One parameter code

```
clear;clc;close all;

prob_name='CAPS0';
delete(strcat('data\',prob_name,'\*.mat'));

Mm=1.8e-3;Mc=1.67e-3;k=62;
Om0=sqrt(k/Mm);
muG=(Mm+Mc)*9.81*0.2293;

p0N(1,1)={'om'}    ;p0(1,1)=(1/Om0)*2*pi*45;
p0N(2,1)={'al'}    ;p0(2,1)=(1/muG)*0.208;
p0N(3,1)={'ga'}    ;p0(3,1)=Mm/Mc;
p0N(4,1)={'k1'}    ;p0(4,1)=(1/k)*27900;
p0N(5,1)={'k2'}    ;p0(5,1)=(1/k)*53500;
p0N(6,1)={'xi'}    ;p0(6,1)=(1/(2*Mm*Om0))*0.0156;
p0N(7,1)={'void'}  ;p0(7,1)=0;
p0N(8,1)={'g1'}    ;p0(8,1)=(k/muG)*1.6e-3;
p0N(9,1)={'g2'}    ;p0(9,1)=(k/muG)*0;
p0N(10,1)={'gb'}   ;p0(10,1)=(k/muG)*7.6e-3; %(k/muG)*8.31e-3
p0N(11,1)={'lb'}   ;p0(11,1)=(k/muG)*1.033e-3;
p0N(12,1)={'B0'}   ;p0(12,1)=(1/k)*(-9.111e-2);
p0N(13,1)={'B1'}   ;p0(13,1)=(muG/k^2)*(3.918e4);
```

- declare the name of the continuation problem "CAPS0" in this case.
- define the parameters

```
p0N(14,1)={'B2'}  ;p0(14,1)=(muG^2/k^3)*(-1.925e7);

modes={ 'Ck2-Vcp-NCb' 'Ck2-Vcn-NCb' 'NCk-Vcn-NCb' 'NCk-Vcp-NCb' 'Ck2-Vcp-NCb' 'Ck2-Vcn-NCb' 'NCk-Vcn-NCb' 'Ck1-Vcn-NCb' 'Ck1-V
events={'Vc0'        'Ck2'         'Vc0'         'Ck2'         'Vc0'         'Ck2'         'Ck1'         'Vc0'         'Ck1'
resets=modes;
resets{end}='pos-res';

MAT=[
0   -4.38926841064442   0   -23.0002797359077   -0.644631438609608  -0.764493419256697  0
0.0968984263681015  11.9235743721816   -0.472185456647041  11.9235743721816   -0.701336891717458  -0.712829892410753  0.0503
0.473768791407814   16.9454810166075   1.56125112837913e-16   22.8375972705266   -0.727602493761267  -0.685998912000398  0.
4.84539026661554   -7.19443020564855   7.00236726745506   -7.19443020564855   -0.904502631583402  0.426467896788251   0.8604
-1.68871686629037   -17.5991775150701   -2.38364883387021e-13   -18.8274532406629   -0.372405479525962  0.928070072138030   1.
-1.89324419879899   -16.5862859812738   -0.214135143072123  -16.5862859812738   -0.355607454310320  0.934635351668450   1.3636
-2.33899005726126   0.196284418891519   1.15716897869378e-15   18.7779239396586   -0.264592757666764  0.964360195056771   1.
```

- define modes involved, events and reset functions
- efine initial points for every mode
- very starting point and ending point should be defined
- simulate each mode as a first guess of the solution

