In[ ]:=
```
(* Initialisation *)
(* Evaluate before starting writing "real code" *)
(* Usage e.g.: "ld [Spacekey]" becomes "⊨",
so writing "a ld 5" turns into "a⊨5" *)
SetOptions[EvaluationNotebook[],
            InputAutoReplacements → {(* special AceGen assignment operators: *)
   "ld" → "⊨", "ls" → "⊢", "rd" → "⥤", "rs" → "⊣",
                                     (* brackets and symbols: *) "dbl" → "⟦",
   "dbr" → "⟧", "lcb" → "{", "rcb" → "}", "lsb" → "[", "rsb" → "]", "->" → "→",
                                     (* shortcuts for
    starting/ending a comment block: *) "co" → "(*", "cc" → "*)"
                                    }
          ]
(* Output the current time,
so we know when AceGen has been executed the last time *)
Now
```

Out[ ]=
```
Thu 20 Jun 2024 14:25:45 GMT+2
```

In[ ]:=
```
(* Clear all old variables initially to have a fresh start *)
ClearAll["Global`*"]
(* Start AceGen *)
<< AceGen`;
(* Name of the to be created subroutine/function
 in the below specified programming language *)
NAME = "HyperElasticity_eigenvalues ";
(* Name of the AceGen session "NAME",
specify the programming language "Language"={C++,Matlab,Fortran,...},
and the execution mode "Mode"={Optimal,Prototype,Debug,Plain} *)
(* @note Changing the output programming language can be very simple here,
so feel free to take advantage of all the available languages. For instance,
first export to a Matlab-code,
because you can quickly and easily debug and check the code and its output. *)
SMSInitialize[NAME, "Language" → "Matlab", "Mode" → "Optimal"];
(* Start a module, which represents the to be created function,
with name "NAME" and the specified input and output arguments *)
SMSModule[NAME, Real[F$$[3, 3], listOfMaterialParameters$$ [2],
    CauchyStress$$[3, 3], Tangent$$[3, 3, 3, 3]],
           "Input" → {F$$, listOfMaterialParameters$$ },
           "Output" → {CauchyStress$$ , Tangent$$ }
         ];
(* Input declaration by copying AceGen variables to Mathematica variables *)
F ⊨ SMSReal[Table[F$$[i, j], {i, 3}, {j, 3}]];
listOfMaterialParameters ⊨ SMSReal[Table[listOfMaterialParameters$$ [i], {i, 11}]];
(* Extract the number of Ogden parts Np *)
Np ⊨ SMSInteger[listOfMaterialParameters$$ [1]];
```

Strain energy density, here for example a Ogden model based on the principal stretches λ_i

$$\Psi^{\text{Ogden}} = \sum_{p=1}^{\text{Np}} \frac{\mu_p}{\alpha_p}\left[{\lambda_1}^{\alpha_p} + {\lambda_2}^{\alpha_p} + {\lambda_3}^{\alpha_p} - 3\right]$$

In[ ]:=
```
Psi_Ogden[{λ1_, λ2_, λ3_}] :=
 (
  (* A SMSDo-construct replaces the above summation ∑ as the number of
       Ogden parts Np is a parameter and not know at compile time *)
  Psi_Ogden = 0;
  SMSDo[p, 1, Np, 1, {Psi_Ogden}];
      (* As the number of Ogden parts Np is not known at compile time,
  we need to use SMSPart to correctly access the material parameters *)
      α_p = SMSPart[listOfMaterialParameters , p + 1];
      μ_p = SMSPart[listOfMaterialParameters , p + Np + 1];
      Psi_Ogden = Psi_Ogden + (μ_p/α_p) (λ1^α_p + λ2^α_p + λ3^α_p - 3);
  SMSEndDo[Psi_Ogden];

  Return[Psi_Ogden];
 );
```

For the Ogden material model the principal stretches are needed. The principal stretches are the eigenvalues of the right stretch tensor U, which is the matrix square root of the right Cauchy Green tensor C. As we below also compute the analytical Ogden Cauchy stress tensor based on the eigen-stresses and its bases, we need the spatial eigenvectors. Therefore, we compute the eigenvalues of the left Cauchy Green tensor LCG_b, which shares the same eigenvalues as C, but with the spatial eigenvectors (resulting in spatial eigenbases "eb"). Normally, we would neither need the eigenbases nor its derivatives.

