

Classroom Tips and Techniques: An Undamped coupled Oscillator

Robert J. Lopez
Emeritus Professor of Mathematics and Maple Fellow
Maplesoft

Introduction

By coincidence, right after completing last month's article *Simultaneous Diagonalization and the Generalized Eigenvalue Problem*, a long-time Maple user sent me a Maple worksheet whose purpose was to solve a three-degree-of-freedom spring-mass system using exactly the mathematics in the as-yet unreleased article. While editing the worksheet, it became clear that although the theory was easy enough to articulate, implementing an efficient solution of a real problem in Maple was something worth detailing.

Consequently, this month's article is a solution of the initial value problem I was sent. The purpose of the article is an articulation of the details of the calculations, showing how the theory developed in last month's article leads to a solution of a system of linear, second-order, constant-coefficient differential equations. In addition, we aim to show how the use of math-mode input simplifies the notation, and illuminates the connection between the underlying mathematics, and Maple's expression of it.

Initializations

```
> restart  
> with(LinearAlgebra) : with(plots) : with(VariationalCalculus) :  
>  
> interface(typesetting = extended) :  
>
```

Data Definitions

We take as given the following definitions of KE and PE , the kinetic and potential energies of the system.

$$\begin{aligned}
> \quad KE &:= \frac{1}{2} \sum_{j=1}^3 m_j \dot{x}_j^2 : \\
PE &:= \frac{1}{2} \left(\sum_{j=1}^3 k_j x_j^2(t) + k_4 (x_2(t) - x_1(t))^2 + k_5 (x_3(t) - x_2(t))^2 \right. \\
&\quad \left. + k_6 (x_3(t) - x_1(t))^2 \right) :
\end{aligned}$$

> The Lagrangian $L = KE - PE$ is then

$$> \quad Lt := KE - PE :$$

> so that the Euler-Lagrange equations for the problem are

$$> \quad EL := \text{convert}(\text{remove}(\text{has}, \text{EulerLagrange}(Lt, t, [x_1(t), x_2(t), x_3(t)]), K), \text{list}) :$$

> where we have removed the first integrals that the **EulerLagrange** command generates. Except for the zeros on the right-hand sides, the individual equations are

$$\begin{aligned}
> \quad \text{for } j \text{ to } 3 \text{ do } E||j &:= -\text{select}(\text{has}, EL, m_j)[] \text{end do} \\
E1 &:= k_1 x_1(t) - k_4 (x_2(t) - x_1(t)) - k_6 (x_3(t) - x_1(t)) + m_1 \ddot{x}_1(t) \\
E2 &:= k_2 x_2(t) + k_4 (x_2(t) - x_1(t)) - k_5 (x_3(t) - x_2(t)) + m_2 \ddot{x}_2(t) \\
E3 &:= k_3 x_3(t) + k_5 (x_3(t) - x_2(t)) + k_6 (x_3(t) - x_1(t)) + m_3 \ddot{x}_3(t)
\end{aligned}$$

> If the following matrices are defined, the system can be written in the form $M\ddot{\mathbf{u}} + K\mathbf{u} = \mathbf{0}$, where \mathbf{u} is a vector whose components are the unknowns $x_j(t)$.

$$\begin{aligned}
> \quad \text{SymbolicK} &:= \text{GenerateMatrix}([E1, E2, E3], [\text{seq}(x_j(t), j = 1 .. 3)])_1 ; \\
\text{SymbolicM} &:= \text{DiagonalMatrix}([m_1, m_2, m_3])
\end{aligned}$$

$$\text{SymbolicK} := \begin{bmatrix} k_1 + k_6 + k_4 & -k_4 & -k_6 \\ -k_4 & k_2 + k_5 + k_4 & -k_5 \\ -k_6 & -k_5 & k_6 + k_3 + k_5 \end{bmatrix}$$

$$\text{SymbolicM} := \begin{bmatrix} m_1 & 0 & 0 \\ 0 & m_2 & 0 \\ 0 & 0 & m_3 \end{bmatrix}$$

> The parameters appearing in the mass and stiffness equations are entered in set of equations, not hard-coded as assigned values. This keeps all the symbols as variables, and does not conflict the

calculations with floats where they are not needed.