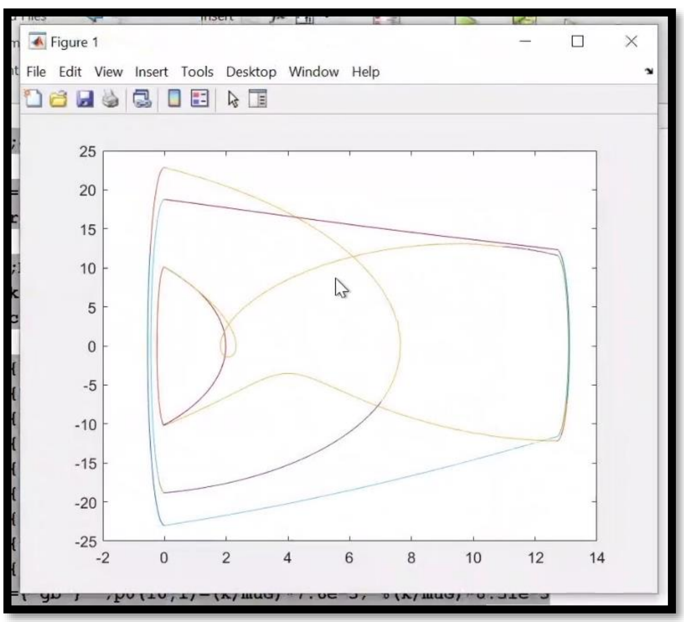
```
-1.59778560923237    2.81001656133719    12.7089343197156    -11.6145548462513    0.410496313818697    -0.91
-1.47821466643057    -4.38926554961059    -2.15449098177941e-15    -23.0002793960055    -0.644631254317500    -
];

[t1,x1]=ode45(@(t,x) vec_fields(x,p0,modes{1}),[0 MAT(2,end)-MAT(1,end)],MAT(1,1:end-1)');
[t2,x2]=ode45(@(t,x) vec_fields(x,p0,modes{2}),[0 MAT(3,end)-MAT(2,end)],MAT(2,1:end-1)');
[t3,x3]=ode45(@(t,x) vec_fields(x,p0,modes{3}),[0 MAT(4,end)-MAT(3,end)],MAT(3,1:end-1)');
[t4,x4]=ode45(@(t,x) vec_fields(x,p0,modes{4}),[0 MAT(5,end)-MAT(4,end)],MAT(4,1:end-1)');
[t5,x5]=ode45(@(t,x) vec_fields(x,p0,modes{5}),[0 MAT(6,end)-MAT(5,end)],MAT(5,1:end-1)');
[t6,x6]=ode45(@(t,x) vec_fields(x,p0,modes{6}),[0 MAT(7,end)-MAT(6,end)],MAT(6,1:end-1)');
[t7,x7]=ode45(@(t,x) vec_fields(x,p0,modes{7}),[0 MAT(8,end)-MAT(7,end)],MAT(7,1:end-1)');
[t8,x8]=ode45(@(t,x) vec_fields(x,p0,modes{8}),[0 MAT(9,end)-MAT(8,end)],MAT(8,1:end-1)');
[t9,x9]=ode45(@(t,x) vec_fields(x,p0,modes{9}),[0 MAT(10,end)-MAT(9,end)],MAT(9,1:end-1)');
[t10,x10]=ode45(@(t,x) vec_fields(x,p0,modes{10}),[0 MAT(11,end)-MAT(10,end)],MAT(10,1:end-1)');
[t11,x11]=ode45(@(t,x) vec_fields(x,p0,modes{11}),[0 MAT(12,end)-MAT(11,end)],MAT(11,1:end-1)');
[t12,x12]=ode45(@(t,x) vec_fields(x,p0,modes{12}),[0 MAT(13,end)-MAT(12,end)],MAT(12,1:end-1)');
[t13,x13]=ode45(@(t,x) vec_fields(x,p0,modes{13}),[0 MAT(14,end)-MAT(13,end)],MAT(13,1:end-1)');
[t14,x14]=ode45(@(t,x) vec_fields(x,p0,modes{14}),[0 MAT(15,end)-MAT(14,end)],MAT(14,1:end-1)');
[t15,x15]=ode45(@(t,x) vec_fields(x,p0,modes{15}),[0 MAT(16,end)-MAT(15,end)],MAT(15,1:end-1)');
[t16,x16]=ode45(@(t,x) vec_fields(x,p0,modes{16}),[0 MAT(17,end)-MAT(16,end)],MAT(16,1:end-1)');
```

```
[t17,x17]=ode45(@(t,x) vec_fields(x,p0,modes{17}),[0 MAT(18,end)-MAT(17,end)],MAT(17,1:end-1)');
[t18,x18]=ode45(@(t,x) vec_fields(x,p0,modes{18}),[0 MAT(19,end)-MAT(18,end)],MAT(18,1:end-1)');
[t19,x19]=ode45(@(t,x) vec_fields(x,p0,modes{19}),[0 MAT(20,end)-MAT(19,end)],MAT(19,1:end-1)');
[t20,x20]=ode45(@(t,x) vec_fields(x,p0,modes{20}),[0 2*2*pi/p0(1,1)-MAT(20,end)],MAT(20,1:end-1)');

t0={t1 t2 t3 t4 t5 t6 t7 t8 t9 t10 t11 t12 t13 t14 t15 t16 t17 t18 t19 t20};
x0={x1 x2 x3 x4 x5 x6 x7 x8 x9 x10 x11 x12 x13 x14 x15 x16 x17 x18 x19 x20};

%plot(x1(:,3),x1(:,4),x2(:,3),x2(:,4),x3(:,3),x3(:,4),x4(:,3),x4(:,4),x5(:,3),x5(:,4),x6(:,3),x6(:,4
```



