**Source code 5 : persistent dorsal mesocardium**

function m = gpt\_source\_code\_5\_persistent\_dm\_20170512( m )

%m = gpt\_source\_code\_4\_persistent\_dm\_20170512( m )

% Morphogen interaction function.

% Written at 2017-05-12 10:37:18.

% GFtbox revision 5411, 2016-02-26 12:00.

% The user may edit any part of this function between delimiters

% of the form "USER CODE..." and "END OF USER CODE...". The

% delimiters themselves must not be moved, edited, deleted, or added.

if isempty(m), return; end

fprintf( 1, '%s found in %s\n', mfilename(), which(mfilename()) );

try

m = local\_setproperties( m );

catch

end

setGlobals();

realtime = m.globalDynamicProps.currenttime;

dt = m.globalProps.timestep;

%%% USER CODE: INITIALISATION

% In this section you may modify the mesh in any way whatsoever.

%%% END OF USER CODE: INITIALISATION

%%% SECTION 1: ACCESSING MORPHOGENS AND TIME.

%%% AUTOMATICALLY GENERATED CODE: DO NOT EDIT.

polariser\_i = FindMorphogenRole( m, 'POLARISER' );

P = m.morphogens(:,polariser\_i);

[kapar\_i,kapar\_p,kapar\_a,kapar\_l] = getMgenLevels( m, 'KAPAR' );

[kaper\_i,kaper\_p,kaper\_a,kaper\_l] = getMgenLevels( m, 'KAPER' );

[kbpar\_i,kbpar\_p,kbpar\_a,kbpar\_l] = getMgenLevels( m, 'KBPAR' );

[kbper\_i,kbper\_p,kbper\_a,kbper\_l] = getMgenLevels( m, 'KBPER' );

[knor\_i,knor\_p,knor\_a,knor\_l] = getMgenLevels( m, 'KNOR' );

[strainret\_i,strainret\_p,strainret\_a,strainret\_l] = getMgenLevels( m, 'STRAINRET' );

[arrest\_i,arrest\_p,arrest\_a,arrest\_l] = getMgenLevels( m, 'ARREST' );

[id\_venous\_pole\_i,id\_venous\_pole\_p,id\_venous\_pole\_a,id\_venous\_pole\_l] = getMgenLevels( m, 'ID\_VENOUS\_POLE' );

[id\_arterial\_pole\_i,id\_arterial\_pole\_p,id\_arterial\_pole\_a,id\_arterial\_pole\_l] = getMgenLevels( m, 'ID\_ARTERIAL\_POLE' );

[id\_right\_ventricle\_i,id\_right\_ventricle\_p,id\_right\_ventricle\_a,id\_right\_ventricle\_l] = getMgenLevels( m, 'ID\_RIGHT\_VENTRICLE' );

[id\_left\_ventricle\_i,id\_left\_ventricle\_p,id\_left\_ventricle\_a,id\_left\_ventricle\_l] = getMgenLevels( m, 'ID\_LEFT\_VENTRICLE' );

[id\_oft\_i,id\_oft\_p,id\_oft\_a,id\_oft\_l] = getMgenLevels( m, 'ID\_OFT' );

[id\_atria\_i,id\_atria\_p,id\_atria\_a,id\_atria\_l] = getMgenLevels( m, 'ID\_ATRIA' );

[id\_art\_band\_i,id\_art\_band\_p,id\_art\_band\_a,id\_art\_band\_l] = getMgenLevels( m, 'ID\_ART\_BAND' );

[id\_vein\_band\_i,id\_vein\_band\_p,id\_vein\_band\_a,id\_vein\_band\_l] = getMgenLevels( m, 'ID\_VEIN\_BAND' );

[id\_iv\_band\_i,id\_iv\_band\_p,id\_iv\_band\_a,id\_iv\_band\_l] = getMgenLevels( m, 'ID\_IV\_BAND' );

[id\_dorsal\_line\_i,id\_dorsal\_line\_p,id\_dorsal\_line\_a,id\_dorsal\_line\_l] = getMgenLevels( m, 'ID\_DORSAL\_LINE' );

[id\_ventral\_line\_i,id\_ventral\_line\_p,id\_ventral\_line\_a,id\_ventral\_line\_l] = getMgenLevels( m, 'ID\_VENTRAL\_LINE' );

[id\_right\_i,id\_right\_p,id\_right\_a,id\_right\_l] = getMgenLevels( m, 'ID\_RIGHT' );

[id\_left\_i,id\_left\_p,id\_left\_a,id\_left\_l] = getMgenLevels( m, 'ID\_LEFT' );

[id\_inner\_curvature\_i,id\_inner\_curvature\_p,id\_inner\_curvature\_a,id\_inner\_curvature\_l] = getMgenLevels( m, 'ID\_INNER\_CURVATURE' );

[id\_outer\_curvature\_i,id\_outer\_curvature\_p,id\_outer\_curvature\_a,id\_outer\_curvature\_l] = getMgenLevels( m, 'ID\_OUTER\_CURVATURE' );

[id\_sdorsal\_line\_i,id\_sdorsal\_line\_p,id\_sdorsal\_line\_a,id\_sdorsal\_line\_l] = getMgenLevels( m, 'ID\_SDORSAL\_LINE' );

[id\_sleft\_i,id\_sleft\_p,id\_sleft\_a,id\_sleft\_l] = getMgenLevels( m, 'ID\_SLEFT' );

[id\_siv\_i,id\_siv\_p,id\_siv\_a,id\_siv\_l] = getMgenLevels( m, 'ID\_SIV' );

[s\_iv\_i,s\_iv\_p,s\_iv\_a,s\_iv\_l] = getMgenLevels( m, 'S\_IV' );

[s\_dorsal\_line\_i,s\_dorsal\_line\_p,s\_dorsal\_line\_a,s\_dorsal\_line\_l] = getMgenLevels( m, 'S\_DORSAL\_LINE' );

[s\_left\_i,s\_left\_p,s\_left\_a,s\_left\_l] = getMgenLevels( m, 'S\_LEFT' );

[s\_art\_band\_i,s\_art\_band\_p,s\_art\_band\_a,s\_art\_band\_l] = getMgenLevels( m, 'S\_ART\_BAND' );

[id\_sart\_band\_i,id\_sart\_band\_p,id\_sart\_band\_a,id\_sart\_band\_l] = getMgenLevels( m, 'ID\_SART\_BAND' );

[s\_vein\_band\_i,s\_vein\_band\_p,s\_vein\_band\_a,s\_vein\_band\_l] = getMgenLevels( m, 'S\_VEIN\_BAND' );

[id\_svein\_band\_i,id\_svein\_band\_p,id\_svein\_band\_a,id\_svein\_band\_l] = getMgenLevels( m, 'ID\_SVEIN\_BAND' );

[s\_ventral\_line\_i,s\_ventral\_line\_p,s\_ventral\_line\_a,s\_ventral\_line\_l] = getMgenLevels( m, 'S\_VENTRAL\_LINE' );

