Implementation of Bar3 Template in Mathematica
# TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>§I.1</th>
<th>Auxiliary Modules</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>§I.1.1</td>
<td>Name To Signature Mapper</td>
<td>I–3</td>
</tr>
<tr>
<td>§I.1.2</td>
<td>Mass Template Variant Parameter Mapper</td>
<td>I–4</td>
</tr>
<tr>
<td>§I.2</td>
<td>Element Level Modules</td>
<td>I–5</td>
</tr>
<tr>
<td>§I.2.1</td>
<td>Element Mass and Stiffness Modules</td>
<td>I–5</td>
</tr>
<tr>
<td>§I.3</td>
<td>Assembly Level Modules</td>
<td>I–6</td>
</tr>
<tr>
<td>§I.3.1</td>
<td>Lattice Master Mass and Stiffness Modules</td>
<td>I–7</td>
</tr>
<tr>
<td>§I.3.2</td>
<td>Lattice Patch Modules</td>
<td>I–8</td>
</tr>
<tr>
<td>§I.4</td>
<td>Dispersion Analysis and Display Modules</td>
<td>I–8</td>
</tr>
<tr>
<td>§I.4.1</td>
<td>Characteristic Equation Module</td>
<td>I–8</td>
</tr>
<tr>
<td>§I.4.2</td>
<td>Dispersion Branches And Taylor Series</td>
<td>I–9</td>
</tr>
<tr>
<td>§I.4.3</td>
<td>Dispersion Diagram Plotting</td>
<td>I–11</td>
</tr>
</tbody>
</table>
This Appendix presents the computer implementation of the mass-stiffness template pair for the three-node bar element, abbreviated to Bar3. It is written in the *Mathematica* language. Although the element is admittedly simple it is not trivial. In fact the implementation illustrates the use of template variants to simplify customization. Why *Mathematica*?. As observed in the Introduction, use of a CAS is essential for template development because analytical derivations soon exceed human endurance. Once the development phase is completed, a production version in a compiled language can be easily produced. But the CAS version should not be discarded.

The hierarchical organization of the modules presented in this Appendix is shown in Figure I.1. The bottom-up description that follows starts from the lowest level of that chart, going up and traversing against the arrows.

§I.1. Auxiliary Modules

The two outside modules at the lowest level of the chart of Figure I.1 provide auxiliary services to modules at all levels.

§I.1.1. Name To Signature Mapper

Auxiliary module Bar3TempSignature, listed at the top of Figure I.2, maps an abbreviated template instance name to its full signature definition. It is invoked by

\[ \text{tsign=} \text{Bar3TempSignature}[\text{name}] \]  

The only argument is name: a character string of 3 or 4 letters that abbreviates a template instance. Examples: "CMM" for the consistent mass matrix or "SLMM" for Simpson-lumped mass matrix.

---

Figure I.1. Organization of Bar3 template analysis modules presented in this Appendix.
Table I.1. Bar3 Template Signature List Specification

