the
data file: ','s');
```

use this to identify the input as a string

```
 % break up file nam                            in order
 % to assign data vector
 [path,name,ext]=fileparts(fname);
 data=eval(name);
```

fileparts is a function that returns the name the

use the eval function to assign the number contained in *name* to *data*

hgramf2.m New Statements

```
%Obtaining number/bin and bin
%centers:
[n, bin_centers] = hist (vector, m);
%Plotting the histogram:
hist (vector, m);
```

```
%Finding the number of measurements and
%the fraction in each bin:
num_meas = length (vector);
frequency = n/num_meas;
```

hgramf2.m New Statements(cont.)

```
%Printing the bin data:
fprintf( ‘\n There were
          %3.0fmeasurements.\n\n',
          num_meas);
disp(' bin Center (psi)
        count frequency')
%You have to put them in an array.
A=[bin_centers;n;frequency];
%Blanks left for orderly appearance.
fprintf('            %4.3f
              %2.0f    %6.4f\n', A)
%Note: MATLAB takes the transpose of A
%when printing.
```

histplot.m Demonstration

- run histplot.m in MATLAB command window

!!Extra Credit!!

- On a sheet of paper, describe the differences or similarities between script m-files, function m-files, and MATLAB commands such as plot