[id\_sventral\_line\_i,id\_sventral\_line\_p,id\_sventral\_line\_a,id\_sventral\_line\_l] = getMgenLevels( m, 'ID\_SVENTRAL\_LINE' );

[s\_right\_i,s\_right\_p,s\_right\_a,s\_right\_l] = getMgenLevels( m, 'S\_RIGHT' );

[id\_sright\_i,id\_sright\_p,id\_sright\_a,id\_sright\_l] = getMgenLevels( m, 'ID\_SRIGHT' );

[id\_ventral\_left\_i,id\_ventral\_left\_p,id\_ventral\_left\_a,id\_ventral\_left\_l] = getMgenLevels( m, 'ID\_VENTRAL\_LEFT' );

[s\_outer\_curvature\_i,s\_outer\_curvature\_p,s\_outer\_curvature\_a,s\_outer\_curvature\_l] = getMgenLevels( m, 'S\_OUTER\_CURVATURE' );

[id\_souter\_curvature\_i,id\_souter\_curvature\_p,id\_souter\_curvature\_a,id\_souter\_curvature\_l] = getMgenLevels( m, 'ID\_SOUTER\_CURVATURE' );

[s\_ventral\_left\_i,s\_ventral\_left\_p,s\_ventral\_left\_a,s\_ventral\_left\_l] = getMgenLevels( m, 'S\_VENTRAL\_LEFT' );

[id\_sventral\_left\_i,id\_sventral\_left\_p,id\_sventral\_left\_a,id\_sventral\_left\_l] = getMgenLevels( m, 'ID\_SVENTRAL\_LEFT' );

[s\_inner\_curvature\_i,s\_inner\_curvature\_p,s\_inner\_curvature\_a,s\_inner\_curvature\_l] = getMgenLevels( m, 'S\_INNER\_CURVATURE' );

[id\_sinner\_curvature\_i,id\_sinner\_curvature\_p,id\_sinner\_curvature\_a,id\_sinner\_curvature\_l] = getMgenLevels( m, 'ID\_SINNER\_CURVATURE' );

[id\_dorsal\_right\_i,id\_dorsal\_right\_p,id\_dorsal\_right\_a,id\_dorsal\_right\_l] = getMgenLevels( m, 'ID\_DORSAL\_RIGHT' );

[s\_dorsal\_right\_i,s\_dorsal\_right\_p,s\_dorsal\_right\_a,s\_dorsal\_right\_l] = getMgenLevels( m, 'S\_DORSAL\_RIGHT' );

[id\_sdorsal\_right\_i,id\_sdorsal\_right\_p,id\_sdorsal\_right\_a,id\_sdorsal\_right\_l] = getMgenLevels( m, 'ID\_SDORSAL\_RIGHT' );

[s\_oft\_i,s\_oft\_p,s\_oft\_a,s\_oft\_l] = getMgenLevels( m, 'S\_OFT' );

[id\_soft\_i,id\_soft\_p,id\_soft\_a,id\_soft\_l] = getMgenLevels( m, 'ID\_SOFT' );

[s\_atria\_i,s\_atria\_p,s\_atria\_a,s\_atria\_l] = getMgenLevels( m, 'S\_ATRIA' );

[id\_satria\_i,id\_satria\_p,id\_satria\_a,id\_satria\_l] = getMgenLevels( m, 'ID\_SATRIA' );

[id\_clone\_density\_i,id\_clone\_density\_p,id\_clone\_density\_a,id\_clone\_density\_l] = getMgenLevels( m, 'ID\_CLONE\_DENSITY' );

[s\_arterial\_pole\_i,s\_arterial\_pole\_p,s\_arterial\_pole\_a,s\_arterial\_pole\_l] = getMgenLevels( m, 'S\_ARTERIAL\_POLE' );

[s\_venous\_pole\_i,s\_venous\_pole\_p,s\_venous\_pole\_a,s\_venous\_pole\_l] = getMgenLevels( m, 'S\_VENOUS\_POLE' );

[id\_left\_side\_i,id\_left\_side\_p,id\_left\_side\_a,id\_left\_side\_l] = getMgenLevels( m, 'ID\_LEFT\_SIDE' );

[id\_right\_side\_i,id\_right\_side\_p,id\_right\_side\_a,id\_right\_side\_l] = getMgenLevels( m, 'ID\_RIGHT\_SIDE' );

[id\_sarterial\_pole\_i,id\_sarterial\_pole\_p,id\_sarterial\_pole\_a,id\_sarterial\_pole\_l] = getMgenLevels( m, 'ID\_SARTERIAL\_POLE' );

[s\_right\_ventricle\_i,s\_right\_ventricle\_p,s\_right\_ventricle\_a,s\_right\_ventricle\_l] = getMgenLevels( m, 'S\_RIGHT\_VENTRICLE' );

[id\_sright\_ventricle\_i,id\_sright\_ventricle\_p,id\_sright\_ventricle\_a,id\_sright\_ventricle\_l] = getMgenLevels( m, 'ID\_SRIGHT\_VENTRICLE' );

[id\_oft\_band\_i,id\_oft\_band\_p,id\_oft\_band\_a,id\_oft\_band\_l] = getMgenLevels( m, 'ID\_OFT\_BAND' );

[id\_soft\_band\_i,id\_soft\_band\_p,id\_soft\_band\_a,id\_soft\_band\_l] = getMgenLevels( m, 'ID\_SOFT\_BAND' );

[s\_oft\_band\_i,s\_oft\_band\_p,s\_oft\_band\_a,s\_oft\_band\_l] = getMgenLevels( m, 'S\_OFT\_BAND' );