```
In[ ]:= (* Compute the left Cauchy-
         Green (LCG) strain tensor LCG_b from the deformation gradient. *)
        LCG_b := F.F^T;
        (* Compute the eigenvalues, eigenbases, and derivatives of LCG_b
         (principal stretches EW_U are the square root of eigenvalues of LCG_b) *)
        subrEIG = SMSCall["eig_EW_EB_d_AG", LCG_b, 2, Real[EW$$[3],
            eb$$[3, 3, 3], dEWdb$$[3, 3, 3], debdb$$[3, 3, 3, 3, 3]], "System" → False];
        dEWdb := SMSReal[Table[dEWdb$$[iEW, i, j], {iEW, 3}, {i, 3}, {j, 3}],
           "Subordinate" → subrEIG];
        EW_b := SMSReal[Table[0, {iEW, 3}]];
        EW_U := SMSReal[Table[0, {iEW, 3}]];
        Do[{
           EW_b⟦iEW⟧ := SMSReal[EW$$[iEW], "Subordinate" → subrEIG,
             "Dependency" → Flatten[Table[{LCG_b⟦k, l⟧, dEWdb⟦iEW, k, l⟧}, {k, 3}, {l, 3}], 1]];
           EW_U[[iEW]] := SMSSqrt[EW_b⟦iEW⟧];
         }, {iEW, 3}];
        debdb := SMSReal[Table[debdb$$[iEW, i, j, k, l], {iEW, 3}, {i, 3}, {j, 3}, {k, 3}, {l, 3}],
           "Subordinate" → subrEIG];
        eb := SMSReal[Table[0, {iEW, 3}, {i, 3}, {j, 3}]];
        Do[{
           Do[{
              Do[{
                 eb⟦iEW, i, j⟧ := SMSReal[eb$$[iEW, i, j], "Subordinate" → subrEIG, "Dependency" →
                    Flatten[Table[{LCG_b⟦k, l⟧, debdb⟦iEW, i, j, k, l⟧}, {k, 3}, {l, 3}], 1]]
               }, {j, 3}]
            }, {i, 3}]
         }, {iEW, 3}];

In[ ]:= (* Compute the first Piola-
         Kirchhoff stress tensor PK1_P derived from the energy *)
        PK1_P := SMSD[Psi_Ogden[EW_U], F];
        (* Push-forward PK1_P to Cauchy stress Cauchy_sigma *)
        Cauchy_sigma := 1/SMSDet[F] PK1_P . F^T;
        (* Export the output variables by copying
         the Mathematica variables to AceGen variables *)
        (* Output/Export the stress vector Cauchy_sigma as variable CauchyStress$$ *)
        SMSExport[Cauchy_sigma, Table[CauchyStress$$[i, j], {i, 3}, {j, 3}]];

In[ ]:= (* Compute the dP/dF elasticity tangent *)
        DPK1_DF := SMSD[PK1_P, F];
        (* Output/Export the derivative as a fourth-order tensor *)
        SMSExport[DPK1_DF, Table[Tangent$$[i, j, k, l], {i, 3}, {j, 3}, {k, 3}, {l, 3}]];
```

```
In[ ]:= (* Debugging/Verification *)
(* Output the stress tensor to the screen *)
SMSPrintMessage[NAME <> "<< Cauchy stress AceGen= ", Cauchy_sigma];

(* Incompressible neo-Hookean material equal to Ogden with (Np=1, μ=μ1):
  SMSPrintMessage[
   NAME<>"<< incompr. NH: Cauchy stress analytical=", μ[[1]]/SMSDet[F] F.F^T]; *)

(* Analytical Ogden Cauchy stress based on principal Cauchy stresses
 by [https://public.websites.umich.edu/~bme332/ch6consteqelasticity/
     bme332consteqelasticity.htm#ogdenfunction] *)
Cauchy_sigma_ay = SMSReal[Table[0, {i, 3}, {j, 3}]];
SMSDo[p, 1, Np, 1, {Cauchy_sigma_ay}];
    α_p = SMSPart[listOfMaterialParameters, p + 1];
    μ_p = SMSPart[listOfMaterialParameters, p + Np + 1];
    Cauchy_sigma_ay = Cauchy_sigma_ay +
   1/SMSDet[F] Table[Sum[EW_U[[iEW]] μ_p EW_U[[iEW]]^(α_p - 1) eb[[iEW, i, j]], {iEW, 3}], {i, 3}, {j, 3}];
SMSEndDo[Cauchy_sigma_ay];

(* SMSPrintMessage[NAME<>"<< Cauchy stress analytical=",Cauchy_sigma_ay]; *)
SMSPrintMessage[NAME <> "<< Cauchy stress relative error=",
