function m = gpt\_model\_paper\_fromday10\_20151101( m )

%m = gpt\_model\_paper\_fromday10\_20151101( m )

% Morphogen interaction function.

% Written at 2015-11-03 11:22:15.

% GFtbox revision 5296, 2015-10-06 15:00.

% 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.

if (Steps(m)==0) && m.globalDynamicProps.doinit % First iteration

% Set up names for variant models. Useful for running multiple models on a cluster.

m.userdata.ranges.modelname.range = {'MODEL7\_div', 'MODEL7\_div\_1', 'MODEL7\_WT', 'MODEL7\_WT\_2', 'MODEL7\_WT\_3'}; % CLUSTER

m.userdata.ranges.modelname.index =2; % CLUSTER

%m = leaf\_setzeroz( m ); % keeps the noise in the canvas to zero

m = leaf\_bowlz(m, 0.01 ); %induce a slightly curve in the medio-lateral axis of the canvas

m = leaf\_setthickness(m, 0.06); %thickness of canvas, relates to the 60 um of the real data

%ID\_EDGE

m = leaf\_mgen\_edge( m, 'ID\_EDGE', 1);

% m = leaf\_fix\_vertex( m, 'ID\_EDGE', 'dfs', 'z' );

% m = leaf\_fix\_vertex( m, 'vertex', find(m.nodes(:,2)>=min(m.nodes(:,2)+0.01)), 'dfs', 'z' );

% m = leaf\_fix\_vertex( m, 'vertex', find(m.nodes(:,1)>=max(m.nodes(:,1)-0.01)), 'dfs', 'z' );

%

%m = leaf\_fix\_vertex( m, 'vertex', find(m.nodes(:,1)<=min(m.nodes(:,1)+0.01)), 'dfs', 'xyz' );

%m = leaf\_fix\_vertex( m, 'vertex', find(m.nodes(:,1)>=max(m.nodes(:,1)-0.01)), 'dfs', 'xyz' );

% set colour of polariser gradient arrows

m=leaf\_plotoptions(m,'highgradcolor',[0,0,0],'lowgradcolor',[0,0,0]);

m=leaf\_plotoptions(m,'decorscale',1.5);

m=leaf\_plotoptions(m,'arrowthickness',1.3);

m.globalProps.timestep=1; %each step relates to 10 hours in real time

end

modelname = m.userdata.ranges.modelname.range{m.userdata.ranges.modelname.index}; % CLUSTER

disp(sprintf('model2',mfilename, modelname));

switch modelname

case {'MODEL1','MODEL2'}

otherwise

% If you reach here, you probably forgot a case.

end

%%% END OF USER CODE: INITIALISATION

%%% SECTION 1: ACCESSING MORPHOGENS AND TIME.

%%% AUTOMATICALLY GENERATED CODE: DO NOT EDIT.

global gNEW\_KA\_PAR gNEW\_KA\_PER gNEW\_KB\_PAR gNEW\_KB\_PER

global gNEW\_K\_NOR gNEW\_POLARISER gNEW\_STRAINRET gNEW\_ARREST

polariser\_i = gNEW\_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\_sink\_i,id\_sink\_p,id\_sink\_a,id\_sink\_l] = getMgenLevels( m, 'ID\_SINK' );

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

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

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

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

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

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

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

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

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

[v\_flower\_i,v\_flower\_p,v\_flower\_a,v\_flower\_l] = getMgenLevels( m, 'V\_FLOWER' );

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

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

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

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

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

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

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

[v\_speckareal\_i,v\_speckareal\_p,v\_speckareal\_a,v\_speckareal\_l] = getMgenLevels( m, 'V\_SPECKAREAL' );

[v\_specaniso\_i,v\_specaniso\_p,v\_specaniso\_a,v\_specaniso\_l] = getMgenLevels( m, 'V\_SPECANISO' );

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

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

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

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

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

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

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

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

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

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

[v\_kaniso\_i,v\_kaniso\_p,v\_kaniso\_a,v\_kaniso\_l] = getMgenLevels( m, 'V\_KANISO' );

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

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

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

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

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

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

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

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

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

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

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

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

[v\_diff\_i,v\_diff\_p,v\_diff\_a,v\_diff\_l] = getMgenLevels( m, 'V\_DIFF' );