% Mesh type: cylinder

% basecap: 0

% baseheight: 1

% baserings: 0

% circumdivs: 30

% height: 1.5

% heightdivs: 30

% radius: 0.1

% randomness: 0

% topcap: 0

% topheight: 1

% toprings: 0

% version: 1

% Morphogen Diffusion Decay Dilution Mutant

% --------------------------------------------------

% KAPAR ---- ---- ---- ----

% KAPER ---- ---- ---- ----

% KBPAR ---- ---- ---- ----

% KBPER ---- ---- ---- ----

% KNOR ---- ---- ---- ----

% POLARISER 0.05 ---- ---- ----

% STRAINRET ---- ---- ---- ----

% ARREST ---- ---- ---- ----

% ID\_VENOUS\_POLE ---- ---- ---- ----

% ID\_ARTERIAL\_POLE ---- ---- ---- ----

% ID\_RIGHT\_VENTRICLE ---- ---- ---- ----

% ID\_LEFT\_VENTRICLE ---- ---- ---- ----

% ID\_OFT ---- ---- ---- ----

% ID\_ATRIA ---- ---- ---- ----

% ID\_ART\_BAND ---- ---- ---- ----

% ID\_VEIN\_BAND ---- ---- ---- ----

% ID\_IV\_BAND ---- ---- ---- ----

% ID\_DORSAL\_LINE ---- ---- ---- ----

% ID\_VENTRAL\_LINE ---- ---- ---- ----

% ID\_RIGHT ---- ---- ---- ----

% ID\_LEFT ---- ---- ---- ----

% ID\_INNER\_CURVATURE ---- ---- ---- ----

% ID\_OUTER\_CURVATURE ---- ---- ---- ----

% ID\_SDORSAL\_LINE ---- ---- ---- ----

% ID\_SLEFT ---- ---- ---- ----

% ID\_SIV ---- ---- ---- ----

% S\_IV 0.0001 0.1 ---- ----

% S\_DORSAL\_LINE 0.0001 0.1 ---- ----

% S\_LEFT 0.0001 0.1 ---- ----

% S\_ART\_BAND 0.0001 0.1 ---- ----

% ID\_SART\_BAND ---- ---- ---- ----

% S\_VEIN\_BAND 0.0001 0.1 ---- ----

% ID\_SVEIN\_BAND ---- ---- ---- ----

% S\_VENTRAL\_LINE 0.0001 0.1 ---- ----

% ID\_SVENTRAL\_LINE ---- ---- ---- ----

% S\_RIGHT 0.0001 0.1 ---- ----

% ID\_SRIGHT ---- ---- ---- ----

% ID\_VENTRAL\_LEFT ---- ---- ---- ----

% S\_OUTER\_CURVATURE 0.0001 0.1 ---- ----

% ID\_SOUTER\_CURVATURE ---- ---- ---- ----

% S\_VENTRAL\_LEFT 0.0001 0.1 ---- ----

% ID\_SVENTRAL\_LEFT ---- ---- ---- ----

% S\_INNER\_CURVATURE 0.0001 0.1 ---- ----

% ID\_SINNER\_CURVATURE ---- ---- ---- ----

% ID\_DORSAL\_RIGHT ---- ---- ---- ----

% S\_DORSAL\_RIGHT 0.0001 0.1 ---- ----

% ID\_SDORSAL\_RIGHT ---- ---- ---- ----

% S\_OFT 0.0001 0.1 ---- ----

% ID\_SOFT ---- ---- ---- ----

% S\_ATRIA 0.0001 0.1 ---- ----

% ID\_SATRIA ---- ---- ---- ----

% ID\_CLONE\_DENSITY ---- ---- ---- ----

% S\_ARTERIAL\_POLE 0.01 0.05 ---- ----

% S\_VENOUS\_POLE 0.0001 0.1 ---- ----

% ID\_LEFT\_SIDE ---- ---- ---- ----

% ID\_RIGHT\_SIDE ---- ---- ---- ----

% ID\_SARTERIAL\_POLE ---- ---- ---- ----

% S\_RIGHT\_VENTRICLE 0.0001 0.1 ---- ----

% ID\_SRIGHT\_VENTRICLE ---- ---- ---- ----

% ID\_OFT\_BAND ---- ---- ---- ----

% ID\_SOFT\_BAND ---- ---- ---- ----

% S\_OFT\_BAND ---- 0.1 ---- ----

%%% USER CODE: MORPHOGEN INTERACTIONS

% In this section you may modify the mesh in any way that does not

% alter the set of nodes.

m=leaf\_fix\_vertex(m,'vertex',[],'dfs','');

m=leaf\_fix\_vertex(m,'vertex',38:30:878,'dfs','xyz');% all vertices of the DM are fixed in (was yz)

m=leaf\_fix\_vertex(m,'vertex',39:30:879,'dfs','xyz');% all vertices of the DM are fixed in (was yz)

m=leaf\_fix\_vertex(m,'vertex',908:909,'dfs','xyz');% attachment of arterial pole (allowing rotation)

m=leaf\_fix\_vertex(m,'vertex',1:30,'dfs','xyz');% all vertices of the venous pole are fixed in xyz

m=leaf\_fix\_vertex(m,'vertex',901:907,'dfs','z');% all vertices of the arterial pole are fixed in z

m=leaf\_fix\_vertex(m,'vertex',910:930,'dfs','z');% all vertices of the arterial pole are fixed in z

% m=leaf\_fix\_vertex(m,'vertex',99:30:189,'dfs','xyz');% vertices on the left side of the venous pole are fixed

% m=leaf\_fix\_vertex(m,'vertex',100:30:190,'dfs','xyz');

%

% m=leaf\_fix\_vertex(m,'vertex',816:30:906,'dfs','xyz');% vertices on the left of the arterial pole are fixed (mesocard attachment)

% m=leaf\_fix\_vertex(m,'vertex',817:30:907,'dfs','xyz');

if Steps(m)==0

% Put any code here that should only be performed at the start of

% the simulation, for example, to set up initial morphogen values.

% geometry

maxz=max(m.nodes(:,3)); % the 3 stands for z axis, i.e. x,y,z are 1,2,3

minz=min(m.nodes(:,3));

epsilon=0.02;

id\_venous\_pole\_p(m.nodes(:,3)<minz+epsilon)=1;

id\_arterial\_pole\_p(m.nodes(:,3)>maxz-epsilon)=1;

d=maxz-minz;

top=d\*0.77;

bot=d\*0.62;

id\_right\_ventricle\_p((m.nodes(:,3)>bot+minz)&(m.nodes(:,3)<top+minz))=1;

top=d\*0.70;

bot=d\*0.20;

id\_left\_ventricle\_p((m.nodes(:,3)>bot+minz)&(m.nodes(:,3)<top+minz))=1;

top=d\*0.85;

bot=d\*0.75;

id\_oft\_p((m.nodes(:,3)>bot+minz)&(m.nodes(:,3)<top+minz))=1;

top=d\*0.50;

bot=d\*0.00;

id\_atria\_p((m.nodes(:,3)>minz)&(m.nodes(:,3)<top+minz))=1;

top=d\*0.62;

bot=d\*0.60;

id\_iv\_band\_p((m.nodes(:,3)>bot+minz)&(m.nodes(:,3)<top+minz))=1;

top=d\*0.87;

bot=d\*0.62;

id\_art\_band\_p((m.nodes(:,3)>bot+minz)&(m.nodes(:,3)<top+minz))=1;

top=d\*0.60;

bot=d\*0.35;

id\_vein\_band\_p((m.nodes(:,3)>bot+minz)&(m.nodes(:,3)<top+minz))=1;

top=d\*1.0;

bot=d\*0.70;

id\_oft\_band\_p((m.nodes(:,3)>bot+minz)&(m.nodes(:,3)<top+minz))=1;

dorsal=max(m.nodes(:,2));

id\_dorsal\_line\_p(m.nodes(:,2)>dorsal-epsilon/4)=1;

ventral=min(m.nodes(:,2));

id\_ventral\_line\_p(m.nodes(:,2)<ventral+epsilon/4)=0.5; % was 1

left=max(m.nodes(:,1));

right=min(m.nodes(:,1));

id\_right\_p(m.nodes(:,1)<right+epsilon/4)=1;

id\_left\_p(m.nodes(:,1)>left-epsilon/4)=1;