$$\begin{aligned} > \text{params} := \left\{ k_1 = 1000, k_2 = 1000, k_3 = 1000, k_4 = 1000, k_5 = 1000, k_6 \right. \\ & \quad \left. = 1000, m_1 = \frac{1}{2}, m_2 = 1., m_3 = 3/2 \right\}; \end{aligned}$$

>

To obtain the numeric mass and stiffness matrices, evaluate the symbolic forms at the parameter values. By including one mass value as a float, all calculations involving these matrices will be executed in floating-point form. The exact representations of the eigenpairs for these matrices suffer badly from expression swell. This is to be expected, since the underlying math is the exact solution of a cubic characteristic equation.

$$> M := \text{eval}(\text{SymbolicM}, \text{params}); K := \text{eval}(\text{SymbolicK}, \text{params})$$

$$M := \begin{bmatrix} \frac{1}{2} & 0 & 0 \\ 0 & 1. & 0 \\ 0 & 0 & \frac{3}{2} \end{bmatrix}$$
$$K := \begin{bmatrix} 3000 & -1000 & -1000 \\ -1000 & 3000 & -1000 \\ -1000 & -1000 & 3000 \end{bmatrix}$$

>

The initial data for the problem is likewise entered as a set of equations, not hard-coded as assignments.

$$\begin{aligned} > \text{inits} := \{ x_1(0) = 0.5, x_2(0) = -0.05, x_3(0) = 0, D(x_1)(0) = 0, \\ & \quad D(x_2)(0) = 0, D(x_3)(0) = 0 \}; \end{aligned}$$

>

dsolve Solution

Although the system is linear, with constant coefficients, an exact solution suffers from expression swell because, in essence, an exact solution needs to access the exact roots of a cubic equation. A numeric solution is therefore sought. The specific ODEs for this problem are then

$$> \text{DEs} := \text{eval}(\{E1, E2, E3\}, \text{params})$$

$$DEs := \left\{ 3000x_1(t) - 1000x_2(t) - 1000x_3(t) + \frac{\ddot{x}_1(t)}{2}, 3000x_2(t) - 1000x_1(t) - 1000x_3(t) + \ddot{x}_2(t), 3000x_3(t) - 1000x_2(t) - 1000x_1(t) + \frac{3\ddot{x}_3(t)}{2} \right\}$$

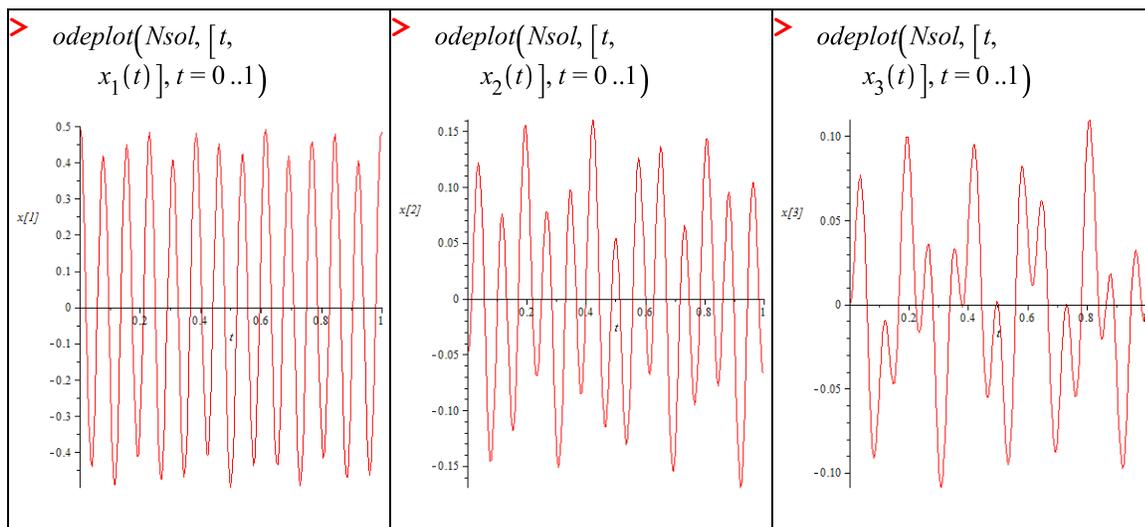
>

Invoke the **dsolve** command with the numeric option. The adroit use of sets allows the equations and the initial data to be coalesced by set union.