  1/Cauchy_sigma[[1, 1]] Sum[(Cauchy_sigma - Cauchy_sigma_ay)[[i, j]]^2, {i, 3}, {j, 3}]];
```

In[ ]:=
```
(* Output the time at the end of the execution *)
Now
(* Write output file containing all the
 above defined functions introduced by SMSModule *)
(* Create output file named "NAME", '"LocalAuxiliaryVariables " →
 True' is a command to exclude the AceGen internal array "v" from
  the list of input and output arguments of the created subroutine *)
(* @note The specification of WorkingVectorSize was requested
  by AceGen and is done as recommended for SMSWrite→Help *)
SMSWrite[NAME, "LocalAuxiliaryVariables " → True,
  "WorkingVectorSize " → Function[{lengths}, Max[lengths /. {"Np" → 5}]]];
(* Print the content of the just created file on screen
 (sensible only for small file sizes) *)
FilePrint[StringJoin[NAME, Which[SMSLanguage == "Fortran", ".f",
                                 SMSLanguage == "Matlab", ".m",
                                 SMSLanguage == "C++", ".cpp",
                                 SMSLanguage == "C", ".c"
                                ]
                     ]
          ]
```

Out[ ]= Thu 20 Jun 2024 14:25:52 GMT+2

```
Method :     HyperElasticity_eigenvalues       Working length must be more than:     1191 + 2 * '
```

File: HyperElasticity_eigenvalues .m **Size:** 17920 **Time:** 12

Method	HyperElasticity_eigenvalues
No.Formulae	322
No.Leafs	7818

```
%************************************************************
%* AceGen     7.505 Linux (16 Aug 22)                       *
%*            Co. J. Korelc  2020           20 Jun 24 14:25:59  *
%************************************************************
% User      : Full professional version
% Notebook  : AceGen-HyperElasticity-eigenvalues
% Evaluation time                 : 12 s     Mode   : Optimal
% Number of formulae              : 322      Method: Automatic
% Subroutine                      : HyperElasticity_eigenvalues   size: 7818
% Total size of Mathematica  code : 7818 subexpressions
% Total size of Matlab code       : 16984 bytes

%**********************   F U N C T I O N ************************
function[CauchyStress,Tangent]=HyperElasticity_eigenvalues(F,listOfMaterialParameters);
persistent v;
if size(v)<1201
  v=zeros(1201,'double');
end;
i0=32;
```

```
v(1)=F(1,1);
v(395)=2e0*v(1);
v(2)=F(1,2);
v(396)=2e0*v(2);
v(3)=F(1,3);
v(397)=2e0*v(3);
v(4)=F(2,1);
v(398)=2e0*v(4);
v(5)=F(2,2);
v(399)=2e0*v(5);
v(6)=F(2,3);
v(400)=2e0*v(6);
v(7)=F(3,1);
v(401)=2e0*v(7);
v(8)=F(3,2);
v(402)=2e0*v(8);
v(9)=F(3,3);
v(403)=2e0*v(9);
v(385)=1/(v(3)*(-(v(5)*v(7))+v(4)*v(8))+v(2)*(v(6)*v(7)-v(4)*v(9))+v(1)*(-(v(6)*v(8))+v(5)*v(9)));
v(1067)=listOfMaterialParameters(1);
v(1068)=listOfMaterialParameters(2);
v(1069)=listOfMaterialParameters(3);
v(1070)=listOfMaterialParameters(4);
v(1071)=listOfMaterialParameters(5);
v(1072)=listOfMaterialParameters(6);
v(1073)=listOfMaterialParameters(7);
v(1074)=listOfMaterialParameters(8);
v(1075)=listOfMaterialParameters(9);
v(1076)=listOfMaterialParameters(10);
v(1077)=listOfMaterialParameters(11);
i21=listOfMaterialParameters(1);
v(23)=v(1)*v(4)+v(2)*v(5)+v(3)*v(6);
v(24)=v(1)*v(7)+v(2)*v(8)+v(3)*v(9);
v(26)=v(4)*v(7)+v(5)*v(8)+v(6)*v(9);