- this is the initial guess for the solution, every color refers to each mode.

```
hspo_args={{@vec_fields,@event_funcs,@reset_funcs},{@Df_Dsvar,@Dh_Dsvar,@Dg_Dsvar},{@Df_Dpar,@Dh_Dpar,@Dg_Dpar},modes,events,

prob=coco_prob();
prob=coco_set(prob,'cont','ItMX',5000,'h_min',0.01,'h_max',3,'h0',0.1,'NPR',20);
prob=coco_set(prob,'coll','NTST',20,'NCOL',4);

prob=ode_isol2hspo(prob,'',hspo_args{:});

fprintf('\n Run=''%s'': Continue family of multi-segment periodic orbits.\n',prob_name);
```

- enter vector fields, event functions, reset functions and derivatives (derivatives improve the accuracy of the program)
- define parameters, (5000 is the number of points on the curve, 0.01 is the step size), 0.01 and 3 are the minimum and maximum step size (range), the initial step size is 0.1.

```
[data1,uidx1]=coco_get_func_data(prob,'hspo.orb.bvp.seg1.coll','data','uidx');
maps1=data1.coll_seg.maps;
prob=coco_add_pars(prob,'pars1',uidx1(maps1.T_idx),'T1','active');

[data2,uidx2]=coco_get_func_data(prob,'hspo.orb.bvp.seg2.coll','data','uidx');
maps2=data2.coll_seg.maps;
prob=coco_add_pars(prob,'pars2',uidx2(maps2.T_idx),'T2','active');

[data3,uidx3]=coco_get_func_data(prob,'hspo.orb.bvp.seg3.coll','data','uidx');
maps3=data3.coll_seg.maps;
prob=coco_add_pars(prob,'pars3',uidx3(maps3.T_idx),'T3','active');

[data4,uidx4]=coco_get_func_data(prob,'hspo.orb.bvp.seg4.coll','data','uidx');
maps4=data4.coll_seg.maps;
prob=coco_add_pars(prob,'pars4',uidx4(maps4.T_idx),'T4','active');
```