% inner curvature = dorsal\_left

id\_inner\_curvature\_p((m.nodes(:,1)>(left\*sqrt(2)/2)-epsilon) &(m.nodes(:,2)>(dorsal\*sqrt(2)/2)-epsilon))=1;

% outer curvature = ventral\_right

id\_outer\_curvature\_p((m.nodes(:,1)<(right\*sqrt(2)/2)+epsilon) &(m.nodes(:,2)<(ventral\*sqrt(2)/2)+epsilon))=1;

% ventral\_left

id\_ventral\_left\_p((m.nodes(:,1)>(left\*sqrt(2)/2)-epsilon) &(m.nodes(:,2)<(ventral\*sqrt(2)/2)+epsilon))=1;

% dorsal\_right

id\_dorsal\_right\_p((m.nodes(:,1)<(right\*sqrt(2)/2)+1\*epsilon) &(m.nodes(:,2)>(dorsal\*sqrt(2)/2)-1\*epsilon))=1;% was 1\*epsilon

% left and right sides

id\_left\_side\_p(m.nodes(:,1)>0)=1;

id\_right\_side\_p(m.nodes(:,1)<0)=1;

end

m = leaf\_setproperty( m, 'useGrowthTensors', true, 'useMorphogens', true );

m = leaf\_setstrainpolarisation(m, 'mode', 'compression', 'scaling', 5, 'proportional', true, 'useanisotropy', true);

if (realtime>10-dt) && (realtime<10+dt)% set morphogen gradients

id\_sdorsal\_line\_p=1.\*s\_dorsal\_line\_l;

id\_sventral\_line\_p=1.\*s\_ventral\_line\_l;

id\_sleft\_p =1.\*s\_left\_l;

id\_sright\_p =1.\*s\_right\_l;

id\_sart\_band\_p =1.\*s\_art\_band\_l;

id\_svein\_band\_p =1.\*s\_vein\_band\_l;

id\_soft\_p =1.\*s\_oft\_l;

id\_satria\_p =1.\*s\_atria\_l;

id\_siv\_p =1.0.\*s\_iv\_l;

id\_souter\_curvature\_p =1.0.\*s\_outer\_curvature\_l;

id\_sinner\_curvature\_p =1.0.\*s\_inner\_curvature\_l;

id\_sventral\_left\_p =1.0.\*s\_ventral\_left\_l;

id\_sdorsal\_right\_p =1.0.\*s\_dorsal\_right\_l;

id\_sarterial\_pole\_p =1.0.\*s\_arterial\_pole\_l;

id\_sright\_ventricle\_p =1.0.\*s\_right\_ventricle\_l;

id\_soft\_band\_p =1.0.\*s\_oft\_band\_l;

elseif (realtime>10+dt)&& (realtime<20)

BASICGROWTH=0.02;

kaper\_p=6.0\*BASICGROWTH\* (id\_left\_side\_p - id\_right\_side\_p).\*s\_arterial\_pole\_l...% arterial pole rotation

+ 0.7\*BASICGROWTH\*id\_svein\_band\_p...% LV inflation

+ 0.40\*BASICGROWTH\*id\_sright\_ventricle\_p.\*inh(100,id\_sinner\_curvature\_p);% RV inflation

kbper\_p=6.0\*BASICGROWTH\* (id\_left\_side\_p - id\_right\_side\_p).\*s\_arterial\_pole\_l...% arterial pole rotation

+ 0.7\*BASICGROWTH\*id\_svein\_band\_p...% LV inflation

+ 0.40\*BASICGROWTH\*id\_sright\_ventricle\_p.\*inh(100,id\_sinner\_curvature\_p);% RV inflation

kapar\_p=BASICGROWTH \* (5.0.\*id\_siv\_p.\*inh(100,id\_sdorsal\_line\_p));% ventral bending

kbpar\_p=BASICGROWTH \* (1.0);

elseif (realtime>=20)&& (realtime<30)

m=leaf\_fix\_vertex(m,'vertex',[],'dfs','');

m=leaf\_fix\_vertex(m,'vertex',908:909,'dfs','xyz');% attachment of arterial pole (allowing rotation)

m=leaf\_fix\_vertex(m,'vertex',1:30,'dfs','xyz');% all vertices of the venous pole are fixed in xyz

m=leaf\_fix\_vertex(m,'vertex',901:907,'dfs','z');% all vertices of the arterial pole are fixed in z

m=leaf\_fix\_vertex(m,'vertex',910:930,'dfs','z');% all vertices of the arterial pole are fixed in z

% persistent dorsal mesocardium

m=leaf\_fix\_vertex(m,'vertex',38:30:878,'dfs','xyz');% all vertices of the DM are fixed in xyz

m=leaf\_fix\_vertex(m,'vertex',39:30:879,'dfs','xyz');% all vertices of the DM are fixed in xyz

BASICGROWTH=0.02;

kaper\_p=6.0\*BASICGROWTH\* (id\_left\_side\_p - id\_right\_side\_p).\*s\_arterial\_pole\_l...% arterial pole rotation

+ 0.7\*BASICGROWTH\*id\_svein\_band\_p...% LV inflation

+ 0.40\*BASICGROWTH\*id\_sright\_ventricle\_p.\*inh(100,id\_sinner\_curvature\_p);% RV inflation

kbper\_p=6.0\*BASICGROWTH\* (id\_left\_side\_p - id\_right\_side\_p).\*s\_arterial\_pole\_l...% arterial pole rotation

+ 0.7\*BASICGROWTH\*id\_svein\_band\_p...% LV inflation

+ 0.40\*BASICGROWTH\*id\_sright\_ventricle\_p.\*inh(100,id\_sinner\_curvature\_p);% RV inflation

kapar\_p=BASICGROWTH \* (5.0.\*id\_siv\_p.\*inh(100,id\_sdorsal\_line\_p));% ventral bending

kbpar\_p=BASICGROWTH \* (1.0);

elseif (realtime>=30)&& (realtime<40)

m=leaf\_fix\_vertex(m,'vertex',[],'dfs','');

m=leaf\_fix\_vertex(m,'vertex',908:909,'dfs','xyz');% attachment of arterial pole (allowing rotation)

m=leaf\_fix\_vertex(m,'vertex',1:30,'dfs','xyz');% all vertices of the venous pole are fixed in xyz

m=leaf\_fix\_vertex(m,'vertex',901:907,'dfs','z');% all vertices of the arterial pole are fixed in z

m=leaf\_fix\_vertex(m,'vertex',910:930,'dfs','z');% all vertices of the arterial pole are fixed in z

%persistent dorsal mesocardium

m=leaf\_fix\_vertex(m,'vertex',38:30:878,'dfs','xyz');% all vertices of the DM are fixed in xyz

m=leaf\_fix\_vertex(m,'vertex',39:30:879,'dfs','xyz');% all vertices of the DM are fixed in xyz

BASICGROWTH=0.02;

kaper\_p=0.004 ...