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

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

% Mesh type: rectangle

% base: 0

% centre: 0

% layers: 0

% randomness: 0.1

% thickness: 0

% xdivs: 106

% xwidth: 1.06

% ydivs: 12

% ywidth: 0.12

% Morphogen Diffusion Decay Dilution Mutant

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

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

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

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

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

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

% POLARISER ---- ---- ---- ----

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

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

% ID\_SINK ---- ---- ---- ----

% ID\_SOURCE ---- ---- ---- ----

% ID\_EDGE ---- ---- ---- ----

% ID\_MED ---- ---- ---- ----

% ID\_JUNCTION ---- ---- ---- ----

% ID\_RIM ---- ---- ---- ----

% S\_SOURCE ---- ---- ---- ----

% S\_RIM ---- ---- ---- ----

% S\_MED ---- ---- ---- ----

% S\_JUNCTION ---- ---- ---- ----

% V\_FLOWER ---- ---- ---- ----

% ID\_SPOTTY ---- ---- ---- ----

% ID\_STRIPY ---- ---- ---- ----

% ID\_LATE ---- ---- ---- ----

% ID\_SINUS ---- ---- ---- ----

% S\_SINUS ---- ---- ---- ----

% S\_SINK ---- ---- ---- ----

% ID\_LIPCLIFF ---- ---- ---- ----

% V\_SPECKAREAL ---- ---- ---- ----

% V\_SPECANISO ---- ---- ---- ----

% ID\_EARLY ---- ---- ---- ----

% ID\_PALATE ---- ---- ---- ----

% ID\_HINGE ---- ---- ---- ----

% ID\_VENTMIDVEIN ---- ---- ---- ----

% ID\_SECVEIN ---- ---- ---- ----

% S\_VENTMIDVEIN ---- 0.02 ---- ----

% S\_SECVEIN ---- ---- ---- ----

% ID\_DIV ---- ---- ---- ----

% S\_DIV ---- ---- ---- ----

% ID\_LAT ---- ---- ---- ----

% V\_KANISO ---- ---- ---- ----

% ID\_LIPBEND ---- ---- ---- ----

% ID\_TUBE ---- ---- ---- ----

% ID\_RAD ---- ---- ---- ----

% S\_RAD 0.0005 0.005 ---- ----

% ID\_LIPDISTAL ---- ---- ---- ----

% ID\_LIP ---- ---- ---- ----

% ID\_TWINPEAKS ---- ---- ---- ----

% ID\_CHEEKS ---- ---- ---- ----

% ID\_FOCI ---- ---- ---- ----

% S\_FOCI ---- ---- ---- ----

% ID\_DORSALEDGE ---- ---- ---- ----

% S\_DORSALEDGE ---- ---- ---- ----

% V\_DIFF ---- ---- ---- ----

% ID\_LOBE ---- ---- ---- ----

% ID\_SUBDIVISION ---- ---- ---- ----

%%% USER CODE: MORPHOGEN INTERACTIONS

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

% alter the set of nodes.

%ID\_SUBDIVISION - Region that subdivides

id\_subdivision\_p (m.nodes(:,2) < 0.0391 & m.nodes(:,2) > - 0.0151) = 1

% m = leaf\_fix\_vertex( m, 'ID\_EDGE', 'dfs', 'z' );

if (Steps(m)==2) && m.globalDynamicProps.doinit % Initialisation code.

switch modelname

case {'MODEL7\_div','MODEL7\_div\_1', 'MODEL7\_WT', 'MODEL7\_WT\_2', 'MODEL7\_WT\_3'}

m = leaf\_mgen\_conductivity( m, 'POLARISER', 0.001 ); %specifies the diffusion rate of polariser

m = leaf\_mgen\_absorption( m, 'POLARISER', 0.01 ); % specifies degradation rate of polariser

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

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

id\_sink\_p(top) = 1;

id\_source\_p(bottom) = 1;

P(:) = 0.1;

P(bottom) = 1;

P(top) = 0.07;

m.morphogenclamp(bottom, polariser\_i) = 1;

m.morphogenclamp(top, polariser\_i) = 1;

%Dorso-Ventral Regional factors (IDs) and Signals (Ss)

%ID\_DIV

id\_div\_p (m.nodes(:,1) > - 0.171 & m.nodes(:,1) < 0.171) = 1;

%S\_DIV

s\_div\_p(:) = 0;

s\_div\_p(:) = id\_div\_p(:);

m = leaf\_mgen\_conductivity( m, 'S\_DIV', 0.005 );

m = leaf\_mgen\_absorption( m, 'S\_DIV', 0.05 );

%ID\_LAT

id\_lat\_p (m.nodes(:,1) < - 0.169 | m.nodes(:,1) > 0.171) = 1;

%ID\_RAD

id\_rad\_p (m.nodes(:,1) < - 0.5 | m.nodes(:,1) > 0.5) = 1;

%S\_RAD

s\_rad\_p(:) = 0;

s\_rad\_p(:) = 2\*id\_rad\_p(:);

m = leaf\_mgen\_conductivity( m, 'S\_RAD', 0.0002 );

m = leaf\_mgen\_absorption( m, 'S\_RAD', 0.002 );

%Proximo-distal Regional factors (IDs) and Signals (Ss)

%ID\_RIM

id\_rim\_p (:) = 0;

id\_rim\_p(m.nodes(:,2) < 0.020 & m.nodes(:,2) > - 0.0151) = 1;

%S\_RIM

s\_rim\_p(:) = 0;

s\_rim\_p(:) = id\_rim\_p;

m = leaf\_mgen\_conductivity( m, 'S\_RIM', 0.0004 );

m = leaf\_mgen\_absorption( m, 'S\_RIM', 0.004 );

%ID\_LOBE

id\_lobe\_p (:) = 0;

id\_lobe\_p (m.nodes(:,2) > 0.049) = 1;

%ID\_LIP

id\_lip\_p (:) = 0;

id\_lip\_p (m.nodes(:,2) > 0.018 & m.nodes(:,2)< 0.0501) = 1;

%ID\_LIPDISTAL

id\_lipdistal\_p (:) = 0;

id\_lipdistal\_p(m.nodes(:,2) > 0.0391 & m.nodes(:,2)< 0.0501) = 1;

%ID\_LIPBEND

id\_lipbend\_p (:) = 0;

id\_lipbend\_p(m.nodes(:,2) > 0.0391 & m.nodes(:,2)< 0.0401) = 1;

%ID\_LIPCLIFF

id\_lipcliff\_p (:) = 0;

id\_lipcliff\_p (m.nodes(:,2) < 0.0391 & m.nodes(:,2) > 0.018) = 1;

%PALATE

id\_palate\_p (:) = 0;

id\_palate\_p (m.nodes(:,2) < - 0.0155 & m.nodes (:,2) > -0.0491) = 1;

%TUBE

id\_tube\_p (:) = 0;

id\_tube\_p(m.nodes(:,2) < - 0.0491) = 1;

