[redacted] BACKGROUND [redacted]

BAUER

HW; 2, 3

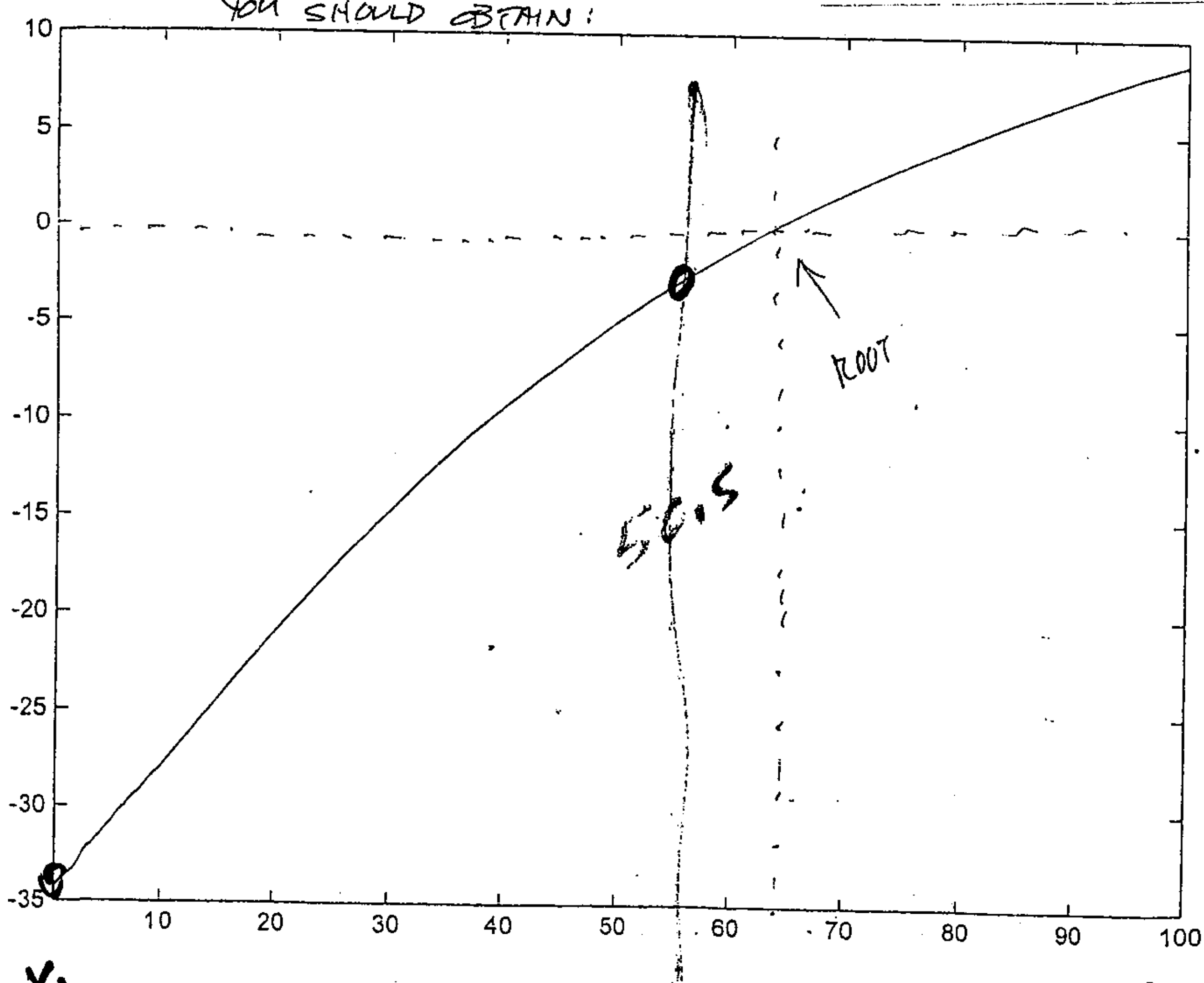
1) CREATE An "m-file" definition of the function you want to process. For problem # 5.12, p. 131 in Chapra & Canale, this would look like:

```
function y=f(x)
g=9.8;
c=14.;
v=35.;
t=7.;
y=(g*x/c)*(1-exp(-(c/x)*t))-v;
```

2) save this on the C-drive or your floppy disk.