<table>
<thead>
<tr>
<th>Template variant</th>
<th>Ref. eqn.</th>
<th># of pars</th>
<th>Signature format, identified as tsign in Mathematica code</th>
<th>Mass conservation constraint*</th>
</tr>
</thead>
<tbody>
<tr>
<td>General, $\mu_i$ pars</td>
<td>(23.2)</td>
<td>5</td>
<td>${ \text{&quot;GEN&quot;}, { \beta }, { \mu_1, \mu_2, \mu_3, \mu_4 } }$</td>
<td>$2\mu_1 + 2\mu_2 + 2\mu_3 + 4\mu_4 = 0$</td>
</tr>
<tr>
<td>General, $\chi_i$ pars</td>
<td>(23.7)</td>
<td>4</td>
<td>${ \text{&quot;GEX&quot;}, { \beta }, { \chi_1, \chi_2, \chi_3 } }$</td>
<td>preimposed</td>
</tr>
<tr>
<td>Lumped</td>
<td>(23.26)</td>
<td>3</td>
<td>${ \text{&quot;LUM&quot;}, { \beta }, { \mu_{L1}, \mu_{L2} } }$</td>
<td>$2\mu_{L1} + \mu_{L2} = 0$</td>
</tr>
<tr>
<td>Spectral</td>
<td>(23.34)</td>
<td>3</td>
<td>${ \text{&quot;SPE&quot;}, { \beta }, { \mu_{S1}, \mu_{S2} } }$</td>
<td>preimposed</td>
</tr>
<tr>
<td>Selective Mass Scaling</td>
<td>(23.44)</td>
<td>4</td>
<td>${ \text{&quot;SMS&quot;}, { \beta }, { \mu_{L1}, \mu_{L2}, \mu_K } }$</td>
<td>$2\mu_{L1} + \mu_{L2} = 0$</td>
</tr>
<tr>
<td>Constant Optical Branch†</td>
<td>(23.47,23.49)</td>
<td>1</td>
<td>${ \text{&quot;COBA&quot;}, { 1 }, { \nu_A } }$</td>
<td>preimposed</td>
</tr>
<tr>
<td>or</td>
<td>${ \text{&quot;COBB&quot;}, { 1 }, { \nu_B } }$</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

* When doing symbolic work, the mass conservation constraint is not always preimposed in some template variants, as that may complicate intermediate expressions
† For this variant, two families: COBA and COBB, are implemented. Cf. §23.1.13

Names currently implemented can be gathered by examining the code. Some of these are also listed in Table 23.1. The function returns tsign as template signature. This is a list of the form

$$\{ \text{tvar}, \{ \text{kpars} \}, \{ \text{mpars} \} \}$$  \hspace{1cm} (I.2)

Here tvar is a character string that identifies template variants, kpars a list of stiffness parameters, and mpars a list of mass parameters. Configuration details for this data structure are given in Table I.1. If name is not recognized, a warning message is printed and "CMM" is assumed.

Example: Bar3TempSignature["SLMM"] returns $\{ \text{"LUM"}, \{ 1 \}, \{ 0, 0 \} \}$ as function value.

§I.1.2. Mass Template Variant Parameter Mapper

Auxiliary module Bar3MasVarParMap, listed at the bottom of Figure I.2, returns a replacement rule that maps the four $\mu$ parameters of the Bar3 general mass template (23.2) to those of a variant. The latter is called the target form. The rule is used to specialize results such as dispersion equations; cf. the link drawn in Figure I.1. It is invoked by

$$\text{rule}=\text{Bar3MasVarParMap}[\text{gmpars}, \text{tsign}]$$  \hspace{1cm} (I.3)

The arguments are:

- gmpars A list of symbols used for the free parameters of of the Bar3 mass template (23.2). Normally the parameters are labeled $\{ \mu_1, \mu_2, \mu_3, \mu_4 \}$. Those symbols will appear in the left side of the replacement rule.
- tsign Signature of the target form.

Items in gmpars must be individual symbols, while those in tsign may be symbolic or numeric (see examples below). Note that $\beta$ in tsign is used if the target pertains to the SMS variant.
I.2. Element Level Modules

§I.2.1. Element Mass and Stiffness Modules

The Bar3 element mass and stiffness template modules are called Bar3ElemMassTemp and Bar3ElemStiffTemp, respectively. They are listed in Figure I.3. The call sequences are

\[
M_e = \text{Bar3ElemMassTemp[Le, rho, A, tsign, num]
}\]  