v01(1,1)=(v(1)*v(1))+(v(2)*v(2))+(v(3)*v(3));
v01(1,2)=v(23);
v01(1,3)=v(24);
v01(2,1)=v(23);
v01(2,2)=(v(4)*v(4))+(v(5)*v(5))+(v(6)*v(6));
v01(2,3)=v(26);
v01(3,1)=v(24);
v01(3,2)=v(26);
v01(3,3)=(v(7)*v(7))+(v(8)*v(8))+(v(9)*v(9));
i02=2e0;
eig_EW_EB_d_AG(v01,i02,EW,eb,dEWdb,debdb);
v(31)=dEWdb(1,1,1);
v(558)=dEWdb(1,1,2)+dEWdb(1,2,1);
v(35)=dEWdb(1,2,2);
v(555)=dEWdb(1,1,3)+dEWdb(1,3,1);
v(552)=dEWdb(1,2,3)+dEWdb(1,3,2);
v(39)=dEWdb(1,3,3);
v(40)=dEWdb(2,1,1);
v(557)=dEWdb(2,1,2)+dEWdb(2,2,1);
v(44)=dEWdb(2,2,2);
v(554)=dEWdb(2,1,3)+dEWdb(2,3,1);
```

```
v(551)=dEWdb(2,2,3)+dEWdb(2,3,2);
v(48)=dEWdb(2,3,3);
v(49)=dEWdb(3,1,1);
v(556)=dEWdb(3,1,2)+dEWdb(3,2,1);
v(53)=dEWdb(3,2,2);
v(553)=dEWdb(3,1,3)+dEWdb(3,3,1);
v(550)=dEWdb(3,2,3)+dEWdb(3,3,2);
v(57)=dEWdb(3,3,3);
v(413)=sqrt(EW(1));
v(823)=1/(2e0*v(413));
v(423)=(v(39)*v(403)+v(3)*v(555)+v(552)*v(6))*v(823);
v(422)=(v(39)*v(402)+v(5)*v(552)+v(2)*v(555))*v(823);
v(421)=(v(39)*v(401)+v(4)*v(552)+v(1)*v(555))*v(823);
v(420)=v(823)*(v(35)*v(400)+v(3)*v(558)+v(552)*v(9));
v(419)=(v(35)*v(399)+v(2)*v(558)+v(552)*v(8))*v(823);
v(418)=(v(35)*v(398)+v(1)*v(558)+v(552)*v(7))*v(823);
v(417)=v(823)*(v(31)*v(397)+v(558)*v(6)+v(555)*v(9));
v(416)=(v(31)*v(396)+v(5)*v(558)+v(555)*v(8))*v(823);
v(414)=(v(31)*v(395)+v(4)*v(558)+v(555)*v(7))*v(823);
v(433)=sqrt(EW(2));
v(824)=1/(2e0*v(433));
v(443)=(v(403)*v(48)+v(3)*v(554)+v(551)*v(6))*v(824);
v(442)=(v(402)*v(48)+v(5)*v(551)+v(2)*v(554))*v(824);
v(441)=(v(401)*v(48)+v(4)*v(551)+v(1)*v(554))*v(824);
v(440)=v(824)*(v(400)*v(44)+v(3)*v(557)+v(551)*v(9));
v(439)=(v(399)*v(44)+v(2)*v(557)+v(551)*v(8))*v(824);
v(438)=(v(398)*v(44)+v(1)*v(557)+v(551)*v(7))*v(824);
v(437)=v(824)*(v(397)*v(40)+v(557)*v(6)+v(554)*v(9));
v(436)=(v(396)*v(40)+v(5)*v(557)+v(554)*v(8))*v(824);
v(434)=(v(395)*v(40)+v(4)*v(557)+v(554)*v(7))*v(824);
v(453)=sqrt(EW(3));
v(825)=1/(2e0*v(453));
v(463)=(v(3)*v(553)+v(403)*v(57)+v(550)*v(6))*v(825);
v(462)=(v(5)*v(550)+v(2)*v(553)+v(402)*v(57))*v(825);
v(461)=(v(4)*v(550)+v(1)*v(553)+v(401)*v(57))*v(825);
v(460)=v(825)*(v(400)*v(53)+v(3)*v(556)+v(550)*v(9));
v(459)=(v(399)*v(53)+v(2)*v(556)+v(550)*v(8))*v(825);
v(458)=(v(398)*v(53)+v(1)*v(556)+v(550)*v(7))*v(825);
v(457)=v(825)*(v(397)*v(49)+v(556)*v(6)+v(553)*v(9));
v(456)=(v(396)*v(49)+v(5)*v(556)+v(553)*v(8))*v(825);
v(454)=(v(395)*v(49)+v(4)*v(556)+v(553)*v(7))*v(825);
v(307)=eb(1,1,1);
v(308)=eb(1,1,2);
v(309)=eb(1,1,3);
v(310)=eb(1,2,1);
v(311)=eb(1,2,2);
v(312)=eb(1,2,3);
v(313)=eb(1,3,1);
v(314)=eb(1,3,2);
v(315)=eb(1,3,3);
v(316)=eb(2,1,1);
v(317)=eb(2,1,2);
v(318)=eb(2,1,3);
v(319)=eb(2,2,1);
v(320)=eb(2,2,2);
```

```
v(321)=eb(2,2,3);
v(322)=eb(2,3,1);
v(323)=eb(2,3,2);
v(324)=eb(2,3,3);
v(325)=eb(3,1,1);
v(326)=eb(3,1,2);
v(327)=eb(3,1,3);
v(328)=eb(3,2,1);
v(329)=eb(3,2,2);
v(330)=eb(3,2,3);
v(331)=eb(3,3,1);
v(332)=eb(3,3,2);
v(333)=eb(3,3,3);
i347=i0;
i0=i0+i21;
i349=i0;
i0=i0+i21;
for i335=1:1:i21;
 i826=1+i335;
 v(1059+i335+i347)=v(1066+i826);
 v(1059+i335+i349)=v(1066+i21+i826);
end;
v(339)=1e0;
v(464)=0;
v(465)=0;
v(466)=0;
v(467)=0;
v(468)=0;
v(469)=0;
v(470)=0;
v(471)=0;
v(472)=0;
v(340)=0;
v(473)=0;
v(474)=0;
v(475)=0;
v(476)=0;
v(477)=0;
v(478)=0;
v(479)=0;
v(480)=0;
v(481)=0;
v(341)=0;
v(482)=0;
v(483)=0;
v(484)=0;