> `Nsol := dsolve(DEs ∪ inits, numeric) :`

>

Obtain graphs of each dependent variable. Note the use of the **odeplot** command from the *plots* package.



>

The motion is highly oscillatory, so we expect the angular frequencies (which are the square roots of the generalized eigenvalues) to be relatively large.

Structure

This section sketches the algorithm for the simultaneous diagonalization of the mass and stiffness matrices. This algorithm requires both matrices to be symmetric, and for M to be positive definite, which in this case, it certainly is.

The differential equations are $M\ddot{\mathbf{u}} + K\mathbf{u} = \mathbf{0}$. A matrix S exists, and with it, both M and K are simultaneously diagonalized. Simultaneous diagonalization of the matrices K and M means

that $S^T M S = I$ and $S^T K S = \Lambda$, so that $M = (S^T)^{-1} S^{-1}$ and $K = (S^T)^{-1} \Lambda S^{-1}$. Hence, the differential equation becomes

$$(S^T)^{-1} S^{-1} \ddot{\mathbf{u}} + (S^T)^{-1} \Lambda S^{-1} \mathbf{u} = \mathbf{0}$$

or

$$S^{-1} \ddot{\mathbf{u}} + \Lambda S^{-1} \mathbf{u} = \mathbf{0}$$

upon multiplying from the left by S^T . Defining $\mathbf{X} = S^{-1} \mathbf{u}$, or equivalently, $\mathbf{u} = S \mathbf{X}$, puts the differential equation into the uncoupled form $\ddot{\mathbf{X}} + \Lambda \mathbf{X} = \mathbf{0}$, from which a solution is immediately available. The square roots of the diagonal elements in Λ are the angular frequencies for the normal modes of oscillation of this undamped vibrating system.

Alternatively, if solutions of the form $\mathbf{u} = e^{\omega t} \mathbf{v}$ are sought, where \mathbf{v} is a constant vector, the differential equation becomes $\omega^2 M \mathbf{v} + K \mathbf{v} = \mathbf{0}$, a generalized eigenvalue problem.

The matrix S is obtained as follows. Obtain $M = LL^T$, the Cholesky decomposition of M , where L is a lower triangular factor. Form the matrix $L^{-1} K (L^{-1})^T$ and obtain its eigenpairs. Apparently, it is useful to sort the eigenpairs for physical reasons, but the mathematics underlying these calculations does not require this.

Form H , a matrix whose columns are the normalized eigenvectors just obtained. (Maple returns normalized eigenvectors by default, so this step does not need to be implemented in the calculations below.) The required matrix S is given by $S = (L^{-1})^T H$.

The References section at the end of this document point to three different sources where this algorithm is derived.

Generalized Eigenvalue Problem and Its Eigenpairs

Maple's **Eigenvalues** and **Eigenvectors** commands will solve the generalized eigenvalue problem. There's more work involved in sorting the eigenpairs than there is in finding them.

```
> temp := map(simplify, [Eigenvectors(K, M, output = list)]):
Eigenpairs := sort(temp[ ], (a, b) -> is(a[1] < b[1])):
> print~(Eigenpairs)[1];
```

$$\left[\begin{array}{c} 950.2505048, 1, \left\{ \begin{array}{c} -0.4873561197 \\ -0.5632820932 \\ -0.6672310665 \end{array} \right\} \\ 3351.809967, 1, \left\{ \begin{array}{c} 0.2295974482 \\ 0.8232250697 \\ -0.5192162328 \end{array} \right\} \end{array} \right]$$

$$\left[6697.939528, 1, \left\{ \left[\begin{array}{c} 0.9668177257 \\ -0.2332993625 \\ -0.1040907911 \end{array} \right] \right\} \right]$$

>

The generalized eigenvalues can be put into a vector Λ , and their square roots, into a vector λ .

