19 February 2023

EXERCISE SOLUTIONS

for

COMPUTATION AND PROBLEM SOLVING IN UNDERGRADUATE PHYSICS

IDL		MATLAB	OCTAVE
PYTHON	●	MAXIMA	MAPLE
MATHEMATICA		FORTRAN	C
NUMERICAL RECIPES		LSODE	MUDPACK
LaTeX		TGIF	

DAVID M. COOK

with assistance from

DANICA DRALUS, LU '02
RYAN PETERSON, LU '03
SCOTT KAMINSKI, LU '04
MICHELLE MILNE, LU '04
LAUREN KOST, LU '05
CLAIRE WEISS, LU '07
ERIK GARBACIK, LU '08

Department of Physics
Lawrence University
711 E Boldt Way SPC 24
Appleton, Wisconsin 54911

Preface

This document contains solutions to selected exercises from *Computation and Problem Solving in Undergraduate Physics* (*CPSUP*), a book that has grown from small beginnings in the 1990s to a flexible volume that provides an orientation to a subset of tools chosen from

- the general purpose programs IDL, MATLAB, OCTAVE, PYTHON, MAXIMA, MAPLE, and MATHEMATICA,
- the programming languages FORTRAN and C,
- FORTRAN NUMERICAL RECIPES,
- C NUMERICAL RECIPES,
- the FORTRAN procedure library LSODE,
- the UNIX drawing program TGIF, and
- the tool LaTeX for preparation of technical documents.

In addition, chapters on ordinary differential equations, integration, and root finding provide examples of the use of the selected subset of tools for solving a wide variety of problems in physics. Problems from mechanics, electromagnetic theory, quantum mechanics, thermodynamics, statistical mechanics, relativity, and other subarea of physics are included.

This document admits the same flexibility in composition that characterizes `CPSUP`. Both *CPSUP* and this document can be configured to include all of the possibilities or only a selected subset of the options. Because there are 13 different components, each of which can be included or not, there are technically $2^{13} = 8012$ versions of these items. To be sure, the vast majority of these possibilities makes no sense. Still, the number of versions is staggering. Creating documents with this degree of flexibility would be impossible without exploiting the elegant features of the `ifthen` package in LaTeX, and I owe an immense debt to Donald Knuth, Leslie Lamport, and numerous others who have contributed to the development of that publishing system.

David M. Cook
Appleton, Wisconsin
19 February 2023

Table of Contents

Preface iii

Table of Contents v

6 Introduction to MAXIMA 1

6.2 A Few MAXIMA Manipulations . . . 1
6.3 More MAXIMA Manipulations . . . 3
6.4 Damped Harmonic Oscillation . . . 6
6.5 Legendre Polynomials . . . 12
6.6 Wheatstone Bridge . . . 20
6.7 Quantum Wave Scattering . . . 23
6.8 Black Body Spectrum . . . 26
6.10 Laguerre Polynomials . . . 30
6.12 Finding Eigenvectors/Values . . . 35
6.13 Stark Effect for $n = 2$ and $n = 3$ . . . 37
6.15 Lagrangian Approach to Equations of Motion . . . 41
6.17 Elliptic Integrals . . . 47
6.18 Laplace Transforms . . . 49
6.19 Fitting Parabola Through Three Points . . . 51
6.20 Deducing Simpson's Rule . . . 52
6.21 Accuracy in Simpson's Rule . . . 53
6.22 Potential Energy with a Bump . . . 55
6.23 Solving ODEs Symbolically . . . 58
6.25 Vector Derivatives in Cylindrical Coordinates . . . 65
6.27 Vector Product Identities . . . 68

9 Introduction to Programming 71

in Psuedocode

9.2 Relationship between IF-THEN-ELSE and CASE . . . 71
9.7 Tracking Both Extremes and their Positions . . . 74
9.9 Mystery Procedure 1 . . . 76

11 Solving ODEs 79

symbolically with MAXIMA

11.1 Driven Oscillator (MAXIMA) . . . 79
11.2 Two Masses, One Spring (MAXIMA) . . . 84
11.4 Compound Pendulum (MAXIMA) . . . 88
11.6 Series Solution (MAXIMA) . . . 92
11.7 Projectile in Linear Air Resistance (MAXIMA) . . . 99
11.12 Vibrating String Fixed at Both Ends (MAXIMA) . . . 102