%Medio-Lateral Regional factors (IDs) and Signals (Ss)

%ID\_JUNCTION

id\_junction\_p((m.nodes(:,1) < 0.180 & m.nodes(:,1) > 0.165)) = 1;

id\_junction\_p((m.nodes(:,1) > - 0.180 & m.nodes(:,1) < - 0.162) ) = 1;

id\_junction\_p((m.nodes(:,1) > 0.52 | m.nodes(:,1) < - 0.52) ) = 1;

id\_junction\_p = id\_junction\_p ;

%S\_JUNCTION

s\_junction\_p = 10\*id\_junction\_p;

m = leaf\_mgen\_conductivity( m, 'S\_JUNCTION', 0.001 );

m = leaf\_mgen\_absorption( m, 'S\_JUNCTION', 0.01 );

%ID\_MED

id\_med\_p (m.nodes(:,1) > 0.352 & m.nodes(:,1) < 0.367) = 1;

id\_med\_p (m.nodes(:,1) < - 0.352 & m.nodes(:,1) > - 0.367) = 1;

id\_med\_p (m.nodes(:,1) > - 0.008 & m.nodes(:,1) < 0.008) = 1;

%S\_MED

s\_med\_p(:) = 0;

s\_med\_p(:) = id\_med\_p(:);

m = leaf\_mgen\_conductivity( m, 's\_med', 0.001 );

m = leaf\_mgen\_absorption( m, 's\_med', 0.01 );

%ID\_secvein

id\_secvein\_p (m.nodes(:,1) > 0.261 & m.nodes(:,1) < 0.277) = 1;

id\_secvein\_p (m.nodes(:,1) < - 0.261 & m.nodes(:,1) > - 0.277) = 1;

id\_secvein\_p (m.nodes(:,1) > 0.431 & m.nodes(:,1) < 0.447) = 1;

id\_secvein\_p (m.nodes(:,1) < - 0.431 & m.nodes(:,1) > - 0.447) = 1;

id\_secvein\_p (m.nodes(:,1) > 0.081 & m.nodes(:,1) < 0.097) = 1;

id\_secvein\_p (m.nodes(:,1) < - 0.081 & m.nodes(:,1) > - 0.097) = 1;

%S\_secvein

s\_secvein\_p(:) = 0;

s\_secvein\_p(:) = id\_secvein\_p(:);

m = leaf\_mgen\_conductivity( m, 'S\_SECVEIN', 0.0002 );

m = leaf\_mgen\_absorption( m, 'S\_SECVEIN', 0.002 );

%Other factors we might not need at all

%ID\_EDGE

m = leaf\_mgen\_edge( m, 'ID\_EDGE', 1);

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

%ID\_HINGE

id\_hinge\_p (m.nodes(:,1) < - 0.361 | m.nodes(:,1) > 0.361) = 1;

%S\_SOURCE

s\_source\_p(:) = 0;

s\_source\_p(:) = id\_source\_p;

m = leaf\_mgen\_conductivity( m, 'S\_SOURCE', 0.0002 );

m = leaf\_mgen\_absorption( m, 'S\_SOURCE', 0.002 );

%S\_SINK

s\_sink\_p(:) = 0;

s\_sink\_p(:) = id\_sink\_p;

m = leaf\_mgen\_conductivity( m, 'S\_SINK', 0.0002 );

m = leaf\_mgen\_absorption( m, 'S\_SINK', 0.002 );

%ID\_FOCI

id\_foci\_p = id\_rim\_p .\* id\_junction\_p;

%S\_FOCI

s\_foci\_p(:) = 0;

s\_foci\_p(:) = id\_foci\_p;

m = leaf\_mgen\_conductivity( m, 'S\_FOCI', 0.0004 );

m = leaf\_mgen\_absorption( m, 'S\_FOCI', 0.004 );

end

end

if(Steps(m)==6)

m = leaf\_mgen\_conductivity( m, 'POLARISER', 0 ); %specifies the diffusion rate of polariser

m = leaf\_mgen\_absorption( m, 'POLARISER', 0 );

m = leaf\_mgen\_conductivity( m, 'S\_MED', 0 );

m = leaf\_mgen\_absorption( m, 'S\_MED', 0 );

m = leaf\_mgen\_conductivity( m, 'S\_SOURCE', 0 );

m = leaf\_mgen\_absorption( m, 'S\_SOURCE', 0 );

m = leaf\_mgen\_conductivity( m, 'S\_JUNCTION', 0 );

m = leaf\_mgen\_absorption( m, 'S\_JUNCTION', 0 );

m = leaf\_mgen\_conductivity( m, 'S\_RIM', 0 );

m = leaf\_mgen\_absorption( m, 'S\_RIM', 0 );

m = leaf\_mgen\_conductivity( m, 'S\_SINK', 0 );

m = leaf\_mgen\_absorption( m, 'S\_SINK', 0 );

m = leaf\_mgen\_conductivity( m, 'S\_SECVEIN', 0 );

m = leaf\_mgen\_absorption( m, 'S\_SECVEIN', 0 );

m = leaf\_mgen\_conductivity( m, 'S\_DIV', 0 );

m = leaf\_mgen\_absorption( m, 'S\_DIV', 0 );

m = leaf\_mgen\_conductivity( m, 'S\_FOCI', 0);

m = leaf\_mgen\_absorption( m, 'S\_FOCI', 0);

m = leaf\_mgen\_conductivity( m, 'S\_DORSALEDGE', 0 );

m = leaf\_mgen\_absorption( m, 'S\_DORSALEDGE', 0 );

id\_twinpeaks\_p (:) = 0;

id\_twinpeaks\_p = 1- abs (s\_div\_p -0.22);

% s\_med\_p = s\_med\_p .\* inh (2, s\_rad\_p);

%add clones

m = leaf\_makesecondlayer( m, ... % This function adds biological cells.