> $\Lambda := \langle \text{seq}(\text{Eigenpairs}_j[1], j = 1..3) \rangle$

$\lambda := \text{map}(\text{sqrt}, \Lambda)$

$$\Lambda := \begin{bmatrix} 950.2505048 \\ 3351.809967 \\ 6697.939528 \end{bmatrix}$$

$$\lambda := \begin{bmatrix} 30.82613347 \\ 57.89481814 \\ 81.84094041 \end{bmatrix}$$

>

Simultaneous Diagonalization

The following calculations implement the algorithm for finding S that simultaneously diagonalizes the mass and stiffness matrices. Begin by obtaining the Cholesky decomposition of M .

> $L := \text{LUDecomposition}(M, \text{method} = \text{Cholesky})$

$$L := \begin{bmatrix} 0.7071067811865476 & 0. & 0. \\ 0.0 & 1.0 & 0. \\ 0.0 & 0.0 & 1.224744871391589 \end{bmatrix}$$

>

Form the matrix $L^{-1}K(L^{-1})^T$ and obtain its eigenpairs, sorted. The eigenvectors are automatically normalized by Maple's **Eigenvectors** command. Form H , a matrix of the sorted eigenvectors. Then define $S = (L^{-1})^T H$. These steps are summarized in Table 1.

<ul style="list-style-type: none"> Form the matrix $L^{-1}K(L^{-1})^T$. 	<p>> $\text{Temp1} := L^{-1}.K.(L^{-1})^{\%T}$:</p>
<ul style="list-style-type: none"> Obtain the eigenpairs of this matrix, each in the form $[a, b, c]$, where a is the eigenvalue, b is its algebraic multiplicity, and c is a set of any corresponding eigenvectors. 	<p>> $\text{Temp2} := \text{simplify}([\text{Eigenvectors}(\text{Temp1}, \text{output} = \text{list})])$:</p>

<ul style="list-style-type: none"> Sort the three lists by ordering the eigenvalues from smallest to largest. 	<pre>> EP := sort(Temp2[], (a, b) → is(a[1] < b[1])) :</pre>
<ul style="list-style-type: none"> Construct the matrix H whose columns are the eigenvectors, now both unit and sorted. 	<pre>> H := Matrix([seq(EP[k][3][], k = 1 ..3)]) :</pre>
<ul style="list-style-type: none"> Form the matrix $S = (L^{-1})^T H$. 	<pre>> S := (L^{-1})%T.H :</pre>
<ul style="list-style-type: none"> Form the vector Λ whose elements are the sorted eigenvalues. These are the squares of the modal angular frequencies. 	<pre>> Λ := ⟨seq(EP[k][1], k = 1 ..3)⟩; Λ := $\begin{bmatrix} 950.2505048 \\ 3351.809967 \\ 6697.939528 \end{bmatrix}$</pre>
<ul style="list-style-type: none"> Form the vector λ whose elements are the modal angular frequencies. 	<pre>> λ := map(sqrt, Λ) λ := $\begin{bmatrix} 30.82613347 \\ 57.89481814 \\ 81.84094041 \end{bmatrix}$</pre>
<p>Table 1 Obtaining the matrix S</p>	

>

The matrix S is

> S

```
[[ -0.463866737142020, -0.218078018204323, 1.31805524469171],
 [ -0.536133262900000, -0.781921981900000,
  -0.318055244700000],
 [ -0.635072147896599, 0.493165964854146,
  -0.141906183042453]]
```

>

The number of digits in the elements of S is a function of the calculation that produced L . The numerical routine that returned the Cholesky decomposition of M works in "hardware floats," that is, floating-point numbers implemented in the hardware of the computer. These are different from "software floats," the Maple floating-point numbers that can have an arbitrary number of digits. For a software float, the **evalf** command can be used to change the number of digits. Doing this with a hardware float requires a bit more care.

> $map(evalf[10], S)$

$$\begin{bmatrix} -0.4638667371 & -0.2180780182 & 1.318055245 \\ -0.5361332629 & -0.7819219819 & -0.3180552447 \\ -0.6350721479 & 0.4931659649 & -0.1419061830 \end{bmatrix}$$

>

Table 2 contains some verifications of the preceding calculations.