3) use a statement like ">> fplot('f',[1,100]) (CR)"

YOU SHOULD OBTAIN:



X_L

```
bauer cut at bisection function root finder...
inction dummy= bisect(xl,xu,es,imax,xr,iter,ea)
                         in=[xl,xu,es,imax,xr,iter,ea];
sp(in)
:er=0;
nile iter<imax
   xrold=xr;
   xr=(xl+xu)/2;
   iter=iter+1;
   if xr~=0
        ea=abs((xr-xrold)/xr)*100;
   end
   test=f(xl)*f(xr);
    disp(test)
   if test<0
        xu=xr;
   elseif test>0
        xl=xr;
   end
   out=[xl,f(xl),xr,f(xr),xu,f(xu),ea,iter];
   disp(out)
   bisect=xr;
nd
```

Not needed

NOTE THIS VERSION DOES NOT HAVE EA CUTOFF CHECK AGAINST ES

ALSO, THIS VERSION DOES NOT CHECK (X_L, X_U) INPUTS FOR VALID ROOT INTERVAL

```
> bisect(1,100,.1,100,0,0,1000)
1.0e+003 *

0.0010    0.1000    0.0001    0.1000    0    0    1.0000
```

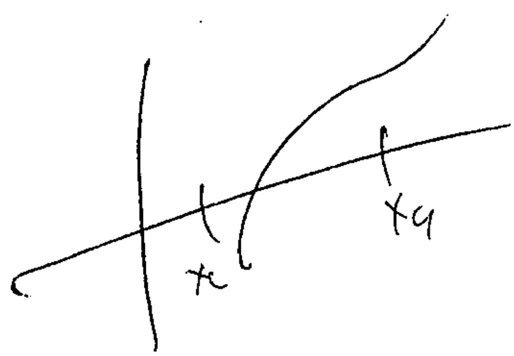
ECHO OF INPUT

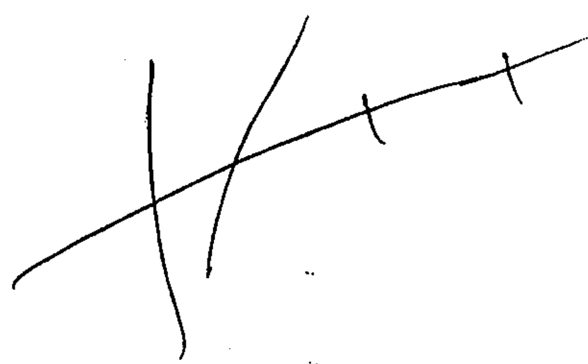
xL	f(xL)	xR	f(xR)	xu	f(xu)	εa	ITER
50.5000	-4.7269	50.5000	-4.7269	100.0000	8.7282	100.0000	1.0000
50.5000	-4.7269	75.2500	3.3527	75.2500	3.3527	32.8904	2.0000
62.8750	-0.2486	62.8750	-0.2486	75.2500	3.3527	19.6819	3.0000
62.8750	-0.2486	69.0625	1.6467	69.0625	1.6467	8.9593	4.0000
62.8750	-0.2486	65.9688	0.7244	65.9688	0.7244	4.6897	5.0000
62.8750	-0.2486	64.4219	0.2445	64.4219	0.2445	2.4012	6.0000
63.6484	-0.0004	63.6484	-0.0004	64.4219	0.2445	1.2152	7.0000
63.6484	-0.0004	64.0352	0.1225	64.0352	0.1225	0.6039	8.0000
63.6484	-0.0004	63.8418	0.0611	63.8418	0.0611	0.3029	9.0000
63.6484	-0.0004	63.7451	0.0304	63.7451	0.0304	0.1517	10.0000
63.6484	-0.0004	63.6968	0.0150	63.6968	0.0150	0.0759	11.0000
63.6484	-0.0004	63.6726	0.0073	63.6726	0.0073	0.0380	12.0000
63.6484	-0.0004	63.6605	0.0035	63.6605	0.0035	0.0190	13.0000
63.6484	-0.0004	63.6545	0.0015	63.6545	0.0015	0.0095	14.0000
63.6484	-0.0004	63.6515	0.0006	63.6515	0.0006	0.0047	15.0000
63.6484	-0.0004	63.6499	0.0001	63.6499	0.0001	0.0024	16.0000
63.6492	-0.0002	63.6492	-0.0002	63.6499	0.0001	0.0012	17.0000
63.6496	-0.0000	63.6496	-0.0000	63.6499	0.0001	0.0006	18.0000
63.6496	-0.0000	63.6498	0.0000	63.6498	0.0000	0.0003	19.0000
63.6497	-0.0000	63.6497	-0.0000	63.6498	0.0000	0.0001	20.0000
63.6497	-0.0000	63.6497	0.0000	63.6497	0.0000	0.0001	21.0000
63.6497	-0.0000	63.6497	-0.0000	63.6497	0.0000	0.0000	22.0000
63.6497	-0.0000	63.6497	0.0000	63.6497	0.0000	0.0000	23.0000
63.6497	-0.0000	63.6497	0.0000	63.6497	0.0000	0.0000	24.0000
63.6497	-0.0000	63.6497	-0.0000	63.6497	0.0000	0.0000	25.0000
63.6497	-0.0000	63.6497	0.0000	63.6497	0.0000	0.0000	26.0000