+ 0.7\*BASICGROWTH\*id\_svein\_band\_p...% LV inflation

+ 0.40\*BASICGROWTH\*id\_sright\_ventricle\_p.\*inh(100,id\_sinner\_curvature\_p);% RV inflation

kbper\_p=0.004 ...

+ 0.7\*BASICGROWTH\*id\_svein\_band\_p...% LV inflation

+ 0.40\*BASICGROWTH\*id\_sright\_ventricle\_p.\*inh(100,id\_sinner\_curvature\_p);% RV inflation

kapar\_p=BASICGROWTH \* (5.0.\*id\_siv\_p.\*inh(100,id\_sdorsal\_line\_p)...% ventral bending

+ 4.0\*(4\*s\_venous\_pole\_l).\*id\_satria\_p.\*inh(100,id\_sleft\_p));% venous pole asymmetry

kbpar\_p=BASICGROWTH \* (1.0...

+ 4.0\*(4\*s\_venous\_pole\_l).\*id\_satria\_p.\*inh(100,id\_sleft\_p));% venous pole asymmetry

elseif (realtime>=40)&& (realtime<50)

m=leaf\_fix\_vertex(m,'vertex',[],'dfs','');

m=leaf\_fix\_vertex(m,'vertex',908:909,'dfs','xyz');% attachment of arterial pole (allowing rotation)

m=leaf\_fix\_vertex(m,'vertex',1:30,'dfs','xyz');% all vertices of the venous pole are fixed in xyz

m=leaf\_fix\_vertex(m,'vertex',901:907,'dfs','z');% all vertices of the arterial pole are fixed in z

m=leaf\_fix\_vertex(m,'vertex',910:930,'dfs','z');% all vertices of the arterial pole are fixed in z

% persistent dorsal mesocardium

m=leaf\_fix\_vertex(m,'vertex',38:30:878,'dfs','xyz');% all vertices of the DM are fixed in xyz

m=leaf\_fix\_vertex(m,'vertex',39:30:879,'dfs','xyz');% all vertices of the DM are fixed in xyz

BASICGROWTH=0.02;

kaper\_p=0.004 ...

+ 0.7\*BASICGROWTH\*id\_svein\_band\_p...% LV inflation

+ 0.40\*BASICGROWTH\*id\_sright\_ventricle\_p.\*inh(100,id\_sinner\_curvature\_p);% RV inflation

kbper\_p=0.004 ...

+ 0.7\*BASICGROWTH\*id\_svein\_band\_p...% LV inflation

+ 0.40\*BASICGROWTH\*id\_sright\_ventricle\_p.\*inh(100,id\_sinner\_curvature\_p);% RV inflation

kapar\_p=BASICGROWTH \* (1.3...

+ 4.0\*(4\*s\_venous\_pole\_l).\*id\_satria\_p.\*inh(100,id\_sleft\_p));% venous pole asymmetry

kbpar\_p=BASICGROWTH \* (1.3...

+ 4.0\*(4\*s\_venous\_pole\_l).\*id\_satria\_p.\*inh(100,id\_sleft\_p));% venous pole asymmetry

elseif (realtime>=50)&& (realtime<60)

m=leaf\_fix\_vertex(m,'vertex',[],'dfs','');

m=leaf\_fix\_vertex(m,'vertex',908:909,'dfs','xyz');% attachment of arterial pole (allowing rotation)

m=leaf\_fix\_vertex(m,'vertex',1:30,'dfs','xyz');% all vertices of the venous pole are fixed in xyz

m=leaf\_fix\_vertex(m,'vertex',901:907,'dfs','yz');% all vertices of the arterial pole are fixed in yz

m=leaf\_fix\_vertex(m,'vertex',910:930,'dfs','yz');% all vertices of the arterial pole are fixed in yz

% persistent dorsal mesocardium

m=leaf\_fix\_vertex(m,'vertex',38:30:878,'dfs','yz');% all vertices of the DM are fixed in yz

m=leaf\_fix\_vertex(m,'vertex',39:30:879,'dfs','yz');% all vertices of the DM are fixed in yz

BASICGROWTH=0.02;

kaper\_p=0.004 ...

+ 0.7\*BASICGROWTH\*id\_svein\_band\_p...% LV inflation

+ 0.40\*BASICGROWTH\*id\_sright\_ventricle\_p.\*inh(100,id\_sinner\_curvature\_p);% RV inflation

kbper\_p=0.004 ...

+ 0.7\*BASICGROWTH\*id\_svein\_band\_p...% LV inflation

+ 0.40\*BASICGROWTH\*id\_sright\_ventricle\_p.\*inh(100,id\_sinner\_curvature\_p);% RV inflation

kapar\_p=BASICGROWTH \* (1.3...

+ 4.0\*(4\*s\_venous\_pole\_l).\*id\_satria\_p.\*inh(100,id\_sleft\_p));% venous pole asymmetry

kbpar\_p=BASICGROWTH \* (1.3...

+ 4.0\*(4\*s\_venous\_pole\_l).\*id\_satria\_p.\*inh(100,id\_sleft\_p)); % venous pole asymmetry

elseif (realtime>=60)&& (realtime<70)

m=leaf\_fix\_vertex(m,'vertex',[],'dfs','');

m=leaf\_fix\_vertex(m,'vertex',908:909,'dfs','xyz');% attachment of arterial pole

m=leaf\_fix\_vertex(m,'vertex',1:30,'dfs','xyz');% all vertices of the venous pole are fixed in xyz

m=leaf\_fix\_vertex(m,'vertex',901:907,'dfs','yz');% all vertices of the arterial pole are fixed in yz

m=leaf\_fix\_vertex(m,'vertex',910:930,'dfs','yz');% all vertices of the arterial pole are fixed in yz

% persistent dorsal mesocardium

m=leaf\_fix\_vertex(m,'vertex',38:30:878,'dfs','yz');% all vertices of the DM are fixed in yz

m=leaf\_fix\_vertex(m,'vertex',39:30:879,'dfs','yz');% all vertices of the DM are fixed in yz

BASICGROWTH=0.02;

kaper\_p=0.008 ...

+ 0.7\*BASICGROWTH\*id\_svein\_band\_p...% LV inflation

+ 0.4\*BASICGROWTH\*id\_sart\_band\_p.\*inh(100,id\_sinner\_curvature\_p);% RV inflation

kbper\_p=0.008 ...