'mode', 'each', ... % Make biological celrea was 1/16000 of the initial area of the flower.

'probpervx', 'V\_FLOWER', ... % induce tranls randomly scattered over the flower.

'relarea', 1/900, ... % Each cell has asposed cells over whole corolla

'numcells',700,...%number of cells (that will become clones)

'sides', 8, ... % Each cell is approximated as a 6-sided regular polygon.

'colors', [0.5 0.5 0.5], ... % Default colour is gray but

'allowoverlap', false, ...

'colorvariation',1,... % Each cell is a random colour

'add', true ); % These cells are added to any cells existing already

m = leaf\_plotoptions( m, 'cellbodyvalue', '' );

end

if(Steps(m)==7)

switch modelname

case {'MODEL7\_WT', 'MODEL7\_WT\_2', 'MODEL7\_WT\_3'}

%CHEEKS position by the gradients of RIM and SECVEIN and specific to the lipcliff while boosted at a certain interval of VENT (or DIV) expression and fully excluded from the hinge

id\_cheeks\_p(:) = 0;

id\_cheeks\_p = 5\* (id\_twinpeaks\_p .\* id\_lipcliff\_p .\* s\_secvein\_p.\* inh (100, id\_hinge\_p));

%STRIPY position by the gradients of ventral petal boosted by the midvein of the ventral petal while excluded from the rim

id\_stripy\_p(:) = 0;

id\_stripy\_p = pro (5, s\_med\_p).\* inh (0.5, s\_rim\_p) .\* pro (10, s\_div\_p);

end

end

if (Steps(m)>7)

% Code for specific models.

switch modelname

case 'MODEL7\_div' %Adjusting the growth rates to the system use for the Antirrhinum model

% kbpar\_p(:) = 0.1; %.\* pro(1, id\_rim\_p .\* inh (5, s\_rad\_p));

% kbper\_p(:) = 0.1;

% kapar\_p(:) = kbpar\_p...

% .\* inh (2, id\_rim\_p.\* inh (10, s\_rad\_p));

% kaper\_p(:) = kbper\_p;

% knor\_p(:) = 0.044; % @@ Eqn xx

% v\_speckareal\_p = (kapar\_p +kaper\_p +kbpar\_p +kbper\_p)./2;

% v\_specaniso\_p = kapar\_p - kaper\_p;

% v\_diff\_p = kapar\_p./kbpar\_p;

%

%

kbpar\_p(:) = 0.1...

.\* pro (0.8, id\_junction\_p.\*s\_rim\_p.\* inh (2, id\_rad\_p))...

.\* pro (0.7, id\_lip\_p.\* inh (10, s\_med\_p))...

.\* inh (0, id\_rim\_p .\* pro (10, s\_rad\_p))...

.\* inh (0, id\_palate\_p)...

.\* inh (5, s\_rad\_p .\* id\_lip\_p .\*pro (5, id\_rad\_p));

kbper\_p(:) = 0.1 ...

.\* pro (1.5, id\_rim\_p .\* (s\_junction\_p >0.11111) .\* inh (0.5, id\_junction\_p) .\* inh (10, s\_rad\_p .\* pro (2, id\_rad\_p)))...% .\* pro (0.5, id\_rim\_p .\* inh (3, id\_junction\_p) )...

.\* inh (10, s\_med\_p);

kapar\_p(:) = kbpar\_p...

.\* inh (5, id\_rim\_p);

kaper\_p(:) = kbper\_p;

knor\_p(:) = 0.044;

v\_speckareal\_p = (kapar\_p +kaper\_p +kbpar\_p +kbper\_p)./2;

v\_specaniso\_p = kapar\_p - kaper\_p;

v\_kaniso\_p = kapar\_p./(kapar\_p + kaper\_p);

case 'MODEL7\_div\_1' %Adjusting the growth rates to the system use for the Antirrhinum model

%

% kbpar\_p(:) = 0.1.\* pro(1, id\_rim\_p .\* inh (5, s\_rad\_p));

% kbper\_p(:) = 0.1;

% kapar\_p(:) = kbpar\_p...

% .\* inh (2, id\_rim\_p.\* inh (10, s\_rad\_p));

% kaper\_p(:) = kbper\_p;

% knor\_p(:) = 0.044; % @@ Eqn xx

% v\_speckareal\_p = (kapar\_p +kaper\_p +kbpar\_p +kbper\_p)./2;

% v\_specaniso\_p = kapar\_p - kaper\_p;

% v\_diff\_p = kapar\_p./kbpar\_p;

%

kbpar\_p(:) = 0.15...

.\* pro (0, s\_div\_p)...

.\* inh (10, s\_rad\_p);

kbper\_p(:) = 0.07...

.\* inh (1, s\_rad\_p);

kapar\_p(:) = kbpar\_p;

kaper\_p(:) = kbper\_p;

knor\_p(:) = 0.044; % @@ Eqn xx

v\_speckareal\_p = (kapar\_p +kaper\_p +kbpar\_p +kbper\_p)./2;

v\_specaniso\_p = kapar\_p - kaper\_p;

v\_diff\_p = kapar\_p./kbpar\_p;

if (Steps(m)>11)

kbpar\_p(:) = 0.12...

.\* pro (0.3, id\_junction\_p.\*s\_rim\_p.\* inh (0, id\_rad\_p) .\* pro (0, id\_rim\_p))...