Continues for 100 iterations

```
er cut at bisection function root finder...
tion dummy= bisect(xl,xu,es,imax,xr,iter,ea)
                   in=[xl,xu,es,imax,xr,iter,ea];
(in)
=0;
e (iter<imax)  &  (ea>es)
 xrold=xr;
 xr=(xl+xu)/2;
 iter=iter+1;
 if xr~=0
     ea=abs((xr-xrold)/xr)*100;
 end
 test=f(xl)*f(xr);
  disp(test)
 if test<0
     xu=xr;
 elseif test>0
     xl=xr;
 end
 out=[xl,f(xl),xr,f(xr),xu,f(xu),ea,iter];
 disp(out)
 bisect=xr;
```

Adds ε_A vs ε_S test





```
bisect(1,100,.1,100,0,0,1000)
1.0e+003 *
```

0.0010	0.1000	0.0001	0.1000	0	0	1.0000	
50.5000	-4.7269	50.5000	-4.7269	100.0000	8.7282	100.0000	1.0000
50.5000	-4.7269	75.2500	3.3527	75.2500	3.3527	32.8904	2.0000
62.8750	-0.2486	62.8750	-0.2486	75.2500	3.3527	19.6819	3.0000
62.8750	-0.2486	69.0625	1.6467	69.0625	1.6467	8.9593	4.0000
62.8750	-0.2486	65.9688	0.7244	65.9688	0.7244	4.6897	5.0000
62.8750	-0.2486	64.4219	0.2445	64.4219	0.2445	2.4012	6.0000
63.6484	-0.0004	63.6484	-0.0004	64.4219	0.2445	1.2152	7.0000
63.6484	-0.0004	64.0352	0.1225	64.0352	0.1225	0.6039	8.0000
63.6484	-0.0004	63.8418	0.0611	63.8418	0.0611	0.3029	9.0000
63.6484	-0.0004	63.7451	0.0304	63.7451	0.0304	0.1517	10.0000
63.6484	-0.0004	63.6968	0.0150	63.6968	0.0150	0.0759	11.0000

REQUIRED WORK —

a) MODIFY THIS PROGRAM TO INCLUDE AN INITIAL CHECK TO SEE IF YOUR SUPPLIED (XL, XU) VALUES DEFINE AN INTERVAL CONTAINING A ROOT.

b) RUN AN EXAMPLE WHERE THE VALUES DO NOT CONTAIN A ROOT AND DEMONSTRATE THAT THE CODE DETECTS THIS CASE.

EGN 3420 [illegible]
HMWK #3

DR BAUER

(X)NIS X SI X SI7_

PROBLEM FROM
Numerical Method for Engineers,
Chapra + Canale, 5th Edition,
McGRAW-HILL, 2006.