- define the data structures for each mode, in this example 20 modes

```
[data19,uidx19]=coco_get_func_data(prob,'hspo.orb.bvp.seg19.coll','data','uidx');
maps19=data19.coll_seg.maps;
prob=coco_add_pars(prob,'pars19',uidx19(maps19.T_idx),'T19','active');

[data20,uidx20]=coco_get_func_data(prob,'hspo.orb.bvp.seg20.coll','data','uidx');
maps20=data20.coll_seg.maps;
prob=coco_add_pars(prob,'pars20',uidx20(maps20.T_idx),'T20','active');

[data,uidx]=coco_get_func_data(prob,'hspo.orb.bvp','data','uidx');
maps=data.bvp_bc;
```

```
prob=coco_add_func(prob,'P1',@M_Vavg,{data data1 data2 data3 data4 data5 data6 data7 data8 data9 data10 data11 data12 data13 d
prob=coco_add_func(prob,'P2',@M_Pavg,{data data1 data2 data3 data4 data5 data6 data7 data8 data9 data10 data11 data12 data13 d
prob=coco_add_func(prob,'P3',@M_Fcap,{data data1 data2 data3 data4 data5 data6 data7 data8 data9 data10 data11 data12 data13 d
prob=coco_add_event(prob,'UZ','M_Vavg',0);
prob=coco_add_event(prob,'UZ','al',(1/muG)*[0.2084 0.2095 0.2098]);

coco(prob,prob_name,[],1,{'al' 'hspo.test.USTAB' 'M_Vavg' 'M_Fcap' 'M_Pavg' 'T1' 'T2' 'T3' 'T4' 'T5' 'T6' 'T7' 'T8' 'T9' 'T10'
```

- define the measures that you designed for this program, for example "M_Vavg" refers to the average velocity of the device, "M_Pavg" is the power, and "M_Fcap" is the force. Define what you want to measure.
- the 1st "UZ" point showed when the average became zero, and the 2nd shows the value for a different variable.
- the "coco" line is the final definition of the problem.

```
Run='CAPS0': Continue family of multi-segment periodic orbits.

   STEP   DAMPING               NORMS              COMPUTATION TIMES
  IT SIT    GAMMA     ||d||     ||f||     ||U||   F(x)  DF(x)  SOLVE
   0                          3.93e-01  4.43e+03   0.0    0.0    0.0
   1   1  1.00e+00  1.52e-01  9.16e-07  4.43e+03   0.0    0.1    0.1
   2   1  1.00e+00  3.85e-03  2.51e-10  4.43e+03   0.0    0.1    0.1
   3   1  1.00e+00  3.63e-08  2.77e-13  4.43e+03   0.1    0.1    0.2

STEP      TIME        ||U||  LABEL  TYPE         al hspo.test.USTAB      M_Vavg      M_Fcap
   0  00:00:02   4.4312e+03      1  EP   2.6648e+01      0.0000e+00 -1.7920e-01  4.7612e+02
```