(1.4)
Bar3MassTemp[Le_, ρ_, A_, tsign_, numer_] := Module[
  {me = ρ*Le*A, m1, m2, m3, tvar, β, mpars, kw, varOK, MeZ, Me, modname = "Bar3MassTemp"},
  {tvar, {β}, mpars} = tsign; varOK = False;
  kw = ToUpperCase[tvar]; MeZ = Table[0, {3}, {3}];
  If[kw == "GEN", varOK = True; {µ1, µ2, µ3, µ4} = mpars;
    Me = me*{{4 + µ1, -1 + µ3, 2 + µ4}, {-1 + µ3, 4 + µ1, 2 + µ4}, {2 + µ4, 2 + µ4, 16 + µ2}}/30];
  If[kw == "GEX", varOK = True; {χ1, χ2, χ3} = mpars;
    m1 = χ1 + χ2; m2 = χ1 - χ2; m3 = Sqrt[30]*Sqrt[χ1 - χ3] - 2*χ1;
    Me = me*{{m1, m2, m3}, {m2, m1, m3}, {m3, m3, 30 - 4*χ1 - 4*χ3}}/30];
  If[kw == "LUM", varOK = True; {µL1, µL2} = mpars;
    Me = me*{{5 + µL1, 0, 0}, {0, 5 + µL1, 0}, {0, 0, 20 + µL2}}/30];
  If[kw == "SPE", varOK = True; {µS1, µS2} = mpars;
    Me = me*{{10 + µS1 + µS2, 10 - µS1 + µS2, 10 - 2*µS2}, {10 - µS1 + µS2, 10 + µS1 + µS2, 10 - 2*µS2}, {10 - 2*µS2, 10 - 2*µS2, 30 - 4*µS2}}/90];
  If[kw == "SMS", varOK = True; {µL1, µL2, µK} = mpars;
    MKe = µK*me*Bar3StiffTemp[1, 1, 1, tsign, numer];
    Me = me*{{5 + µL1, 0, 0}, {0, 5 + µL1, 0}, {0, 0, 20 + µL2}}/30 + MKe];
  If[!varOK, Print[modname, ": bad template var ", tvar,
    ", zero matrix returned"]; Return[Me];
  If[!numer, Me = Simplify[Me];]
  Return[Me];
]

Bar3StiffTemp[Le_, Em_, A_, tsign_, numer_] := Module[
  {ke = Em*A/Le, tvar, β, mpars, Keb, Keh, Ke},
  {tvar, {β}, mpars} = tsign;
  Keb = ke*{{1, -1, 0}, {-1, 1, 0}, {0, 0, 0}};
  Keh = (4/3)*β*ke*{{1, 1, -2}, {1, 1, -2}, {1, 1, -2}, {1, 1, -2}, {1, 1, -2}, {1, 1, -2}, {1, 1, -2}, {1, 1, -2}, {1, 1, -2}};
  Ke = Keb + Keh; If[!numer, Ke = Simplify[Ke];]
  Return[Ke];
]

The arguments are:

Le  Element length
Em, A, ρ  Elastic modulus, cross section area, and mass density, respectively, of bar

The template signature. See Table I.1 for configuration details.

numer  Logical flag. If True, process in floating point. If False, process symbolically.

As function values the modules return

Me  3 × 3 element mass matrix
Ke  3 × 3 element stiffness matrix
§1.3 Assembly Level Modules

This section covers modules that work at the assembly (master) level. These are the midlevel four pictured in Figure I.1.

§1.3.1. Lattice Master Mass and Stiffness Modules

Modules Bar3ElemMassTempLattice and Bar3StiffTempLattice, listed in Figure I.4, assemble the master mass and stiffness matrices, respectively, of a prismatic homogeneous bar member discretized as a regular lattice with a given number of elements. Since all elements are identical, only one call to the appropriate element-level module is made. The returning matrix is reused in the merge loop. The call sequences are similar:

\[ M = \text{Bar3MassTempLattice}[\text{numele}, \text{Le}, \rho, \text{A}, \text{tdef}, \text{numer}] \]  
\[ K = \text{Bar3StiffTempLattice}[\text{numele}, \text{Le}, \rho, \text{A}, \text{tdef}, \text{numer}] \]  

Figure I.4. Bar3 master mass and stiffness assembler modules.