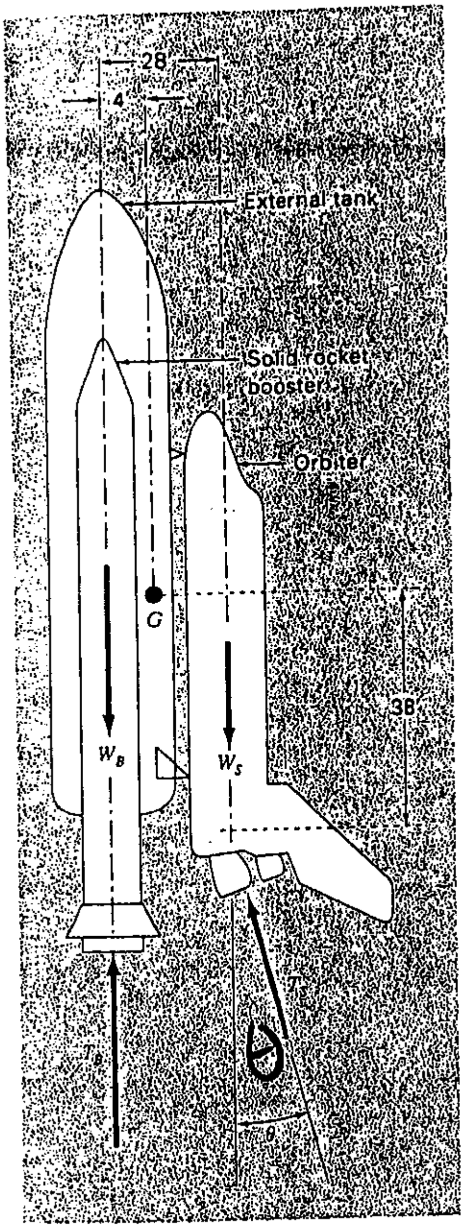


Figure P8.46

8.46 The space shuttle, at lift-off from the launch pad, has four forces acting on it, which are shown on the free-body diagram (Fig. P8.46). The combined weight of the two solid rocket boosters and external fuel tank is $W_B = 1.663 \times 10^6$ lb. The weight of the orbiter with a full payload is $W_S = 0.23 \times 10^6$ lb. The combined thrust of the two solid rocket boosters is $T_B = 5.30 \times 10^6$ lb. The combined thrust of the three liquid fuel orbiter engines is $T_S = 1.125 \times 10^6$ lb.

At liftoff, the orbiter engine thrust is directed at angle θ to make the resultant moment acting on the entire craft assembly (external tank, solid rocket boosters, and orbiter) equal to zero. With the resultant moment equal to zero, the craft will not rotate about its mass center G at liftoff. With these forces, the craft will have a resultant force with components in both the vertical and horizontal direction. The vertical resultant force component is what allows the craft to lift off from the launch pad and fly vertically. The horizontal resultant force component causes the craft to fly horizontally. The resultant moment acting on the craft will be zero when θ is adjusted to the proper value. If this angle is not adjusted properly, and there is some resultant moment acting on the craft, the craft will tend to rotate about it mass center.

(a) Resolve the orbiter thrust T_S into horizontal and vertical components, and then sum moments about point G, the craft mass center. Set the resulting moment equation equal to zero. This equation can now be solved for the value of θ required for liftoff.

(b) Derive an equation for the resultant moment acting on the craft in terms of the angle θ. Plot the resultant moment as a function of the angle θ over a range of -5 radians to $+5$ radians.

(c) Write a computer program to solve for the angle θ using Newton's method to find the root of the resultant moment equation. Make an initial first guess at the root of interest using the plot. Terminate your iterations when the value of θ has better than five significant figures.

(d) Repeat the program for the minimum payload weight of the orbiter of $W_S = 195{,}000$ lb.

NOTE!!! Moment function is
LGB = 4, LGS = 24, LTS = 38
WS = .230E6, WB = 1.663E6
TT = 5.3E6, TS = 1.125E6
f(x) = LGB*WB - LGB*TB - [illegible]
X = θ

NOTE!!! RAD. TO DEGREES
= RAD$\left(\frac{180}{\pi}\right)$

WRITE A MATLAB™ PROGRAM TO FIND THE ROOT OF THE MOMENT EQUATION USING THE NEWTON RAPHSON METHOD. TURN IN YOUR PROGRAM CODE AND 2 GRAPHS AND 2 PROGRAM RUNS PER THE CLASS DISCUSSION.

for Homework 3

$$X = \Theta$$

$$X_{NEXT} = X_{OLD} - \frac{f(X_{OLD})}{f'(X_{OLD})}$$

$$\begin{aligned} f(x) = {} & LGB*WB \\ & -LGB*TB \\ & -LGS*WS \\ & +LGS*TS*COS(x) \\ & -LTS*TS*SIN(x) \end{aligned}$$

$$\begin{aligned} f'(x) = {} & -LGS*TS*SIN(x) \\ & -LTS*TS*COS(x) \end{aligned}$$