.\* pro (1.3, id\_lip\_p)...

.\* inh (1, id\_rim\_p .\* pro (0, s\_rad\_p))...

.\* inh (10, (s\_rad\_p.^3) .\* id\_lip\_p .\* pro (4, id\_rad\_p).\* pro (1, id\_hinge\_p .\* inh (10, s\_secvein\_p)));

kbper\_p(:) = 0.06 ...

.\* pro (25, s\_rim\_p .\*s\_junction\_p.\* pro (0.5, id\_lip\_p.\*s\_rad\_p) .\* inh (2, id\_junction\_p))...

.\* inh (10, s\_med\_p)...

.\* pro (0, s\_rad\_p .\* id\_lip\_p)... .% .\* pro (0.5, id\_rim\_p .\* inh (3, id\_junction\_p) )...

.\* inh (2, id\_rad\_p);

kapar\_p(:) = kbpar\_p...

.\* inh (0, id\_rim\_p.\* inh (10, s\_rad\_p));

kaper\_p(:) = 0.06 ...

.\* pro (15, s\_rim\_p .\*s\_junction\_p.\* pro (1, id\_lip\_p.\*s\_rad\_p) .\* inh (0, id\_rad\_p))...

.\* inh (10, s\_med\_p)...

.\* pro (0, s\_rad\_p .\* id\_lip\_p)... .% .\* pro (0.5, id\_rim\_p .\* inh (3, id\_junction\_p) )...

.\* inh (2, id\_rad\_p);

knor\_p(:) = 0.044;

v\_speckareal\_p = (kapar\_p +kaper\_p +kbpar\_p +kbper\_p)./2;

v\_specaniso\_p = kapar\_p - kaper\_p;

v\_kaniso\_p = kapar\_p./(kapar\_p + kaper\_p);

end

if (Steps(m)>23)

kbpar\_p(:) = 0.1...

.\* pro (0.8, id\_junction\_p.\*s\_rim\_p.\* inh (0, id\_rad\_p) .\* pro (0.5, id\_rim\_p))...

.\* pro (0.8, id\_lip\_p.\* inh (0, s\_med\_p))...

.\* inh (1, id\_rim\_p)...

.\* inh (10, (s\_rad\_p.^3) .\* id\_lip\_p .\* pro (100, id\_rad\_p));

kbper\_p(:) = 0.08 ...

.\* pro (10, s\_rim\_p .\*s\_junction\_p .\* inh (0, id\_junction\_p) .\* inh (0, s\_rad\_p .\* pro (2, id\_rad\_p)))...

.\* inh (5, s\_med\_p)...

.\* pro (0, s\_rad\_p .\* id\_lip\_p)... .% .\* pro (0.5, id\_rim\_p .\* inh (3, id\_junction\_p) )...

.\* inh (5, s\_rad\_p .\* pro (2, id\_rad\_p));

kapar\_p(:) = kbpar\_p...

.\* inh (0, id\_rim\_p.\* inh (10, s\_rad\_p));

kaper\_p(:) = 0.1 ...

.\* pro (5, s\_rim\_p .\*s\_junction\_p .\* inh (0, id\_junction\_p) .\* inh (0, s\_rad\_p .\* pro (2, id\_rad\_p)))...

.\* inh (5, s\_med\_p)...

.\* pro (0, s\_rad\_p .\* id\_lip\_p)... .% .\* pro (0.5, id\_rim\_p .\* inh (3, id\_junction\_p) )...

.\* inh (5, s\_rad\_p .\* pro (2, id\_rad\_p));

knor\_p(:) = 0.044;

v\_speckareal\_p = (kapar\_p +kaper\_p +kbpar\_p +kbper\_p)./2;

v\_specaniso\_p = kapar\_p - kaper\_p;

v\_kaniso\_p = kapar\_p./(kapar\_p + kaper\_p);

end

case 'MODEL7\_WT' %Adjusting the growth rates to the system use for the Antirrhinum model

% kbpar\_p(:) = 0.1; %.\* pro(1, id\_rim\_p .\* inh (5, s\_rad\_p));

% kbper\_p(:) = 0.1;

% kapar\_p(:) = kbpar\_p...

% .\* inh (2, id\_rim\_p.\* inh (10, s\_rad\_p));

% kaper\_p(:) = kbper\_p;

% knor\_p(:) = 0.044; % @@ Eqn xx

% v\_speckareal\_p = (kapar\_p +kaper\_p +kbpar\_p +kbper\_p)./2;

% v\_specaniso\_p = kapar\_p - kaper\_p;

% v\_diff\_p = kapar\_p./kbpar\_p;

%

%

kbpar\_p(:) = 0.1...

.\* pro (0, id\_junction\_p)...% .\* s\_rim\_p .\* inh (1, id\_rim\_p))...%.\* pro (0.2, id\_junction\_p .\* pro (4, id\_rim\_p))...

.\* pro (0.4, id\_lip\_p .\* pro (0, id\_lipdistal\_p))...

.\* inh (1.5, id\_lipcliff\_p .\* id\_junction\_p)...

.\* inh (3, s\_rad\_p .\*pro (5, id\_rad\_p .\* id\_lip\_p))...

.\* inh (0.5, id\_med\_p .\* id\_div\_p .\* pro (1, id\_lip\_p))...

.\* pro (0.8, s\_div\_p .\* id\_palate\_p .\* pro (5, s\_rim\_p).\* pro (10, s\_junction\_p .\* pro (0.6, id\_junction\_p)).\* inh (5, s\_rad\_p));

% .\* pro (0.5, id\_stripy\_p .\* id\_lipdistal\_p)...