Figure I.5. Bar3 mass and stiffness patch extraction modules.
Appendix I: IMPLEMENTATION OF BAR3 TEMPLATE IN MATHEMATICA

The arguments are:

- `numele` Number of elements in lattice. The number of freedoms is `numdof = 2*numele + 1`.
- `Le` Element length. Total member length will be `Le*numele`.
- `Em, A, rho` Elastic modulus, cross section area, and mass density, respectively, of bar.
- `tdef` Template definition argument. Two possibilities:
  - If a list, `tdef` is taken to be the template signature `tsign` configured as shown in Table I.1, and thus passed directly to the element module.
  - If a character string (for example: "CMM"), `tdef` is interpreted as a template instance abbreviation and `Bar3TempSignature` called as per (I.3) to build `tsign`, which is then passed to the element modules.
- `numer` Logical flag; see §I.2.1

As function values the modules return:

- `M` Master mass matrix of order `ndof x ndof`.
- `K` Master stiffness matrix of order `ndof x ndof`.

§I.3.2. Lattice Patch Modules

Modules `Bar3ElemMassPatch` and `Bar3StiffPatch`, listed in Figure I.5, return the assembled mass and stiffness matrices, respectively, of a patch of two identical Bar3 elements. This is done by calling `Bar3MassTempLattice` and `Bar3StiffTempLattice` with `numele=2` and returning only the second and third equations. The call sequences are similar:

\[
\begin{align*}
M &= \text{Bar3MassTempPatch}[Le,\rho,A,tdef,\text{numer}] \\
K &= \text{Bar3StiffTempPatch}[Le,Em,A,tdef,\text{numer}]
\end{align*}
\]

The arguments are identical to those for the lattice master mass and stiffness modules, respectively, described in §I.3.1, except that `numele` is not supplied.

As function values the modules return:

- `Mp` Patch mass equations as a coefficient matrix of order `2 x 5`; see (?)-(?)
- `Kp` Patch stiffness equations as a coefficient matrix of order `2 x 5`; see (?)-(?)

§I.4. Dispersion Analysis and Display Modules

This section describe modules that produce and display dispersion diagrams. Those are the top three shown in Figure I.1.

§I.4.1. Characteristic Equation Module

Module `Bar3CharFreq`, listed in Figure I.6, forms the characteristic equation of a plane wave propagating over a regular Bar3 lattice patch and solves it for the two characteristic frequencies. The calling sequence is:

\[
\{\text{detCm},\Omega_{2aco},\Omega_{2opt}\} = \text{Bar3CharFreq}[\text{wavars},tdef,\text{numer}]
\]

I–8
I.4 DISPERSION ANALYSIS AND DISPLAY MODULES

Bar3CharFreq[wavars_, tdef_, numer_] := Module[
  {κ, ζ, Ω1, Ω2, τ, Bc, Bm, Kp, Mp, up, fm, fc, mfac, cfac, Cm, detCm, 
   Ωsol, Ω21, Ω22, Ω2aco, Ω2opt}, 
  {κ, ζ, Ω1, Ω2, τ, Bc, Bm} = wavars;
  PlaneWave[k_, ω_, B_, x_, t_] := B*Exp[I*(k*x - ω*t)];
  Mp = Bar3MassTempPatch[1, 1, 1, tdef, numer];
  Kp = Bar3StiffTempPatch[1, 1, 1, tdef, numer];
  up = {PlaneWave[κ, Ω1, Bc, -1, τ], PlaneWave[κ, Ω1, Bm, -1/2, τ],
        PlaneWave[κ, Ω1, Bc, 0, τ], PlaneWave[κ, Ω1, Bm, 1/2, τ],
        PlaneWave[κ, Ω1, Bc, 1, τ]};
  {fm, fc} = Simplify[ExpToTrig[(Kp - Ω1^2*Mp).up]];
  mfac = (Cos[κ/2 + Ω1*τ] - I*Sin[κ/2 + Ω1*τ]);
  cfac = (Cos[Ω1*τ] - I*Sin[Ω1*τ]);
  {fm, fc} = Simplify[{fm/mfac, fc/cfac}];
  Cm = {{Coefficient[fm, Bm], Coefficient[fm, Bc]},
        {Coefficient[fc, Bm], Coefficient[fc, Bc]}};
  detCm = Simplify[Det[Cm]];
  Ωsol = Simplify[Solve[detCm == 0, Ω21]];
  Ω2aco = Ω21 = Simplify[Ω2/Ωsol[[1]]];
  Ω2opt = Ω22 = Simplify[Ω2/Ωsol[[2]]];
  If[Ω22 == 0, Ω2aco = Ω22; Ω2opt = Ω21];
  Return[{detCm, Ω2aco, Ω2opt}];
]