v(485)=0;
v(486)=0;
v(487)=0;
v(488)=0;
v(489)=0;
v(490)=0;
v(342)=0;
v(491)=0;
v(492)=0;
```

```
v(493)=0;
v(494)=0;
v(495)=0;
v(496)=0;
v(497)=0;
v(498)=0;
v(499)=0;
v(343)=0;
v(500)=0;
v(501)=0;
v(502)=0;
v(503)=0;
v(504)=0;
v(505)=0;
v(506)=0;
v(507)=0;
v(508)=0;
v(344)=0;
v(509)=0;
v(510)=0;
v(511)=0;
v(512)=0;
v(513)=0;
v(514)=0;
v(515)=0;
v(516)=0;
v(517)=0;
v(345)=0;
for i346=i21:-1:1;
 v(351)=v(1059+i346+i347);
 v(829)=v(351)/2e0;
 v(353)=(v(339)*v(1059+i346+i349))/v(351);
 v(352)=-2e0+v(351);
 v(529)=-1e0+v(352);
 v(528)=v(352)*v(829);
 v(828)=Power(v(433),v(529))*v(528);
 v(827)=Power(v(413),v(529))*v(528);
 v(549)=v(423)*v(827);
 v(548)=v(422)*v(827);
 v(547)=v(421)*v(827);
 v(546)=v(420)*v(827);
 v(545)=v(419)*v(827);
 v(544)=v(418)*v(827);
 v(543)=v(417)*v(827);
 v(542)=v(416)*v(827);
 v(540)=v(414)*v(827);
 v(539)=v(443)*v(828);
 v(538)=v(442)*v(828);
 v(537)=v(441)*v(828);
 v(536)=v(440)*v(828);
 v(535)=v(439)*v(828);
 v(534)=v(438)*v(828);
 v(533)=v(437)*v(828);
 v(532)=v(436)*v(828);
 v(530)=v(434)*v(828);
```

```
v(519)=Power(v(453),v(529))*v(528);
v(527)=v(463)*v(519);
v(526)=v(462)*v(519);
v(525)=v(461)*v(519);
v(524)=v(460)*v(519);
v(523)=v(459)*v(519);
v(522)=v(458)*v(519);
v(521)=v(457)*v(519);
v(520)=v(456)*v(519);
v(518)=v(454)*v(519);
v(356)=Power(v(453),v(352))*v(829);
v(355)=Power(v(433),v(352))*v(829);
v(354)=Power(v(413),v(352))*v(829);
v(509)=v(509)+v(353)*(v(48)*v(530)+v(39)*v(540)+v(518)*v(57));
v(510)=v(510)+v(353)*(v(48)*v(532)+v(39)*v(542)+v(520)*v(57));
v(511)=v(511)+v(353)*(v(48)*v(533)+v(39)*v(543)+v(521)*v(57));
v(512)=v(512)+v(353)*(v(48)*v(534)+v(39)*v(544)+v(522)*v(57));
v(513)=v(513)+v(353)*(v(48)*v(535)+v(39)*v(545)+v(523)*v(57));
v(514)=v(514)+v(353)*(v(48)*v(536)+v(39)*v(546)+v(524)*v(57));
v(515)=v(515)+v(353)*(v(48)*v(537)+v(39)*v(547)+v(525)*v(57));
v(516)=v(516)+v(353)*(v(48)*v(538)+v(39)*v(548)+v(526)*v(57));
v(517)=v(517)+v(353)*(v(48)*v(539)+v(39)*v(549)+v(527)*v(57));
v(345)=v(345)+v(353)*(v(354)*v(39)+v(355)*v(48)+v(356)*v(57));
v(500)=v(500)+v(353)*(v(518)*v(550)+v(530)*v(551)+v(540)*v(552));
v(501)=v(501)+v(353)*(v(520)*v(550)+v(532)*v(551)+v(542)*v(552));
v(502)=v(502)+v(353)*(v(521)*v(550)+v(533)*v(551)+v(543)*v(552));
v(503)=v(503)+v(353)*(v(522)*v(550)+v(534)*v(551)+v(544)*v(552));
v(504)=v(504)+v(353)*(v(523)*v(550)+v(535)*v(551)+v(545)*v(552));
v(505)=v(505)+v(353)*(v(524)*v(550)+v(536)*v(551)+v(546)*v(552));
v(506)=v(506)+v(353)*(v(525)*v(550)+v(537)*v(551)+v(547)*v(552));
v(507)=v(507)+v(353)*(v(526)*v(550)+v(538)*v(551)+v(548)*v(552));
v(508)=v(508)+v(353)*(v(527)*v(550)+v(539)*v(551)+v(549)*v(552));
v(344)=v(344)+v(353)*(v(356)*v(550)+v(355)*v(551)+v(354)*v(552));
v(491)=v(491)+v(353)*(v(518)*v(53)+v(44)*v(530)+v(35)*v(540));
v(492)=v(492)+v(353)*(v(520)*v(53)+v(44)*v(532)+v(35)*v(542));
v(493)=v(493)+v(353)*(v(521)*v(53)+v(44)*v(533)+v(35)*v(543));
v(494)=v(494)+v(353)*(v(522)*v(53)+v(44)*v(534)+v(35)*v(544));
v(495)=v(495)+v(353)*(v(523)*v(53)+v(44)*v(535)+v(35)*v(545));
v(496)=v(496)+v(353)*(v(524)*v(53)+v(44)*v(536)+v(35)*v(546));
v(497)=v(497)+v(353)*(v(525)*v(53)+v(44)*v(537)+v(35)*v(547));
v(498)=v(498)+v(353)*(v(526)*v(53)+v(44)*v(538)+v(35)*v(548));
v(499)=v(499)+v(353)*(v(527)*v(53)+v(44)*v(539)+v(35)*v(549));