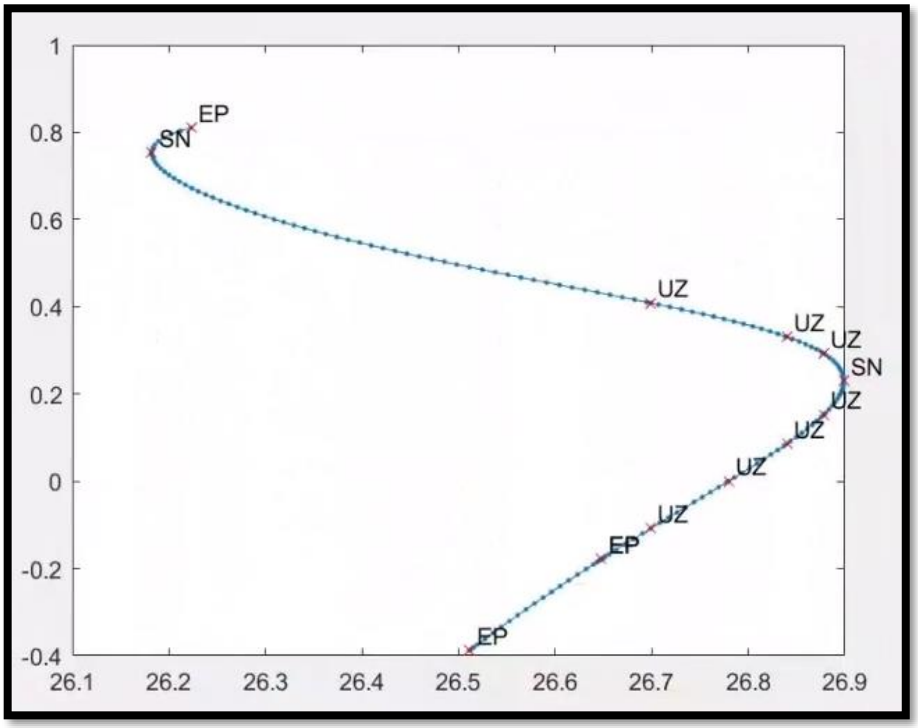
- every point corresponds to one periodic solution of the model along a family of solutions generated by by the alpha (al).

```
STEP      TIME        ||U||  LABEL  TYPE         al hspo.test.USTAB      M_Vavg      M_Fcap
   0  00:00:02   4.4312e+03      1  EP   2.6648e+01      0.0000e+00 -1.7920e-01  4.7612e+02
  20  00:00:18   4.4329e+03      2       2.6515e+01      0.0000e+00 -3.8129e-01  4.8127e+02
  24  00:00:24   4.4330e+03      3  EP   2.6511e+01      0.0000e+00 -3.8797e-01  4.8141e+02

STEP      TIME        ||U||  LABEL  TYPE         al hspo.test.USTAB      M_Vavg      M_Fcap
   0  00:00:25   4.4312e+03      4  EP   2.6648e+01      0.0000e+00 -1.7920e-01  4.7612e+02
  10  00:00:32   4.4306e+03      5  UZ   2.6699e+01      0.0000e+00 -1.0841e-01  4.7363e+02
  20  00:00:43   4.4296e+03      6  UZ   2.6780e+01      0.0000e+00  1.4124e-09  4.6890e+02
  20  00:00:43   4.4295e+03      7       2.6786e+01      0.0000e+00  8.5430e-03  4.6845e+02
  28  00:00:50   4.4287e+03      8  UZ   2.6840e+01      0.0000e+00  8.5598e-02  4.6382e+02
  35  00:00:57   4.4279e+03      9  UZ   2.6878e+01      0.0000e+00  1.5250e-01  4.5857e+02
  40  00:01:01   4.4273e+03     10       2.6895e+01      0.0000e+00  1.9736e-01  4.5414e+02
  45  00:01:10   4.4268e+03     11  FP   2.6899e+01      0.0000e+00  2.2992e-01  4.5043e+02
  45  00:01:10   4.4268e+03     12  SN   2.6899e+01      1.0000e+00  2.2992e-01  4.5043e+02
  54  00:01:19   4.4257e+03     13  UZ   2.6878e+01      1.0000e+00  2.9262e-01  4.4159e+02
  60  00:01:25   4.4249e+03     14  UZ   2.6840e+01      1.0000e+00  3.3116e-01  4.3491e+02
  60  00:01:25   4.4249e+03     15       2.6839e+01      1.0000e+00  3.3216e-01  4.3472e+02
```

74	00:01:37	4.4233e+03	16	UZ	2.6699e+01	1.0000e+00	4.0881e-01	4.1953e+02
80	00:01:42	4.4227e+03	17		2.6618e+01	1.0000e+00	4.4433e-01	4.1241e+02
100	00:01:58	4.4212e+03	18		2.6363e+01	1.0000e+00	5.6673e-01	3.9105e+02
120	00:02:15	4.4208e+03	19		2.6196e+01	1.0000e+00	7.0954e-01	3.7205e+02
126	00:02:24	4.4208e+03	20	FP	2.6182e+01	1.0000e+00	7.5320e-01	3.6706e+02
126	00:02:25	4.4208e+03	21	SN	2.6182e+01	1.0000e+00	7.5320e-01	3.6706e+02

- the “EP” refer to the starting and end points, “SN” is saddle node bifurcation.