The arguments are:

wavars A list of symbols representing plane wave dispersion analysis variables, configured as the list \{κ, ζ, Ω1, Ω2, τ, Bc, Bm\}, in which
κ Dimensionless wavenumber \(κ = k/ℓ\)
ζ Dimensionless space coordinate \(ζ = x/ℓ \in [-1, 1]\) over patch.
Ω Dimensionless characteristic frequency \(ω/ℓc_0\)
Ω2 Dimensionless characteristic squared frequency
τ Dimensionless time \(τ = tc_0/ℓ\)
Bc, Bm Corner and midpoint wave component amplitudes, respectively

These must be \textit{individual symbols}. No numbers or expressions should be in this list, because they are internally used as variables. For instance, entering Ω*Ω or Ω^2 for Ω2 will cause errors.

\(tdef\) Template definition argument; see §I.3.1
\(numer\) Logical flag; see §I.2.1

The module returns the list \{detCm, Ω2aco, Ω2opt\}, in which

detCm Determinant of characteristic matrix \(C_m\) as a function of \(κ\) and \(Ω^2\)
Ω2aco Dimensionless characteristic squared frequency \(Ω^2_a\) of acoustic branch (AB), expressed as function of \(κ\)
Ω2opt Dimensionless characteristic squared frequency \(Ω^2_a\) of optical branch (OB), expressed as function of \(κ\)

The last two expressions: \(Ω2aco\) and \(Ω2opt\), collectively define the dimensionless dispersion diagram (DDD) for the template specified by \(tdef\).
Appendix I: IMPLEMENTATION OF BAR3 TEMPLATE IN MATHEMATICA

Figure I.7. Bar3 dispersion module that returns dimensionless characteristic squared frequencies as function of dimensionless wavenumber, and their Taylor series up to given order about \( \kappa = 0 \)

§I.4.2. Dispersion Branches And Taylor Series

Given a Bar3 template (or instance) definition, module Bar3Dispersion, listed in Figure I.7, returns the dimensionless characteristic squared frequencies \( \Omega_1^2 \) and \( \Omega_0^2 \) of the AB and OB as function of the dimensionless wavenumber \( \kappa \). This module was built by inlining symbolic results produced by Bar3CharFreq with the goal of speeding up direct retrieval of those expressions. In addition, this module can compute and return their Taylor series expansions about \( \kappa = 0 \) up to specified orders. The calling sequence is

\[
\{ \Omega_{2aco}, \Omega_{2opt}, \Omega_{2acos}, \Omega_{2opts} \} = \text{Bar3Dispersion}[\kappa, \text{tdef}, \{ \text{ma}, \text{mo} \}, \text{slevel}] \tag{1.11}
\]

The arguments are:

- \( \kappa \)  Dimensionless wavenumber
- \( \text{tdef} \)  Template definition argument; see §I.3.1
- \( \text{ma} \)  If \( \text{ma} \geq 0 \), return acoustic branch Taylor series (ABTS): \( \Omega_1^2 \) expanded in \( \kappa \) about \( \kappa = 0 \), up to and including order \( \text{ma} \). If a negative integer, return Null.
- \( \text{mo} \)  If \( \text{mo} \geq 0 \), return optical branch Taylor series (OBTS): \( \Omega_0^2 \) expanded in \( \kappa \) about \( \kappa = 0 \), up to and including order \( \text{mo} \). If a negative integer, return Null.
- \( \text{slevel} \)  Simplification level for output results: an integer 0, 1 or 2.
§I.4 DISPERSION ANALYSIS AND DISPLAY MODULES

I.4.3 Dispersion Diagram Plotting

Given a Bar3 template instance (that is, with all-numeric signature) module Bar3DispersionPlot, listed in Figure I.8, can plot its DDD, which includes the acoustic and optical branches returned by Bar3Dispersion. It may also plot its DGVD, which is the ratio $\gamma = \gamma_c / c_0$ of the FEM plane wave speed $c$ to that of the continuum wave speed $c_0$. The module has been used to produce all Bar3 dispersion plots of this paper.

The call sequence is

$$\text{Bar3DispersionPlot}[\kappa, \text{tdef}, \text{plotwhat}, \kappa\text{range}, \text{DVrange}, \text{imgsiz}, \text{title}] \quad (I.12)$$

---

Figure I.8. Bar3 dispersion diagram plotting module.

0: (or negative): no simplifications
1: ordinary simplification using the Simplify function
2: more exhaustive simplification using the FullSimplify function. Note: this level should be used with caution. Reason: full simplification may return unexpected weird results with terms involving Abs, or conditional expressions.

The module returns the list $\{\Omega_2\text{aco}, \Omega_2\text{opt}, \Omega_2\text{acos}, \Omega_2\text{opts}\}$, in which

- $\Omega_2\text{aco}$ AB dimensionless characteristic squared frequency $\Omega_2^\kappa$ expressed as function of $\kappa$
- $\Omega_2\text{opt}$ OB dimensionless characteristic squared frequency $\Omega_2^\kappa$ expressed as function of $\kappa$
- $\Omega_2\text{acos}$ ABTS about $\kappa = 0$ up to and including order $m_a$. If $m_a < 0$, Null is returned.
- $\Omega_2\text{ocos}$ OBTS about $\kappa = 0$ up to and including order $m_a$. If $m_0 < 0$, Null is returned.
The arguments are:

- $\kappa$  
  Dimensionless wavenumber

- tdef  
  Template definition argument; see §I.3.1

- plotwhat  
  A character string of the form "D", "V", or "DV". If the letter D appears, plot the DDD (the so-called "D-plot"). If the letter V appears, plot the DGVD (the so-called "V-plot"). If none of those strings is given, no plot is produced.

- $\kappa$range  
  A 2-item list $\{\kappa_{\text{min}}, \kappa_{\text{max}}\}$ that specifies the (horizontal) plot range for $\kappa$. It is used for both DDD and DGVD plots. Usual range is $\{0, 2\ \pi\}$.

- DVrange  
  A list of the form $\{\Omega_{\text{range}}, \gamma_{\text{range}}\}$. $\Omega_{\text{range}}$ is in turn a two-item list: $\{\Omega_{\text{min}}, \Omega_{\text{max}}\}$ that specifies the DDD plot range for $\Omega$. $\gamma_{\text{range}}$ has a similar configuration: $\{\gamma_{\text{cmin}}, \gamma_{\text{cmax}}\}$ and specifies the DGVD range for $\gamma_c = c/c_0$. Both lists must be supplied even if only one plot is requested. Common specifications are $\{0, 8\}$ for $\Omega_{\text{range}}$ and $\{-2, 2\}$ for $\gamma_{\text{range}}$.

- imgsiz  
  Width of plot in points. Normally set to 300 to 400.

- title  
  An optional character string to be printed before the plot. If " " no title appears.

The module does not return a value. Its output is the plot image object written to the Mathematica default display function. (Its name changed in Version 6.0 from $\$DisplayFunction$ to Print.)