+ 0.7\*BASICGROWTH\*id\_svein\_band\_p...% LV inflation

+ 0.4\*BASICGROWTH\*id\_sart\_band\_p.\*inh(100,id\_sinner\_curvature\_p);% RV inflation

kapar\_p=BASICGROWTH \* (1.3);

kbpar\_p=BASICGROWTH \* (1.3);

elseif (realtime>=70)&& (realtime<80)

m=leaf\_fix\_vertex(m,'vertex',[],'dfs','');

m=leaf\_fix\_vertex(m,'vertex',908:909,'dfs','xyz');% attachment of arterial pole

m=leaf\_fix\_vertex(m,'vertex',1:30,'dfs','xyz');% all vertices of the venous pole are fixed in xyz

m=leaf\_fix\_vertex(m,'vertex',901:907,'dfs','yz');% all vertices of the arterial pole are fixed in yz

m=leaf\_fix\_vertex(m,'vertex',910:930,'dfs','yz');% all vertices of the arterial pole are fixed in yz

% persistent dorsal mesocardium

m=leaf\_fix\_vertex(m,'vertex',38:30:878,'dfs','yz');% all vertices of the DM are fixed in yz

m=leaf\_fix\_vertex(m,'vertex',39:30:879,'dfs','yz');% all vertices of the DM are fixed in yz

BASICGROWTH=0.02;

kaper\_p=0.008 ...

+ 0.7\*BASICGROWTH\*id\_svein\_band\_p...% LV inflation

+ 0.4\*BASICGROWTH\*id\_sart\_band\_p.\*inh(100,id\_sinner\_curvature\_p);% RV inflation

kbper\_p=0.008 ...

+ 0.7\*BASICGROWTH\*id\_svein\_band\_p...% LV inflation

+ 0.4\*BASICGROWTH\*id\_sart\_band\_p.\*inh(100,id\_sinner\_curvature\_p);% RV inflation

kapar\_p=BASICGROWTH \* (1.3);

kbpar\_p=BASICGROWTH \* (1.3);

elseif (realtime>=80)&& (realtime<120)

m=leaf\_fix\_vertex(m,'vertex',[],'dfs','');

m=leaf\_fix\_vertex(m,'vertex',908:909,'dfs','xyz');% attachment of arterial pole

m=leaf\_fix\_vertex(m,'vertex',1:30,'dfs','xyz');% all vertices of the venous pole are fixed in xyz

m=leaf\_fix\_vertex(m,'vertex',901:907,'dfs','yz');% all vertices of the arterial pole are fixed in yz

m=leaf\_fix\_vertex(m,'vertex',910:930,'dfs','yz');% all vertices of the arterial pole are fixed in yz

% persistent dorsal mesocardium

m=leaf\_fix\_vertex(m,'vertex',38:30:878,'dfs','yz');% all vertices of the DM are fixed in (was yz)

m=leaf\_fix\_vertex(m,'vertex',39:30:879,'dfs','yz');% all vertices of the DM are fixed in (was yz)

BASICGROWTH=0.02;

kaper\_p=0.008 ...

+ 0.7\*BASICGROWTH\*id\_svein\_band\_p...% LV inflation

+ 0.4\*BASICGROWTH\*id\_sart\_band\_p.\*inh(100,id\_sinner\_curvature\_p);% RV inflation

kbper\_p=0.008 ...

+ 0.7\*BASICGROWTH\*id\_svein\_band\_p...% LV inflation

+ 0.4\*BASICGROWTH\*id\_sart\_band\_p.\*inh(100,id\_sinner\_curvature\_p);% RV inflation

kapar\_p=BASICGROWTH \* (1.3);

kbpar\_p=BASICGROWTH \* (1.3);

end

m.mgen\_production(:,polariser\_i) = 0.1...

\*(1.0\*id\_venous\_pole\_p...

+ 1.0 ...

- P.\*(id\_arterial\_pole\_l));% polarizer gradient from source to sink

m.mgen\_production(:,s\_iv\_i) = 0.1\*(id\_iv\_band\_p);

m.mgen\_production(:,s\_art\_band\_i) = 0.1\*(id\_art\_band\_p);

m.mgen\_production(:,s\_vein\_band\_i) = 0.1\*id\_vein\_band\_p;

m.mgen\_production(:,s\_oft\_i) = 0.1\*id\_oft\_p;

m.mgen\_production(:,s\_atria\_i) = 0.1\*id\_atria\_p;

m.mgen\_production(:,s\_dorsal\_line\_i) = 0.1\*(id\_dorsal\_line\_p);

m.mgen\_production(:,s\_ventral\_line\_i) = 0.1\*(id\_ventral\_line\_p);

m.mgen\_production(:,s\_left\_i) = 0.1\*(id\_left\_p);

m.mgen\_production(:,s\_right\_i) = 0.1\*(id\_right\_p);

m.mgen\_production(:,s\_outer\_curvature\_i) = 0.1\*(id\_outer\_curvature\_p);

m.mgen\_production(:,s\_inner\_curvature\_i) = 0.1\*(id\_inner\_curvature\_p);

m.mgen\_production(:,s\_ventral\_left\_i) = 0.1\*(id\_ventral\_left\_p);

m.mgen\_production(:,s\_dorsal\_right\_i) = 0.1\*(id\_dorsal\_right\_p);

m.mgen\_production(:,s\_arterial\_pole\_i) = 0.1\*(id\_arterial\_pole\_p);

m.mgen\_production(:,s\_venous\_pole\_i) = 0.1\*(id\_venous\_pole\_p);

m.mgen\_production(:,s\_right\_ventricle\_i) = 0.1\*(id\_right\_ventricle\_p);

m.mgen\_production(:,s\_oft\_band\_i) = 0.1\*(id\_oft\_band\_p);

%%% END OF USER CODE: MORPHOGEN INTERACTIONS

%%% SECTION 3: INSTALLING MODIFIED VALUES BACK INTO MESH STRUCTURE

%%% AUTOMATICALLY GENERATED CODE: DO NOT EDIT.

m.morphogens(:,polariser\_i) = P;

m.morphogens(:,kapar\_i) = kapar\_p;

m.morphogens(:,kaper\_i) = kaper\_p;

m.morphogens(:,kbpar\_i) = kbpar\_p;

m.morphogens(:,kbper\_i) = kbper\_p;

m.morphogens(:,knor\_i) = knor\_p;

m.morphogens(:,strainret\_i) = strainret\_p;

m.morphogens(:,arrest\_i) = arrest\_p;

m.morphogens(:,id\_venous\_pole\_i) = id\_venous\_pole\_p;

m.morphogens(:,id\_arterial\_pole\_i) = id\_arterial\_pole\_p;

m.morphogens(:,id\_right\_ventricle\_i) = id\_right\_ventricle\_p;

m.morphogens(:,id\_left\_ventricle\_i) = id\_left\_ventricle\_p;

m.morphogens(:,id\_oft\_i) = id\_oft\_p;

m.morphogens(:,id\_atria\_i) = id\_atria\_p;

m.morphogens(:,id\_art\_band\_i) = id\_art\_band\_p;

m.morphogens(:,id\_vein\_band\_i) = id\_vein\_band\_p;

m.morphogens(:,id\_iv\_band\_i) = id\_iv\_band\_p;

m.morphogens(:,id\_dorsal\_line\_i) = id\_dorsal\_line\_p;

m.morphogens(:,id\_ventral\_line\_i) = id\_ventral\_line\_p;

