**1…function sol = bvp5c(ode, bc, solinit, options)**

% Check input arguments

% Adjust status of warnings

% Validate arguments and options

[neqn,nparam,nregions,atol,rtol,Nmax,xyVectorized,printstats] = ...

bvparguments(solver\_name,ode,bc,solinit,options);

% Modify equations to accommodate unknown parameters

% Deal with a singular BVP.

[ode,bc,jac,bcjac,Joptions,dBCoptions] = ...

bvpfunctions(solver\_name,ode,bc,options,neqn,nparam,nregions);

**141 bvpfunctions(solver\_name,ode,bc,options,neqn,nparam,nregions);**

% Deal with a singular BVP.

[singularBVP,ode,jac,solinit,PBC] = ...

bvpsingular(solver\_name,solinit,ode,jac,options,neqn,nparam,nregions);

151 % Adjust the problem to accommodate unknown parameters

164 % Four-stage Lobatto IIIa collocation formula (non-trivial coefficients only.)

175 % Constant matrices for the collocation Jacobian

182 % Interpolate solution at collocation points

[X,Y] = interpGuess(solinit);

188 % % **Algebraic solver parameters**

maxNewtIter = 4;

maxProbes = 4; % weak line search

needGlobalJacobian = true;

refinedMesh = false;

203 % **THE MAIN LOOP:**

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

**370 % Nested functions**

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

**function [X,Y] = interpGuess(sol)**

%INTERP\_GUESS Evaluate/interpolate the initial guess at collocation points.

396: end % interpGuess

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

function [X,Y] = interpGuess\_region(sol,sol\_xreg,sol\_yreg,region)

% In a region, evaluate/interpolate the initial guess at collocation points.

450... end % interpGuess\_region

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

**454 function F = odeFcn(X,Y)**

%ODE\_FCN Evaluate the ODE function for all points in X,Y.

467 end % odeFcn

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

**471 function F = odeFcn\_region(X,Y,region)**

% In a region, evaluate the ODE function for all points in X,Y.

486 end % odeFcn\_region

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

**490** **function [Jn,Jnc1,Jnc2,Jnp1] = odeJac\_region(x,y,f,Jpropagated,region)**

% In a region, compute the ODE Jacobian at points in a mesh interval.

528 end % odeJac\_region

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

**532 function res = bcaux(Ya,Yb)**

%BCAUX Reshape the arguments for the BC function.

537 end % bcaux

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

541 function [dBCdya,dBCdyb] = bcJac(ya,yb,bcVal)

%BC\_JAC Compute the BC Jacobian.

560 end % bcJac

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

**function [RHS,RHSbc]= colloc\_RHS(X,Y,F)**

%COLLOC\_RHS Evaluate the system of collocation equations.

% Separately return the residual in the boundary conditions.

590 end % colloc\_RHS

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

**594 function RHSode = colloc\_RHS\_region(X,Y,F)**

% In a region, evaluate the system of collocation equations.

616 end % colloc\_RHS\_region

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

**620 function [Jac,doSeparateBCs] = colloc\_Jac(X,Y,F,bcVal)**

%COLLOC\_JAC Form the global Jacobian of the collocation equations.

723 end % colloc\_Jac

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

**727 function [JacI,JacJ,JacV] = colloc\_JacODE\_region(X,Y,F,region)**

% In a region, form the Jacobian of collocation equations.

768 end % colloc\_JacODE\_region

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

**772 function maxInterpResidual = colloc\_maxresidual(X,Y,F)**

%COLLOC\_MAXRESIDUAL Compute max residual in the collocation equations.

784 end % colloc\_maxresidual

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

788 function maxInterpResidual = colloc\_maxresidual\_region(X,Y,F)

% In a region, compute max residual in the collocation equations.

819 end % colloc\_maxresidual\_region

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

**823 function ymid = interpolateYmid(X,Y,F)**

%INTERPOLATE\_YMID Interpolate Y at the midpoints of mesh subintervals.

835 end % interpolateYmid

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

**839 function ymid = interpolateYmid\_region(X,Y,F)**

% In a region, interpolate Y at the midpoints of mesh subintervals.

h = diff(X(1:nstages:end));

850 end % interpolateYmid\_region

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

**854 function res = residualEstimate(X,Y,Ymid,F,nsamples)**

%RESIDUAL\_ESTIMATE Estimate the residual in each mesh subinterval.

866 end % residualEstimate

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

**870 function res = residualEstimate\_region(X,Y,ymid,F,nsamples,region)**

% In a region, estimate the residual in each mesh subinterval.

908 end % residualEstimate\_region

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

**912 function [XX,YY,FF] = newSolutionProfile(X,Y,Ymid,F,errEst)**

%NEW\_SOLUTION\_PROFILE Redistribute mesh points and approximate the solution.

% Detect mesh oscillations: Was there a mesh with

% the same number of nodes and a similar residual?

917 **% If so, only allow for adding mesh points.**

927 % modify the mesh, interpolate the solution

[xreg,yreg,freg,ymidreg,errEstReg] = getRegionData(1,X,Y,F,Ymid,errEst);

[XX,YY,FF] = newSolutionProfile\_region(xreg,yreg,ymidreg,freg,errEstReg,...

{1},canRemovePoints);

942 end % newSolutionProfile

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

**946 function [XX,YY,FF] = newSolutionProfile\_region(X,Y,ymid,F,errEst,region,...**

canRemovePoints)% In a region, redistribute mesh points and approximate the solution.

1064 end % newSolutionProfile\_region

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

**1068 function sol = outputSol(X,Y,F,ymid)**

%OUTPUT\_SOL Assembly the solution structure.

1099 end % outputSol

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

**1103 function [xout,yout,ypout,ymidout] = outputData(x,y,f,ymid)**

%OUTPUT\_DATA Prepare data for the solution structure.

1109 end % outputData

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

**1113 function MBVP = updateMBVP(X)**

%UPDATE\_MBVP Update indices of the internal boundary points for MBVPs.

1122end % updateMBVP

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

1126 function [xreg,yreg,freg,ymidreg,errest] = getRegionData(region,X,Y,F,Ymid,ErrEst)

%GET\_REGION\_DATA Extract mesh points and solution data for a given region.

1152 end % getRegionData

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

1156 end % bvp5c