13 Evaluating Integrals 105

symbolically with MAXIMA

13.1 Non-relativistic Motion under Constant Force (MAXIMA) . . . 105
13.2 Non-relativistic Motion under Constant Force (MAXIMA) . . . 106
13.3 Non-relativistic Motion under Spring Force (MAXIMA) . . . 108
13.4 Non-Relativistic Motion with Decaying Force (MAXIMA) . . . 110
13.5 Properties of Lorentz Distribution (MAXIMA) . . . 112
13.6 Earth Falling into Sun (MAXIMA) . . . 114
13.7 Electron inside Bohr Radius (MAXIMA) . . . 117
13.8 On-Axis Potential of a Pair of Disks (MAXIMA) . . . 119
13.10 Properties of Cardioid (MAXIMA) . . . 122
13.11 Oscillator Matrix Elements (MAXIMA) . . . 125
13.12 Fourier Coefficients for Sawtooth (MAXIMA) . . . 130
13.13 Expansion in Legendre Polynomials (MAXIMA) . . . 132
13.14 Deduction of Simpson's Rule (MAXIMA) . . . 135

numerically with MAXIMA

13.20 Black Body Radiation (MAXIMA) . . . 137
13.21 Confidence Intervals for Gaussian (MAXIMA) . . . 139
13.23 Properties of Lorentz Distribution (MAXIMA) . . . 141

14 Finding Roots **143**

symbolically with MAXIMA

14.3 Finding Extrema in a Cubic Polynomial (MAXIMA) . . . 143
14.4 Roots of Quadratic Equation (MAXIMA) . . . 145
14.5 Lawnmower Blade Overlap (MAXIMA) . . . 146
14.8 Three Coupled Oscillators (MAXIMA) . . . 148

numerically with MAXIMA

14.18 Single-Slit Diffraction (MAXIMA) . . . 152
14.19 Roots of $J_0(x)$ (MAXIMA) . . . 154
14.20 Double-Welled Potential (MAXIMA) . . . 156

15 Introduction to Partial Differential Equations **159**
15.2 A Vertically Hanging String . . . 159
15.3 Second-Order ODEs Made Self-Adjoint . . . 163
15.7 FDM 1D Homogeneous Helmholtz Equation . . . 165
15.8 FEM 1D Homogeneous Helmholtz Equation . . . 168
15.9 1-D Inhomogeneous Helmholtz Equation . . . 174
15.12 Assembly for Three-Element Linear System . . . 178
15.20 FDM 2D Inhomogeneous Helmholtz Equation . . . 181
15.25 Finite Difference Formulas via Taylor Series . . . 182
15.26 Shape Functions for Two-Node Linear Element . . . 184

symbolically with MAXIMA

15.6 Better Approximation to First Derivative (MAXIMA) . . . 185
15.10 Shape Functions for Three-Node Linear Element (MAXIMA) . . . 187
15.11 Elemental Stiffness Matrix for Three-Node Linear Element (MAXIMA) . . . 192
15.15 Shape Functions for Triangular Element (MAXIMA) . . . 196
15.16 Shape Functions for Four-Node 2D Element (MAXIMA) . . . 201

Chapter 6

Introduction to MAXIMA

6.2 A Few MAXIMA Manipulations

Exercise: Use MAXIMA to convert each of the expressions in the left-hand column in the table below into the expression in the associated right-hand column:

$$\begin{array}{lccc} \text{(a)} & \dfrac{(a-x)^2}{(a^2-2ax+x^2)^{3/2}} & \Longrightarrow & \dfrac{1}{|x-a|} \\ \text{(b)} & \sinh(\,\ln(x+\sqrt{x^2+a^2})-\ln(a)\,) & \Longrightarrow & \dfrac{x}{a} \\ \text{(c)} & \dfrac{1}{x+\sqrt{y}} & \Longrightarrow & \dfrac{x-\sqrt{y}}{x^2-y} \end{array}$$

These samples are chosen to illustrate particularly the use of `radcan` and `ratsimp` but other commands will surely also be needed. Write two or three paragraphs in which you describe your efforts, including some indication of approaches that were *not* successful. Don't be overly concerned about the order of terms within various sets of parentheses; that order is particularly difficult to control. Focus instead on creating the general form of each desired result. *Hint*: For part (c), you might find it useful to look up the option variable `algebraic`.

Solution: (a) To address (a), we try the MAXIMA statements

```
(%i1)  func : (a-x)^2/((a^2-2*a*x+x^2)^(3/2));
```

$$\text{(func)} \qquad \frac{(a-x)^2}{(a^2-2\,a\,x+x^2)^{3/2}}$$

```
(%i2) radcan( func );
```

$$\text{(\%o2)} \qquad \frac{1}{x-a}$$

```
(%i3) ratsimp( func );
```

$$\text{(\%o3)} \qquad \frac{1}{\sqrt{x^2-2ax+a^2}}$$

but discover that `radcan` doesn't recognize that the result depends on whether $x > a$ or $x < a$ and `ratsimp` leaves the square root unevaluated, i.e., doesn't recognize that $x^2 - 2ax + a^2 = (x-a)^2$. We adopt a more elaborate procedure, specifically,

```
(%i4) n : num(func);
```

(n) $(a-x)^2$

```
(%i5) d : denom(func);
```

(d) $(a^2 - 2\,a\,x + x^2)^{3/2}$

```
(%i6) map( factor, d );
```

(%o6) $(x-a)^2|x-a|$

```
(%i7) func2 : factor(n/%);
```

(func2) $\dfrac{1}{|x-a|}$

which completes the transformation.