v(343)=v(343)+v(353)*(v(35)*v(354)+v(355)*v(44)+v(356)*v(53));
v(482)=v(482)+v(353)*(v(518)*v(553)+v(530)*v(554)+v(540)*v(555));
v(483)=v(483)+v(353)*(v(520)*v(553)+v(532)*v(554)+v(542)*v(555));
v(484)=v(484)+v(353)*(v(521)*v(553)+v(533)*v(554)+v(543)*v(555));
v(485)=v(485)+v(353)*(v(522)*v(553)+v(534)*v(554)+v(544)*v(555));
v(486)=v(486)+v(353)*(v(523)*v(553)+v(535)*v(554)+v(545)*v(555));
v(487)=v(487)+v(353)*(v(524)*v(553)+v(536)*v(554)+v(546)*v(555));
v(488)=v(488)+v(353)*(v(525)*v(553)+v(537)*v(554)+v(547)*v(555));
v(489)=v(489)+v(353)*(v(526)*v(553)+v(538)*v(554)+v(548)*v(555));
v(490)=v(490)+v(353)*(v(527)*v(553)+v(539)*v(554)+v(549)*v(555));
v(342)=v(342)+v(353)*(v(356)*v(553)+v(355)*v(554)+v(354)*v(555));
v(473)=v(473)+v(353)*(v(518)*v(556)+v(530)*v(557)+v(540)*v(558));
v(474)=v(474)+v(353)*(v(520)*v(556)+v(532)*v(557)+v(542)*v(558));
```

```
  v(475)=v(475)+v(353)*(v(521)*v(556)+v(533)*v(557)+v(543)*v(558));
  v(476)=v(476)+v(353)*(v(522)*v(556)+v(534)*v(557)+v(544)*v(558));
  v(477)=v(477)+v(353)*(v(523)*v(556)+v(535)*v(557)+v(545)*v(558));
  v(478)=v(478)+v(353)*(v(524)*v(556)+v(536)*v(557)+v(546)*v(558));
  v(479)=v(479)+v(353)*(v(525)*v(556)+v(537)*v(557)+v(547)*v(558));
  v(480)=v(480)+v(353)*(v(526)*v(556)+v(538)*v(557)+v(548)*v(558));
  v(481)=v(481)+v(353)*(v(527)*v(556)+v(539)*v(557)+v(549)*v(558));
  v(341)=v(341)+v(353)*(v(356)*v(556)+v(355)*v(557)+v(354)*v(558));
  v(464)=v(464)+v(353)*(v(49)*v(518)+v(40)*v(530)+v(31)*v(540));
  v(465)=v(465)+v(353)*(v(49)*v(520)+v(40)*v(532)+v(31)*v(542));
  v(466)=v(466)+v(353)*(v(49)*v(521)+v(40)*v(533)+v(31)*v(543));
  v(467)=v(467)+v(353)*(v(49)*v(522)+v(40)*v(534)+v(31)*v(544));
  v(468)=v(468)+v(353)*(v(49)*v(523)+v(40)*v(535)+v(31)*v(545));
  v(469)=v(469)+v(353)*(v(49)*v(524)+v(40)*v(536)+v(31)*v(546));
  v(470)=v(470)+v(353)*(v(49)*v(525)+v(40)*v(537)+v(31)*v(547));
  v(471)=v(471)+v(353)*(v(49)*v(526)+v(40)*v(538)+v(31)*v(548));
  v(472)=v(472)+v(353)*(v(49)*v(527)+v(40)*v(539)+v(31)*v(549));
  v(340)=v(340)+v(353)*(v(31)*v(354)+v(355)*v(40)+v(356)*v(49));
end;
v(787)=2e0*v(340);
v(654)=2e0*v(343);
v(575)=2e0*v(345);
v(339)=0;
v(369)=v(3)*v(342)+v(344)*v(6)+v(575)*v(9);
v(370)=v(2)*v(342)+v(344)*v(5)+v(575)*v(8);
v(371)=v(1)*v(342)+v(344)*v(4)+v(575)*v(7);
v(375)=v(3)*v(341)+v(6)*v(654)+v(344)*v(9);
v(376)=v(2)*v(341)+v(5)*v(654)+v(344)*v(8);
v(377)=v(1)*v(341)+v(4)*v(654)+v(344)*v(7);
v(381)=v(341)*v(6)+v(3)*v(787)+v(342)*v(9);
v(382)=v(341)*v(5)+v(2)*v(787)+v(342)*v(8);
v(383)=v(341)*v(4)+v(342)*v(7)+v(1)*v(787);
v(384)=(v(3)*v(381)+v(2)*v(382)+v(1)*v(383))*v(385);
v(386)=v(385)*(v(383)*v(4)+v(382)*v(5)+v(381)*v(6));
v(387)=v(385)*(v(383)*v(7)+v(382)*v(8)+v(381)*v(9));
v(388)=(v(3)*v(375)+v(2)*v(376)+v(1)*v(377))*v(385);
v(389)=v(385)*(v(377)*v(4)+v(376)*v(5)+v(375)*v(6));
v(390)=v(385)*(v(377)*v(7)+v(376)*v(8)+v(375)*v(9));
v(391)=(v(3)*v(369)+v(2)*v(370)+v(1)*v(371))*v(385);
v(392)=v(385)*(v(371)*v(4)+v(370)*v(5)+v(369)*v(6));
v(393)=v(385)*(v(371)*v(7)+v(370)*v(8)+v(369)*v(9));
CauchyStress(1,1)=v(384);
CauchyStress(1,2)=v(386);
CauchyStress(1,3)=v(387);
CauchyStress(2,1)=v(388);
CauchyStress(2,2)=v(389);
CauchyStress(2,3)=v(390);
CauchyStress(3,1)=v(391);
CauchyStress(3,2)=v(392);