% .\* pro (0.5, id\_stripy\_p .\* id\_lipcliff\_p .\*inh(0, s\_med\_p))...

% .\* inh (0.5, id\_lipbend\_p)...

% .\* inh (0.5, s\_med\_p>0.09)...

% .\* inh (1, id\_rim\_p)...

% .\* inh (0, s\_sink\_p)...

% .\* inh (0, s\_source\_p ) + (0\*id\_lipcliff\_p); % @@ Eqn xx

kbper\_p(:) = 0.1 ...

.\* pro (0.8, id\_rim\_p .\* (s\_junction\_p >0.08) .\* inh (0.5, id\_junction\_p))...% .\* pro (0.5, id\_rim\_p .\* inh (3, id\_junction\_p) )...

.\* inh (0.8, id\_div\_p .\* pro (1, id\_rim\_p).\* pro (5, s\_med\_p))...

.\* inh (0.5, id\_lipbend\_p .\* pro (1, id\_div\_p))...

.\* pro (0.9, id\_cheeks\_p .\* pro (1.2, id\_div\_p))...

.\* inh (5, s\_rad\_p .\* pro (2, id\_rad\_p));

% .\* inh (1.5, id\_lipbend\_p)...

% .\* inh (0.5, s\_med\_p>0.09) ...

% .\* pro (0.7, id\_lipcliff\_p .\* id\_div\_p .\* inh (1, s\_ventmidvein\_p).\* inh (2, id\_junction\_p) .\*inh (5, s\_rim\_p))...

% .\* inh (5, s\_ventmidvein\_p) ...

% .\* inh (0, s\_med\_p)...

% .\* inh (0.5, id\_div\_p) + 0.05\* id\_lipdistal\_p + s\_source\_p .\* pro (1, id\_latrad\_p);

kapar\_p(:) = kbpar\_p...

.\* inh (5, id\_rim\_p);%.\* pro (0.2, id\_lipdistal\_p.\* inh (2, id\_div\_p));

kaper\_p(:) = kbper\_p; % @@ Eqn xx

knor\_p(:) = 0.044; % @@ Eqn xx

v\_speckareal\_p = (kapar\_p +kaper\_p +kbpar\_p +kbper\_p)./2;

v\_specaniso\_p = kapar\_p - kaper\_p;

v\_kaniso\_p = kapar\_p./(kapar\_p + kaper\_p);

case 'MODEL7\_WT\_2' %Adjusting the growth rates to the system use for the Antirrhinum model

% kbpar\_p(:) = 0.1; %.\* pro(1, id\_rim\_p .\* inh (5, s\_rad\_p));

% kbper\_p(:) = 0.1;

% kapar\_p(:) = kbpar\_p...

% .\* inh (2, id\_rim\_p.\* inh (10, s\_rad\_p));

% kaper\_p(:) = kbper\_p;

% knor\_p(:) = 0.044; % @@ Eqn xx

% v\_speckareal\_p = (kapar\_p +kaper\_p +kbpar\_p +kbper\_p)./2;

% v\_specaniso\_p = kapar\_p - kaper\_p;

% v\_diff\_p = kapar\_p./kbpar\_p;

%

%

kbpar\_p(:) = 0.1...

.\* pro (0, id\_junction\_p)...% .\* s\_rim\_p .\* inh (1, id\_rim\_p))...%.\* pro (0.2, id\_junction\_p .\* pro (4, id\_rim\_p))...

.\* pro (0.3, id\_lip\_p .\* pro (0, id\_lipdistal\_p))...

.\* inh (2, id\_lipcliff\_p .\* id\_junction\_p)...

.\* inh (3, s\_rad\_p .\*pro (5, id\_rad\_p .\* id\_lip\_p))...

.\* inh (0.5, id\_med\_p .\* id\_div\_p .\* pro (1, id\_lip\_p))...

.\* pro (2, s\_div\_p .\* id\_palate\_p .\* inh (3, s\_rim\_p).\* pro (10, s\_junction\_p .\* pro (0.6, id\_junction\_p)).\* inh (5, s\_rad\_p));

% .\* pro (0.5, id\_stripy\_p .\* id\_lipdistal\_p)...

% .\* pro (0.5, id\_stripy\_p .\* id\_lipcliff\_p .\*inh(0, s\_med\_p))...

% .\* inh (0.5, id\_lipbend\_p)...

% .\* inh (0.5, s\_med\_p>0.09)...

% .\* inh (1, id\_rim\_p)...

% .\* inh (0, s\_sink\_p)...

% .\* inh (0, s\_source\_p ) + (0\*id\_lipcliff\_p); % @@ Eqn xx

kbper\_p(:) = 0.1 ...

.\* pro (0.8, id\_rim\_p .\* (s\_junction\_p >0.08) .\* inh (0.5, id\_junction\_p))...% .\* pro (0.5, id\_rim\_p .\* inh (3, id\_junction\_p) )...

.\* inh (0.8, id\_div\_p .\* pro (1, id\_rim\_p).\* pro (5, s\_med\_p))...

.\* inh (0, id\_lipbend\_p .\* pro (1, id\_div\_p))...

.\* pro (0.9, id\_cheeks\_p .\* pro (1.5, id\_div\_p))...

.\* inh (5, s\_rad\_p .\* pro (2, id\_rad\_p));

% .\* inh (1.5, id\_lipbend\_p)...

% .\* inh (0.5, s\_med\_p>0.09) ...

% .\* pro (0.7, id\_lipcliff\_p .\* id\_div\_p .\* inh (1, s\_ventmidvein\_p).\* inh (2, id\_junction\_p) .\*inh (5, s\_rim\_p))...