m.morphogens(:,id\_right\_i) = id\_right\_p;

m.morphogens(:,id\_left\_i) = id\_left\_p;

m.morphogens(:,id\_inner\_curvature\_i) = id\_inner\_curvature\_p;

m.morphogens(:,id\_outer\_curvature\_i) = id\_outer\_curvature\_p;

m.morphogens(:,id\_sdorsal\_line\_i) = id\_sdorsal\_line\_p;

m.morphogens(:,id\_sleft\_i) = id\_sleft\_p;

m.morphogens(:,id\_siv\_i) = id\_siv\_p;

m.morphogens(:,s\_iv\_i) = s\_iv\_p;

m.morphogens(:,s\_dorsal\_line\_i) = s\_dorsal\_line\_p;

m.morphogens(:,s\_left\_i) = s\_left\_p;

m.morphogens(:,s\_art\_band\_i) = s\_art\_band\_p;

m.morphogens(:,id\_sart\_band\_i) = id\_sart\_band\_p;

m.morphogens(:,s\_vein\_band\_i) = s\_vein\_band\_p;

m.morphogens(:,id\_svein\_band\_i) = id\_svein\_band\_p;

m.morphogens(:,s\_ventral\_line\_i) = s\_ventral\_line\_p;

m.morphogens(:,id\_sventral\_line\_i) = id\_sventral\_line\_p;

m.morphogens(:,s\_right\_i) = s\_right\_p;

m.morphogens(:,id\_sright\_i) = id\_sright\_p;

m.morphogens(:,id\_ventral\_left\_i) = id\_ventral\_left\_p;

m.morphogens(:,s\_outer\_curvature\_i) = s\_outer\_curvature\_p;

m.morphogens(:,id\_souter\_curvature\_i) = id\_souter\_curvature\_p;

m.morphogens(:,s\_ventral\_left\_i) = s\_ventral\_left\_p;

m.morphogens(:,id\_sventral\_left\_i) = id\_sventral\_left\_p;

m.morphogens(:,s\_inner\_curvature\_i) = s\_inner\_curvature\_p;

m.morphogens(:,id\_sinner\_curvature\_i) = id\_sinner\_curvature\_p;

m.morphogens(:,id\_dorsal\_right\_i) = id\_dorsal\_right\_p;

m.morphogens(:,s\_dorsal\_right\_i) = s\_dorsal\_right\_p;

m.morphogens(:,id\_sdorsal\_right\_i) = id\_sdorsal\_right\_p;

m.morphogens(:,s\_oft\_i) = s\_oft\_p;

m.morphogens(:,id\_soft\_i) = id\_soft\_p;

m.morphogens(:,s\_atria\_i) = s\_atria\_p;

m.morphogens(:,id\_satria\_i) = id\_satria\_p;

m.morphogens(:,id\_clone\_density\_i) = id\_clone\_density\_p;

m.morphogens(:,s\_arterial\_pole\_i) = s\_arterial\_pole\_p;

m.morphogens(:,s\_venous\_pole\_i) = s\_venous\_pole\_p;

m.morphogens(:,id\_left\_side\_i) = id\_left\_side\_p;

m.morphogens(:,id\_right\_side\_i) = id\_right\_side\_p;

m.morphogens(:,id\_sarterial\_pole\_i) = id\_sarterial\_pole\_p;

m.morphogens(:,s\_right\_ventricle\_i) = s\_right\_ventricle\_p;

m.morphogens(:,id\_sright\_ventricle\_i) = id\_sright\_ventricle\_p;

m.morphogens(:,id\_oft\_band\_i) = id\_oft\_band\_p;

m.morphogens(:,id\_soft\_band\_i) = id\_soft\_band\_p;

m.morphogens(:,s\_oft\_band\_i) = s\_oft\_band\_p;

%%% USER CODE: FINALISATION

% In this section you may modify the mesh in any way whatsoever.

%%% END OF USER CODE: FINALISATION

end

%%% USER CODE: SUBFUNCTIONS

function m = local\_setproperties( m )

% This function is called at time zero in the INITIALISATION section of the

% interaction function. It provides commands to set each of the properties

% that are contained in m.globalProps. Uncomment whichever ones you would

% like to set yourself, and put in whatever value you want.

%

% Some of these properties are for internal use only and should never be

% set by the user. At some point these will be moved into a different

% component of m, but for the present, just don't change anything unless

% you know what it is you're changing.

% m = leaf\_setproperty( m, 'trinodesvalid', true );

% m = leaf\_setproperty( m, 'prismnodesvalid', true );

% m = leaf\_setproperty( m, 'thicknessRelative', 0.100000 );

% m = leaf\_setproperty( m, 'thicknessArea', 0.000000 );

% m = leaf\_setproperty( m, 'activeGrowth', 1.000000 );

% m = leaf\_setproperty( m, 'displayedGrowth', 1 );

% m = leaf\_setproperty( m, 'allowNegativeGrowth', true );

% m = leaf\_setproperty( m, 'usePrevDispAsEstimate', true );

% m = leaf\_setproperty( m, 'mingradient', 0.000000 );

% m = leaf\_setproperty( m, 'thresholdsq', 0.005874 );

% m = leaf\_setproperty( m, 'splitmargin', 1.000000 );

% m = leaf\_setproperty( m, 'thresholdmgen', 0.500000 );

% m = leaf\_setproperty( m, 'bulkmodulus', 1.000000 );

% m = leaf\_setproperty( m, 'poissonsRatio', 0.300000 );

% m = leaf\_setproperty( m, 'timestep', 2.500000 );

% m = leaf\_setproperty( m, 'timeunitname', '' );

% m = leaf\_setproperty( m, 'distunitname', 'mm' );

% m = leaf\_setproperty( m, 'validateMesh', true );

% m = leaf\_setproperty( m, 'allowSplitLongFEM', false );

% m = leaf\_setproperty( m, 'longSplitThresholdPower', 0.000000 );

% m = leaf\_setproperty( m, 'allowSplitBentFEM', false );

% m = leaf\_setproperty( m, 'allowSplitBio', true );

% m = leaf\_setproperty( m, 'allowFlipEdges', false );

% m = leaf\_setproperty( m, 'allowElideEdges', false );

% m = leaf\_setproperty( m, 'mincellangle', 0.200000 );

% m = leaf\_setproperty( m, 'alwaysFlat', 0.000000 );

% m = leaf\_setproperty( m, 'flatten', false );

% m = leaf\_setproperty( m, 'flattenratio', 1.000000 );

m = leaf\_setproperty( m, 'useGrowthTensors', true, 'useMorphogens', true );

% m = leaf\_setproperty( m, 'plasticGrowth', false );

% m = leaf\_setproperty( m, 'totalinternalrotation', 1.239019 );

% m = leaf\_setproperty( m, 'stepinternalrotation', 2.000000 );

% m = leaf\_setproperty( m, 'showinternalrotation', false );

% m = leaf\_setproperty( m, 'internallyrotated', false );