CauchyStress(3,3)=v(393);
Tangent(1,1,1,1)=v(395)*v(464)+v(4)*v(473)+v(482)*v(7)+v(787);
Tangent(1,1,1,2)=v(395)*v(465)+v(4)*v(474)+v(483)*v(7);
Tangent(1,1,1,3)=v(395)*v(466)+v(4)*v(475)+v(484)*v(7);
Tangent(1,1,2,1)=v(341)+v(395)*v(467)+v(4)*v(476)+v(485)*v(7);
Tangent(1,1,2,2)=v(395)*v(468)+v(4)*v(477)+v(486)*v(7);
```

```
Tangent(1,1,2,3)=v(395)*v(469)+v(4)*v(478)+v(487)*v(7);
Tangent(1,1,3,1)=v(342)+v(395)*v(470)+v(4)*v(479)+v(488)*v(7);
Tangent(1,1,3,2)=v(395)*v(471)+v(4)*v(480)+v(489)*v(7);
Tangent(1,1,3,3)=v(395)*v(472)+v(4)*v(481)+v(490)*v(7);
Tangent(1,2,1,1)=v(396)*v(464)+v(473)*v(5)+v(482)*v(8);
Tangent(1,2,1,2)=v(396)*v(465)+v(474)*v(5)+v(787)+v(483)*v(8);
Tangent(1,2,1,3)=v(396)*v(466)+v(475)*v(5)+v(484)*v(8);
Tangent(1,2,2,1)=v(396)*v(467)+v(476)*v(5)+v(485)*v(8);
Tangent(1,2,2,2)=v(341)+v(396)*v(468)+v(477)*v(5)+v(486)*v(8);
Tangent(1,2,2,3)=v(396)*v(469)+v(478)*v(5)+v(487)*v(8);
Tangent(1,2,3,1)=v(396)*v(470)+v(479)*v(5)+v(488)*v(8);
Tangent(1,2,3,2)=v(342)+v(396)*v(471)+v(480)*v(5)+v(489)*v(8);
Tangent(1,2,3,3)=v(396)*v(472)+v(481)*v(5)+v(490)*v(8);
Tangent(1,3,1,1)=v(397)*v(464)+v(473)*v(6)+v(482)*v(9);
Tangent(1,3,1,2)=v(397)*v(465)+v(474)*v(6)+v(483)*v(9);
Tangent(1,3,1,3)=v(397)*v(466)+v(475)*v(6)+v(787)+v(484)*v(9);
Tangent(1,3,2,1)=v(397)*v(467)+v(476)*v(6)+v(485)*v(9);
Tangent(1,3,2,2)=v(397)*v(468)+v(477)*v(6)+v(486)*v(9);
Tangent(1,3,2,3)=v(341)+v(397)*v(469)+v(478)*v(6)+v(487)*v(9);
Tangent(1,3,3,1)=v(397)*v(470)+v(479)*v(6)+v(488)*v(9);
Tangent(1,3,3,2)=v(397)*v(471)+v(480)*v(6)+v(489)*v(9);
Tangent(1,3,3,3)=v(342)+v(397)*v(472)+v(481)*v(6)+v(490)*v(9);
Tangent(2,1,1,1)=v(341)+v(1)*v(473)+v(398)*v(491)+v(500)*v(7);
Tangent(2,1,1,2)=v(1)*v(474)+v(398)*v(492)+v(501)*v(7);
Tangent(2,1,1,3)=v(1)*v(475)+v(398)*v(493)+v(502)*v(7);
Tangent(2,1,2,1)=v(1)*v(476)+v(398)*v(494)+v(654)+v(503)*v(7);
Tangent(2,1,2,2)=v(1)*v(477)+v(398)*v(495)+v(504)*v(7);
Tangent(2,1,2,3)=v(1)*v(478)+v(398)*v(496)+v(505)*v(7);
Tangent(2,1,3,1)=v(344)+v(1)*v(479)+v(398)*v(497)+v(506)*v(7);
Tangent(2,1,3,2)=v(1)*v(480)+v(398)*v(498)+v(507)*v(7);
Tangent(2,1,3,3)=v(1)*v(481)+v(398)*v(499)+v(508)*v(7);
Tangent(2,2,1,1)=v(2)*v(473)+v(399)*v(491)+v(500)*v(8);
Tangent(2,2,1,2)=v(341)+v(2)*v(474)+v(399)*v(492)+v(501)*v(8);
Tangent(2,2,1,3)=v(2)*v(475)+v(399)*v(493)+v(502)*v(8);
Tangent(2,2,2,1)=v(2)*v(476)+v(399)*v(494)+v(503)*v(8);
Tangent(2,2,2,2)=v(2)*v(477)+v(399)*v(495)+v(654)+v(504)*v(8);
Tangent(2,2,2,3)=v(2)*v(478)+v(399)*v(496)+v(505)*v(8);
Tangent(2,2,3,1)=v(2)*v(479)+v(399)*v(497)+v(506)*v(8);
Tangent(2,2,3,2)=v(344)+v(2)*v(480)+v(399)*v(498)+v(507)*v(8);
Tangent(2,2,3,3)=v(2)*v(481)+v(399)*v(499)+v(508)*v(8);
Tangent(2,3,1,1)=v(3)*v(473)+v(400)*v(491)+v(500)*v(9);
Tangent(2,3,1,2)=v(3)*v(474)+v(400)*v(492)+v(501)*v(9);
Tangent(2,3,1,3)=v(341)+v(3)*v(475)+v(400)*v(493)+v(502)*v(9);
Tangent(2,3,2,1)=v(3)*v(476)+v(400)*v(494)+v(503)*v(9);
Tangent(2,3,2,2)=v(3)*v(477)+v(400)*v(495)+v(504)*v(9);
Tangent(2,3,2,3)=v(3)*v(478)+v(400)*v(496)+v(654)+v(505)*v(9);
Tangent(2,3,3,1)=v(3)*v(479)+v(400)*v(497)+v(506)*v(9);
Tangent(2,3,3,2)=v(3)*v(480)+v(400)*v(498)+v(507)*v(9);
Tangent(2,3,3,3)=v(344)+v(3)*v(481)+v(400)*v(499)+v(508)*v(9);
Tangent(3,1,1,1)=v(342)+v(1)*v(482)+v(4)*v(500)+v(401)*v(509);