<ul style="list-style-type: none"> Show that $S^TMS = I$. 	<p>> <code>map(simplify@fnormal, S%T.M.S, 9)</code></p> $\begin{bmatrix} 1. & 0. & 0. \\ 0. & 1. & 0. \\ 0. & 0. & 1. \end{bmatrix}$
<ul style="list-style-type: none"> Show that $S^TKS = \Lambda$. 	<p>> <code>map(simplify@fnormal, S%T.K.S, 8)</code></p> $\begin{bmatrix} 950.2505000, 0., 0. \\ 0., 3351.810000, 0. \\ 0., 0., 6697.939500 \end{bmatrix}$
<ul style="list-style-type: none"> Show that $H^TH = I$; i.e., that H is an orthogonal matrix. 	<p>> <code>map(simplify@fnormal, H%T.H, 9)</code></p> $\begin{bmatrix} 1. & 0. & 0. \\ 0. & 1. & 0. \\ 0. & 0. & 1. \end{bmatrix}$
<p>Table 2 Verifications: $S^TMS = I$, $S^TKS = \Lambda$, and $H^TH = I$</p>	

>

To morph these matrices and vectors into an actual solution of the given IVP, write the general solution of the transformed (i.e., uncoupled) system $\ddot{\mathbf{X}} + \Lambda \mathbf{X} = \mathbf{0}$, where here, Λ is a diagonal matrix whose diagonal elements are the components of the vector Λ .

> $\mathbf{X}_g := \langle seq(a_j, j = 1..3) \rangle \cdot \sim \cos \sim (\lambda t) + \langle seq(b_j, j = 1..3) \rangle \cdot \sim \sin \sim (\lambda t)$

$$\mathbf{X}_g := \begin{bmatrix} a_1 \cos(30.82613347 t) + b_1 \sin(30.82613347 t) \\ a_2 \cos(57.89481814 t) + b_2 \sin(57.89481814 t) \\ a_3 \cos(81.84094041 t) + b_3 \sin(81.84094041 t) \end{bmatrix}$$

>

Note the use of the elementwise operator, the tilde. The construct $\cos \sim (\lambda t)$ or $\sin \sim (\lambda t)$ multiplies each element of the vector λ by t , and then computes the cosine or sine of that

product. The vector of cosines is multiplied elementwise by a vector of constants a_j ; the vector of sines, by a vector of constants b_j .

Define vectors of initial values for the original initial value problem.

> $x0 := \langle 0.5, -0.05, 0 \rangle$; $xdot0 := \langle 0, 0, 0 \rangle$

$$x0 := \begin{bmatrix} 0.5 \\ -0.05 \\ 0 \end{bmatrix}$$

$$xdot0 := \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

>

Recall that $\mathbf{X} = S^{-1}\mathbf{u}$, where \mathbf{X} is the solution of the uncoupled system, and \mathbf{u} is the solution of the original coupled system. Since $\mathbf{X}(t)$ is known, but $\mathbf{u}(0)$ and $\dot{\mathbf{u}}(0)$ are given, the constants a_j and b_j in $\mathbf{X}(t)$ are determined by the equations $\mathbf{X}(0) = S^{-1}\mathbf{u}(0)$ and $\dot{\mathbf{X}}(0) = S^{-1}\dot{\mathbf{u}}(0)$. The **Equate** and **solve** commands are instrumental in setting up and solving these algebraic equations.

>

$A := \text{solve}(\text{Equate}(\text{eval}(\mathbf{Xg}, t=0), S^{-1} \cdot x0)); B := \text{solve}(\text{Equate}(\text{eval}(\text{map}(\text{diff}, \mathbf{Xg}, t), t=0), S^{-1} \cdot xdot0))$

$$A := \{a_1 = -0.08916002112, a_2 = -0.01542340544, a_3 = 0.3454165734\}$$

$$B := \{b_1 = 0., b_2 = 0., b_3 = 0.\}$$

>

Substitute these values for the coefficients into \mathbf{Xg} , the general solution of the uncoupled system.