% m = leaf\_setproperty( m, 'maxFEcells', 0.000000 );

% m = leaf\_setproperty( m, 'inittotalcells', 0.000000 );

% m = leaf\_setproperty( m, 'maxBioAcells', 0.000000 );

% m = leaf\_setproperty( m, 'maxBioBcells', 0.000000 );

% m = leaf\_setproperty( m, 'colors', (3 values) );

% m = leaf\_setproperty( m, 'colorvariation', 1.000000 );

% m = leaf\_setproperty( m, 'colorparams', (6 values) );

% m = leaf\_setproperty( m, 'freezing', 0.000000 );

% m = leaf\_setproperty( m, 'mgen\_interaction', (unknown type ''function\_handle'') );

% m = leaf\_setproperty( m, 'mgen\_interactionName', 'heart35i20b' );

% m = leaf\_setproperty( m, 'allowInteraction', 1.000000 );

% m = leaf\_setproperty( m, 'interactionValid', true );

% m = leaf\_setproperty( m, 'gaussInfo', (unknown type ''struct'') );

% m = leaf\_setproperty( m, 'stitchDFs', [] );

% m = leaf\_setproperty( m, 'D', (36 values) );

% m = leaf\_setproperty( m, 'C', (36 values) );

% m = leaf\_setproperty( m, 'G', (6 values) );

% m = leaf\_setproperty( m, 'solver', 'cgs' );

% m = leaf\_setproperty( m, 'solvertolerance', 0.001000 );

% m = leaf\_setproperty( m, 'diffusiontolerance', 0.000010 );

% m = leaf\_setproperty( m, 'maxIters', 40.000000 );

% m = leaf\_setproperty( m, 'maxsolvetime', 1000.000000 );

% m = leaf\_setproperty( m, 'cgiters', 52.000000 );

% m = leaf\_setproperty( m, 'simsteps', 0.000000 );

% m = leaf\_setproperty( m, 'stepsperrender', 0.000000 );

% m = leaf\_setproperty( m, 'growthEnabled', true );

% m = leaf\_setproperty( m, 'diffusionEnabled', true );

% m = leaf\_setproperty( m, 'makemovie', 0.000000 );

% m = leaf\_setproperty( m, 'moviefile', '' );

% m = leaf\_setproperty( m, 'codec', 'None' );

% m = leaf\_setproperty( m, 'autonamemovie', true );

% m = leaf\_setproperty( m, 'overwritemovie', false );

% m = leaf\_setproperty( m, 'framesize', (3 values) );

% m = leaf\_setproperty( m, 'mov', [] );

% m = leaf\_setproperty( m, 'jiggleProportion', 1.000000 );

% m = leaf\_setproperty( m, 'cvtperiter', 0.200000 );

% m = leaf\_setproperty( m, 'boingNeeded', false );

% m = leaf\_setproperty( m, 'initialArea', 0.940756 );

% m = leaf\_setproperty( m, 'bendunitlength', 0.969926 );

% m = leaf\_setproperty( m, 'targetRelArea', 1.000000 );

% m = leaf\_setproperty( m, 'defaultinterp', 'min' );

% m = leaf\_setproperty( m, 'readonly', false );

% m = leaf\_setproperty( m, 'projectdir', 'C:\Documents and Settings\GMD\Mes documents\Le Garrec\heart morpho\Coen\GFToolbox\Heart\_models' );

% m = leaf\_setproperty( m, 'modelname', 'Heart35i20b' );

% m = leaf\_setproperty( m, 'allowsave', 1.000000 );

% m = leaf\_setproperty( m, 'addedToPath', true );

% m = leaf\_setproperty( m, 'bendsplit', 0.300000 );

% m = leaf\_setproperty( m, 'dorsaltop', true );

% m = leaf\_setproperty( m, 'comment', '' );

% m = leaf\_setproperty( m, 'bioAsplitcells', 1.000000 );

% m = leaf\_setproperty( m, 'bioApullin', 0.142857 );

% m = leaf\_setproperty( m, 'bioAfakepull', 0.202073 );

% m = leaf\_setproperty( m, 'coderevision', 3530 );

% m = leaf\_setproperty( m, 'coderevisiondate', '2011-05-10 15:51:48.530599' );

% m = leaf\_setproperty( m, 'modelrevision', 0.000000 );

% m = leaf\_setproperty( m, 'modelrevisiondate', '' );

% m = leaf\_setproperty( m, 'vxgrad', (108 values) );

% m = leaf\_setproperty( m, 'lengthscale', 1.500000 );

% m = leaf\_setproperty( m, 'performinternalrotation', false );

% m = leaf\_setproperty( m, 'legendTemplate', '%T: %q\n%m' );

% m = leaf\_setproperty( m, 'relativepolgrad', false );

% m = leaf\_setproperty( m, 'userpolarisation', false );

% m = leaf\_setproperty( m, 'usepolfreezebc', false );

% m = leaf\_setproperty( m, 'starttime', 0.000000 );

% m = leaf\_setproperty( m, 'displayedMulti', [] );

% m = leaf\_setproperty( m, 'splitmorphogen', '' );

% m = leaf\_setproperty( m, 'scalebarvalue', 0.000000 );

% m = leaf\_setproperty( m, 'rectifyverticals', false );

% m = leaf\_setproperty( m, 'flattenforceconvex', true );

% m = leaf\_setproperty( m, 'defaultazimuth', -45.000000 );

% m = leaf\_setproperty( m, 'defaultelevation', 33.750000 );

% m = leaf\_setproperty( m, 'defaultroll', 0.000000 );

% m = leaf\_setproperty( m, 'defaultViewParams', (unknown type ''struct'') );

% m = leaf\_setproperty( m, 'interactive', false );

% m = leaf\_setproperty( m, 'RecordMeshes', (unknown type ''struct'') );

% m = leaf\_setproperty( m, 'thicknessMode', 'scaled' );

% m = leaf\_setproperty( m, 'usefrozengradient', true );

% m = leaf\_setproperty( m, 'perturbInitGrowthEstimate', 0.000010 );

% m = leaf\_setproperty( m, 'perturbRelGrowthEstimate', 0.010000 );

% m = leaf\_setproperty( m, 'perturbDiffusionEstimate', 0.000100 );

% m = leaf\_setproperty( m, 'resetRand', false );

% m = leaf\_setproperty( m, 'unitbulkmodulus', true );

% m = leaf\_setproperty( m, 'bioApresplitproc', '' );

% m = leaf\_setproperty( m, 'bioApostsplitproc', '' );

% m = leaf\_setproperty( m, 'canceldrift', false );

% m = leaf\_setproperty( m, 'solverprecision', 'double' );

% m = leaf\_setproperty( m, 'solvertolerancemethod', 'norm' );

% m = leaf\_setproperty( m, 'allowsparse', true );

% m = leaf\_setproperty( m, 'savedrunname', '' );

% m = leaf\_setproperty( m, 'savedrundesc', '' );

end