% .\* inh (5, s\_ventmidvein\_p) ...

% .\* inh (0, s\_med\_p)...

% .\* inh (0.5, id\_div\_p) + 0.05\* id\_lipdistal\_p + s\_source\_p .\* pro (1, id\_latrad\_p);

kapar\_p(:) = kbpar\_p...

.\* inh (5, id\_rim\_p);%.\* pro (0.2, id\_lipdistal\_p.\* inh (2, id\_div\_p));

kaper\_p(:) = kbper\_p; % @@ Eqn xx

knor\_p(:) = 0.044; % @@ Eqn xx

v\_speckareal\_p = (kapar\_p +kaper\_p +kbpar\_p +kbper\_p)./2;

v\_specaniso\_p = kapar\_p - kaper\_p;

v\_kaniso\_p = kapar\_p./(kapar\_p + kaper\_p);

case 'MODEL7\_WT\_3' %Adjusting the growth rates to the system use for the Antirrhinum model

% kbpar\_p(:) = 0.1; %.\* pro(1, id\_rim\_p .\* inh (5, s\_rad\_p));

% kbper\_p(:) = 0.1;

% kapar\_p(:) = kbpar\_p...

% .\* inh (2, id\_rim\_p.\* inh (10, s\_rad\_p));

% kaper\_p(:) = kbper\_p;

% knor\_p(:) = 0.044; % @@ Eqn xx

% v\_speckareal\_p = (kapar\_p +kaper\_p +kbpar\_p +kbper\_p)./2;

% v\_specaniso\_p = kapar\_p - kaper\_p;

% v\_diff\_p = kapar\_p./kbpar\_p;

%

%

kbpar\_p(:) = 0.1...

.\* pro (0, id\_junction\_p)...% .\* s\_rim\_p .\* inh (1, id\_rim\_p))...%.\* pro (0.2, id\_junction\_p .\* pro (4, id\_rim\_p))...

.\* pro (0.3, id\_lip\_p .\* pro (0, id\_lipdistal\_p))...

.\* inh (2, id\_lipcliff\_p .\* id\_junction\_p)...

.\* inh (3, s\_rad\_p .\*pro (5, id\_rad\_p .\* id\_lip\_p))...

.\* inh (0.5, id\_med\_p .\* id\_div\_p .\* pro (1, id\_lip\_p))...

.\* pro (2, s\_div\_p .\* id\_palate\_p .\* inh (3, s\_rim\_p).\* pro (10, s\_junction\_p .\* pro (0.6, id\_junction\_p)).\* inh (5, s\_rad\_p));

% .\* pro (0.5, id\_stripy\_p .\* id\_lipdistal\_p)...

% .\* pro (0.5, id\_stripy\_p .\* id\_lipcliff\_p .\*inh(0, s\_med\_p))...

% .\* inh (0.5, id\_lipbend\_p)...

% .\* inh (0.5, s\_med\_p>0.09)...

% .\* inh (1, id\_rim\_p)...

% .\* inh (0, s\_sink\_p)...

% .\* inh (0, s\_source\_p ) + (0\*id\_lipcliff\_p); % @@ Eqn xx

kbper\_p(:) = 0.1 ...

.\* pro (0.8, id\_rim\_p .\* (s\_junction\_p >0.08) .\* inh (0.5, id\_junction\_p))...% .\* pro (0.5, id\_rim\_p .\* inh (3, id\_junction\_p) )...

.\* inh (0.8, id\_div\_p .\* pro (1, id\_rim\_p).\* pro (5, s\_med\_p))...

.\* inh (0, id\_lipbend\_p .\* pro (1, id\_div\_p))...

.\* inh (5, s\_rad\_p .\* pro (2, id\_rad\_p));

% .\* inh (1.5, id\_lipbend\_p)...

% .\* inh (0.5, s\_med\_p>0.09) ...

% .\* pro (0.7, id\_lipcliff\_p .\* id\_div\_p .\* inh (1, s\_ventmidvein\_p).\* inh (2, id\_junction\_p) .\*inh (5, s\_rim\_p))...

% .\* inh (5, s\_ventmidvein\_p) ...

% .\* inh (0, s\_med\_p)...

% .\* inh (0.5, id\_div\_p) + 0.05\* id\_lipdistal\_p + s\_source\_p .\* pro (1, id\_latrad\_p);

kapar\_p(:) = kbpar\_p...

.\* inh (5, id\_rim\_p);%.\* pro (0.2, id\_lipdistal\_p.\* inh (2, id\_div\_p));

kaper\_p(:) = kbper\_p; % @@ Eqn xx

knor\_p(:) = 0.044; % @@ Eqn xx

v\_speckareal\_p = (kapar\_p +kaper\_p +kbpar\_p +kbper\_p)./2;

v\_specaniso\_p = kapar\_p - kaper\_p;

v\_kaniso\_p = kapar\_p./(kapar\_p + kaper\_p);

if (Steps(m)>14)

kbpar\_p(:) = 0.1...

.\* pro (0, id\_junction\_p)...% .\* s\_rim\_p .\* inh (1, id\_rim\_p))...%.\* pro (0.2, id\_junction\_p .\* pro (4, id\_rim\_p))...

.\* pro (0.3, id\_lip\_p .\* pro (0, id\_lipdistal\_p))...

.\* inh (2, id\_lipcliff\_p .\* id\_junction\_p)...

.\* inh (3, s\_rad\_p .\*pro (5, id\_rad\_p .\* id\_lip\_p))...

.\* inh (0.5, id\_med\_p .\* id\_div\_p .\* pro (1, id\_lip\_p))...