one-parameter, to detect bifurcations velocity against alpha

<u>Two parameter code</u>

- again, name the continuation process, “CAPS2”
- prob = ode_hspo2SN(prob, ‘ ‘ , ‘CAPS0’, 10);
- this line gets point 10 from the 1st run (one parameter), which is the saddle node. If it is period doubling the name would be “ode_hspo2PD”

```
clear;clc;close all;

prob_name='CAPS2';
delete(strcat('data\',prob_name,'\*.mat'));

Mm=1.8e-3;Mc=1.67e-3;k=62;
Om0=sqrt(k/Mm);
muG=(Mm+Mc)*9.81*0.2293;

prob=coco_prob();
prob=coco_set(prob,'cont','ItMX',5000,'h_min',0.01,'h_max',100,'h0',0.1,'NPR',10);
prob=coco_set(prob,'coll','NTST',20,'NCOL',4);

prob=ode_hspo2SN(prob,'','CAPS0',10);

fprintf('\n Run=''%s'': Continue family of multi-segment periodic orbits.\n',prob_name);

[data1,uidx1]=coco_get_func_data(prob,'hspo.orb.bvp.seg1.coll','data','uidx');
maps1=data1.coll_seg.maps;
prob=coco_add_pars(prob,'pars1',uidx1(maps1.T_idx),'T1','active');

[data2,uidx2]=coco_get_func_data(prob,'hspo.orb.bvp.seg2.coll','data','uidx');
```

- define the data structures as before.
- importantly, this time define the two parameters to be varied, in this case “al” and “om”.

```
[data20,uidx20]=coco_get_func_data(prob,'hspo.orb.bvp.seg20.coll','data','uidx');
maps20=data20.coll_seg.maps;
prob=coco_add_pars(prob,'pars20',uidx20(maps20.T_idx),'T20','active');

[data,uidx]=coco_get_func_data(prob,'hspo.orb.bvp','data','uidx');
maps=data.bvp_bc;

prob=coco_add_func(prob,'P1',@M_Vavg,{data data1 data2 data3 data4 data5 data6 data7 da
prob=coco_add_func(prob,'P2',@M_Pavg,{data data1 data2 data3 data4 data5 data6 data7 da
prob=coco_add_func(prob,'P3',@M_Fcap,{data data1 data2 data3 data4 data5 data6 data7 da

coco(prob,prob_name,[],1,{'al' 'om' 'M_Vavg' 'M_Fcap' 'M_Pavg' 'T1' 'T2' 'T3' 'T4' 'T5'


%Plot Bifurcation Diagram

bd=coco_bd_read(prob_name);
vert=coco_bd_col(bd,'om');
hor=coco_bd_col(bd,'al');
plot(hor,vert,'.-')
```

These are all the functions he included for this problem.

```
Df_Dpar.m
Df_Dsvar.m
Dg_Dpar.m
Dg_Dsvar.m
Dh_Dpar.m
Dh_Dsvar.m
event_funcs.m
M_Fcap.m
M_Pavg.m
M_Vavg.m
reset_funcs.m
vec_fields.m
makeal0.m
makeCB.m
makeSN.m
sol_bif_plot.m
sol_tester.m
```

```
%Plot Bifurcation Diagram

bd=coco_bd_read(prob_name);
vert=coco_bd_col(bd,'M_Vavg');
hor=coco_bd_col(bd,'al');
plot(hor,vert,'.-')

hold on
idx=coco_bd_idxs(bd,'BP');
```

Answers

- hspo – hybrid system periodic orbit, msbvp - multi segment boundary value problem, msbvp is part of hspo.
- "atlas" is related to many folds along a curve (this is not so relevant for this type of problem, unless if you are inserting certain algebraic equations).

```
prob=coco_prob();
prob=coco_set(prob,'cont','ItMX',5000,'h_min',0.01,'h_max',3,'h0',0.1,'NPR',20);
prob=coco_set(prob,'coll','NTST',20,'NCOL',4);
```

- Coco_set is used to set parameters, in this case cont problem type is continuation, "cont", ItMX is the maximum number of points (5000), h_min and h_max define the minimum and maximum step size, "h0" 0.1 is the initial step size.
- "NPR" 20 shows a set of results after every 20 steps, shown on the results table as 0 20 40 … (new labels are shown if defined by user, for example "UZ'' at al = xxx).
- (Discretization of the periodic solution). Second line sets number of collocation points to 4 and discretization points 20.
- <u>hspo_iso2segs() and msbvp_iso2segs.</u>
  1st one is for setting up an initial solution, 2nd one is more general (multi boundary), we usually use hspo_iso2segs() .