Tangent(3,1,1,2)=v(1)*v(483)+v(4)*v(501)+v(401)*v(510);
Tangent(3,1,1,3)=v(1)*v(484)+v(4)*v(502)+v(401)*v(511);
Tangent(3,1,2,1)=v(344)+v(1)*v(485)+v(4)*v(503)+v(401)*v(512);
Tangent(3,1,2,2)=v(1)*v(486)+v(4)*v(504)+v(401)*v(513);
Tangent(3,1,2,3)=v(1)*v(487)+v(4)*v(505)+v(401)*v(514);
```

```
Tangent(3,1,3,1)=v(1)*v(488)+v(4)*v(506)+v(401)*v(515)+v(575);
Tangent(3,1,3,2)=v(1)*v(489)+v(4)*v(507)+v(401)*v(516);
Tangent(3,1,3,3)=v(1)*v(490)+v(4)*v(508)+v(401)*v(517);
Tangent(3,2,1,1)=v(2)*v(482)+v(5)*v(500)+v(402)*v(509);
Tangent(3,2,1,2)=v(342)+v(2)*v(483)+v(5)*v(501)+v(402)*v(510);
Tangent(3,2,1,3)=v(2)*v(484)+v(5)*v(502)+v(402)*v(511);
Tangent(3,2,2,1)=v(2)*v(485)+v(5)*v(503)+v(402)*v(512);
Tangent(3,2,2,2)=v(344)+v(2)*v(486)+v(5)*v(504)+v(402)*v(513);
Tangent(3,2,2,3)=v(2)*v(487)+v(5)*v(505)+v(402)*v(514);
Tangent(3,2,3,1)=v(2)*v(488)+v(5)*v(506)+v(402)*v(515);
Tangent(3,2,3,2)=v(2)*v(489)+v(5)*v(507)+v(402)*v(516)+v(575);
Tangent(3,2,3,3)=v(2)*v(490)+v(5)*v(508)+v(402)*v(517);
Tangent(3,3,1,1)=v(3)*v(482)+v(403)*v(509)+v(500)*v(6);
Tangent(3,3,1,2)=v(3)*v(483)+v(403)*v(510)+v(501)*v(6);
Tangent(3,3,1,3)=v(342)+v(3)*v(484)+v(403)*v(511)+v(502)*v(6);
Tangent(3,3,2,1)=v(3)*v(485)+v(403)*v(512)+v(503)*v(6);
Tangent(3,3,2,2)=v(3)*v(486)+v(403)*v(513)+v(504)*v(6);
Tangent(3,3,2,3)=v(344)+v(3)*v(487)+v(403)*v(514)+v(505)*v(6);
Tangent(3,3,3,1)=v(3)*v(488)+v(403)*v(515)+v(506)*v(6);
Tangent(3,3,3,2)=v(3)*v(489)+v(403)*v(516)+v(507)*v(6);
Tangent(3,3,3,3)=v(3)*v(490)+v(403)*v(517)+v(575)+v(508)*v(6);
disp(sprintf("\n%s  %f %f %f %f %f %f %f %f %f " ...
 ,"HyperElasticity_eigenvalues<<   Cauchy stress AceGen=     ",v(384),v(386),v(387),v(388),v(389),v(3
 ),v(391),v(392),v(393)));
v(807)=0;
v(808)=0;
v(809)=0;
v(810)=0;
v(811)=0;
v(812)=0;
v(813)=0;
v(814)=0;
v(815)=0;
for i816=1:1:i21;
 i830=1+i816;
 v(817)=v(1066+i830);
 v(821)=Power(v(453),v(817));
 v(820)=Power(v(433),v(817));
 v(819)=Power(v(413),v(817));
 v(831)=v(385)*v(1066+i21+i830);
 v(807)=v(807)+(v(307)*v(819)+v(316)*v(820)+v(325)*v(821))*v(831);
 v(808)=v(808)+(v(308)*v(819)+v(317)*v(820)+v(326)*v(821))*v(831);
 v(809)=v(809)+(v(309)*v(819)+v(318)*v(820)+v(327)*v(821))*v(831);
 v(810)=v(810)+(v(310)*v(819)+v(319)*v(820)+v(328)*v(821))*v(831);
 v(811)=v(811)+(v(311)*v(819)+v(320)*v(820)+v(329)*v(821))*v(831);
 v(812)=v(812)+(v(312)*v(819)+v(321)*v(820)+v(330)*v(821))*v(831);
 v(813)=v(813)+(v(313)*v(819)+v(322)*v(820)+v(331)*v(821))*v(831);
 v(814)=v(814)+(v(314)*v(819)+v(323)*v(820)+v(332)*v(821))*v(831);
 v(815)=v(815)+(v(315)*v(819)+v(324)*v(820)+v(333)*v(821))*v(831);
end;
disp(sprintf("\n%s  %f ","HyperElasticity_eigenvalues<<   Cauchy stress relative error=",(Power(
 -v(807),2)+Power(v(386)-v(808),2)+Power(v(387)-v(809),2)+Power(v(388)-v(810),2)+Power(v(389)-v(811)      ...
 ,2)+Power(v(390)-v(812),2)+Power(v(391)-v(813),2)+Power(v(392)-v(814),2)+Power(v(393)-v(815),2))/v      ...
 (384)));
```

```
function [x]=SMSKDelta(i,j)

if (i==j) , x=1; else x=0; end;

end

function [x]=SMSDeltaPart(a,i,j,k)

l=round(i/j);

if (mod(i,j) ~= 0 | l>k) , x=0; else x=a(l); end;

end

function [x]=Power(a,b)

x=a^b;

end

function [x]=SMSTernaryOperator(a,b,c)

if (c) , x=a; else x=b; end;

end

end
```