> $\mathbf{X} := \text{eval}(\mathbf{Xg}, A \cup B)$

$$X := \begin{bmatrix} -0.08916002112 \cos(30.82613347t) \\ -0.01542340544 \cos(57.89481814t) \\ 0.3454165734 \cos(81.84094041t) \end{bmatrix}$$

>

This is the specific solution of the uncoupled system. It has to be transformed to the solution of the original coupled system. This final solution is $\mathbf{u} = S\mathbf{X}$, obtained below with all the digits of the hardware floats in S , but displayed with just five digits.

> $\mathbf{u} := S \cdot \mathbf{X} : \text{map}(\text{evalf}[5], \mathbf{u})$

$$\begin{aligned}
& [[0.041358 \cos(30.826 t) + 0.0033635 \cos(57.895 t) \\
& \quad + 0.45528 \cos(81.841 t)], \\
& [0.047802 \cos(30.826 t) + 0.012060 \cos(57.895 t) \\
& \quad - 0.10986 \cos(81.841 t)], \\
& [0.056623 \cos(30.826 t) - 0.0076063 \cos(57.895 t) \\
& \quad - 0.049017 \cos(81.841 t)]]
\end{aligned}$$

>

Test that this solution satisfies the initial conditions:

> `map(simplify@fnormal, eval(u, t = 0), 9); map(simplify@fnormal, eval(map(diff, u, t), t = 0))`

$$\begin{bmatrix} 0.5000000000 \\ -0.0500000000 \\ 0. \end{bmatrix}
\begin{bmatrix} 0. \\ 0. \\ 0. \end{bmatrix}$$

>

Comparing Solutions

Ordinarily, a graph would provide acceptable evidence that two solutions were equivalent. Given the highly oscillatory behavior of the solution of this initial value problem, we instead compare the values of the two solutions (one from **dsolve**, and one from uncoupling) at ten equispaced points in the interval $0.1 \leq t \leq 1$.

Once again obtain the numeric solution provided by the **dsolve** command, but this time, set the output to be a list of procedures.

> `Nsol := dsolve(DEs ∪ inits, numeric, output = listprocedure) :`

>

To see what the list of procedures is, call the solution at some given time.

> `Nsol(1)`

$$\begin{aligned}
& \left[t(1) = 1., x_1(t)(1) = 0.484620716081673, \frac{d}{dt} x_1(t)(1) = \right. \\
& \quad \left. -5.39977091827232, x_2(t)(1) = -0.0660550136699385, \right. \\
& \quad \left. \frac{d}{dt} x_2(t)(1) = 1.56720454329482, x_3(t)(1) = \right. \\
& \quad \left. -0.00303217163310944, \frac{d}{dt} x_3(t)(1) = 2.03737405500597 \right]
\end{aligned}$$

>

Extract the procedures that give the values of the three dependent variables describing position.

> $Y1 := rhs(Nsol_2) : Y2 := rhs(Nsol_4) : Y3 := rhs(Nsol_6) :$

>

At ten equispaced points in the interval $0.1 \leq t \leq \pi$, calculate the norm of the difference between the numeric solution and the vector solution obtained by the simultaneous diagonalization process detailed in this document.

> $seq(\|map(simplify@fnormal, evalf(eval(Y1(t), Y2(t), Y3(t)) - u, t = j\pi/10)), 7)\|, j = 1..10)$

0., 0., 0., 0., 0., 0., 0., 0., 0., 0.

>

To seven places, the solutions agree at these ten points.

>

References

- Numerical Analysis: A Practical Approach, 3rd ed., Melvin J. Marin and Robert J. Lopez, Wadsworth Publishing Company, 1991.
- Advanced Engineering Mathematics, Robert J. Lopez, Addison-Wesley Publishing Company, 2001. Now available from Maplesoft as the ebook [Advanced Engineering Mathematics with Maple](#).
- Applied Linear Algebra, 3rd ed., Ben Noble and James W. Daniel, Prentice-Hall Publishing Company, 1988.

Legal Notice: © Maplesoft, a division of Waterloo Maple Inc. 2012. Maplesoft and Maple are trademarks of Waterloo Maple Inc. This application may contain errors and Maplesoft is not liable for any damages resulting from the use of this material. This application is intended for non-commercial, non-profit use only. Contact Maplesoft for permission if you wish to use this application in for-profit activities.