.\* pro (2, s\_div\_p .\* id\_palate\_p .\* inh (3, s\_rim\_p).\* pro (10, s\_junction\_p .\* pro (0.6, id\_junction\_p)).\* inh (5, s\_rad\_p));

% .\* pro (0.5, id\_stripy\_p .\* id\_lipdistal\_p)...

% .\* pro (0.5, id\_stripy\_p .\* id\_lipcliff\_p .\*inh(0, s\_med\_p))...

% .\* inh (0.5, id\_lipbend\_p)...

% .\* inh (0.5, s\_med\_p>0.09)...

% .\* inh (1, id\_rim\_p)...

% .\* inh (0, s\_sink\_p)...

% .\* inh (0, s\_source\_p ) + (0\*id\_lipcliff\_p); % @@ Eqn xx

kbper\_p(:) = 0.1 ...

.\* pro (0.8, id\_rim\_p .\* (s\_junction\_p >0.08) .\* inh (0.5, id\_junction\_p))...% .\* pro (0.5, id\_rim\_p .\* inh (3, id\_junction\_p) )...

.\* inh (0.8, id\_div\_p .\* pro (1, id\_rim\_p).\* pro (5, s\_med\_p))...

.\* inh (0, id\_lipbend\_p .\* pro (1, id\_div\_p))...

.\* pro (1.2, id\_cheeks\_p .\* pro (1.5, id\_div\_p))...

.\* inh (5, s\_rad\_p .\* pro (2, id\_rad\_p));

% .\* inh (1.5, id\_lipbend\_p)...

% .\* inh (0.5, s\_med\_p>0.09) ...

% .\* pro (0.7, id\_lipcliff\_p .\* id\_div\_p .\* inh (1, s\_ventmidvein\_p).\* inh (2, id\_junction\_p) .\*inh (5, s\_rim\_p))...

% .\* inh (5, s\_ventmidvein\_p) ...

% .\* inh (0, s\_med\_p)...

% .\* inh (0.5, id\_div\_p) + 0.05\* id\_lipdistal\_p + s\_source\_p .\* pro (1, id\_latrad\_p);

end

otherwise

% If this happens, maybe you forgot a model.

end

end

%%% 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\_sink\_i) = id\_sink\_p;

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

%%% USER CODE: FINALISATION

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

% If needed force FE to subdivide (increase number FE's) here

if (Steps(m)==1)

m = leaf\_subdivide( m, 'morphogen','id\_subdivision',...

'min',0.5,'max',1,...

'mode','mid','levels','all');

end

% if (Steps(m)==4)

% m = leaf\_subdivide( m, 'morphogen','id\_lipcliff',...

% 'min',0.5,'max',1,...

% 'mode','mid','levels','all');

% end

% Cut the mesh along the seams (see above)

% if m.userdata.CutOpen==1

% m=leaf\_dissect(m);

% m.userdata.CutOpen=2;

% Relax accumulated stresses slowly i.e. 0.95 to 0.999

% m = leaf\_setproperty( m, 'freezing', 0.999 );

% end

%%% 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', 0.200000 );

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

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

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

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

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

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

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

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

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

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

% m = leaf\_setproperty( m, 'thresholdsq', 0.105169 );

% m = leaf\_setproperty( m, 'splitmargin', 1.400000 );

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

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

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

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

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

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

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

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

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

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

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

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

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

% 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', true );

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

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

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

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

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

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

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

% m = leaf\_setproperty( m, 'maxFEcells', 0 );

% m = leaf\_setproperty( m, 'inittotalcells', 0 );

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

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

% m = leaf\_setproperty( m, 'maxBioAcells', 0 );

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

% m = leaf\_setproperty( m, 'biosplitarrestmgen', 'ARREST' );

% m = leaf\_setproperty( m, 'biosplitarrestmgenthreshold', 0.990000 );

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

% m = leaf\_setproperty( m, 'colorvariation', 0.050000 );

% m = leaf\_setproperty( m, 'colorparams', (12 values) );

% m = leaf\_setproperty( m, 'biocolormode', 'auto' );

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

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

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

% m = leaf\_setproperty( m, 'mgen\_interactionName', 'gpt\_model11\_20140612' );

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

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

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

% 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, 'solverprecision', 'double' );

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

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

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

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

% m = leaf\_setproperty( m, 'maxIters', 0 );

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

% m = leaf\_setproperty( m, 'cgiters', 0 );

% m = leaf\_setproperty( m, 'simsteps', 0 );

% m = leaf\_setproperty( m, 'stepsperrender', 0 );

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

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

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

% 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', [] );

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

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

% m = leaf\_setproperty( m, 'initialArea', 4.000000 );

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

% 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:\Users\Xana\Desktop\Modelling for Palate paper' );

% m = leaf\_setproperty( m, 'modelname', 'GPT\_model11\_20140612' );

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

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

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

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

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

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

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

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

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

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

% m = leaf\_setproperty( m, 'coderevision', 5038.000000 );

% m = leaf\_setproperty( m, 'coderevisiondate', '2014-06-09 12:52:56' );

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

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

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

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

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

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

% m = leaf\_setproperty( m, 'mincellrelarea', 0.040000 );

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

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

end

% Here you may write any functions of your own, that you want to call from

% the interaction function, but never need to call from outside it.

% Remember that they do not have access to any variables except those

% that you pass as parameters, and cannot change anything except by

% returning new values as results.

% Whichever section they are called from, they must respect the same

% restrictions on what modifications they are allowed to make to the mesh.

% For example:

% function m = do\_something( m )

% % Change m in some way.

% end

% Call it from the main body of the interaction function like this:

% m = do\_something( m );