(b) To address (b), we invoke the MAXIMA statements

```
(%i8) tmp1 : sinh( log(x + sqrt(x^2 + a^2)) - log(a) );
```

(tmp1) $\sinh(\ln(x+\sqrt{x^2+a^2}) - \ln(a))$

```
(%i9) tmp2 : exponentialize( tmp1 );
```

(tmp2) $$\frac{\dfrac{\sqrt{x^2+a^2}+x}{a} - \dfrac{a}{\sqrt{x^2+a^2}+x}}{2}$$

```
(%i10) ratsimp(tmp2);
```

(%o10) $\dfrac{x}{a}$

```
(%i11) radcan(tmp2);
```

(%o11) $\dfrac{x}{a}$

`expand`, `simplify`, `radcan`, and `ratsimp` alone were not enough to get the desired form of the expression. The command `exponentialize` converts the `sinh` function to its exponential form and then recognizes that $e^{\ln(z} = z$. Then, both `radcan` and `ratsimp` complete the transformation.

(c) To address (c), we invoke the MAXIMA statements

```
(%i12) m : 1/(x+sqrt(y));
```

(m) $\dfrac{1}{\sqrt{y}+x}$

```
(%i13) ratsimp(m), algebraic: true;
```

(%o13) $\dfrac{\sqrt{y}-x}{y-x^2}$

```
(%i14) radcan(m), algebraic: true;
```

(%o14) $\dfrac{\sqrt{y}-x}{y-x^2}$

Here, both `ratsimp` and `radcan` achieve the desired result right away, though not without setting the keyword `algebraic` to `true`.

6.3 More MAXIMA Manipulations

Exercise: Use MAXIMA to convert each of the expressions in the left-hand column in the table below into the expression in the associated right-hand column:

(a)	$\frac{d}{dx}\left(x^2 e^{-x^2}\right)$	$\Longrightarrow$	$-2x(x^2-1)e^{-x^2}$
(b)	$\sin(\sqrt{a^2(a+3x)+x^2(3a+x)}+y)$	$\Longrightarrow$	$\sin((a+x)^{3/2}+y)$
(c)	$cg+cf+b^2d+2abd+a^2d+b^2c+2abc+a^2c$	$\Longrightarrow$	$(a+b)^2(c+d)+c(f+g)$
(d)	$a\,e^{(-b+i\omega)t}+a\,e^{(-b-i\omega)t}$	$\Longrightarrow$	$a\,e^{-bt}\,(e^{-i\omega t}+e^{i\omega t})$
(e)	$a\,e^{(-b+i\omega)t}+a\,e^{(-b-i\omega)t}$	$\Longrightarrow$	$2\,a\,e^{-bt}\cos(\omega t)$
(f)	$x^2+y^2+z^2-2a(x+y)+2a^2$	$\Longrightarrow$	$(x-a)^2+(y-a)^2+z^2$

In several cases, you may need to invoke `part`, `pickapart`, `subst`, and/or `substpart` but other commands will surely also be needed. Invoke microscopic dissection of the expressions only as a last resort. Write two or three paragraphs in which you describe your efforts, including some indication of approaches that were *not* successful. Don't be overly concerned about the order of terms within various sets of parentheses in the final form; that order is particularly difficult to control. Focus instead on creating the general form of each desired result.

(a) We begin by evaluating the derivative and factoring the expression to bring together the terms that multiply the exponential factor with the statements

```
(%i1) d : diff(x^2*exp(-x^2), x);
```

(%o1) $2xe^{-x^2}-2x^3e^{-x^2}$

```
(%i2) m : factor(d)
```

(m) $-2(x-1)x(x+1)e^{-x^2}$

We now need to multiply the two binomials without disturbing the remaining factors. Extraction of parts appears to be the only way to do this. Part 0 of `m` is the minus sign, while part 1 of `m` is the rest of the expression. Thus, we need to extract parts 2 and 4 of part 1 of `m`, multiply those two parts together and then substitute them in `m`. We invoke the statements

```
(%i3) a : part( part(m,1),2 ) * part( part(m,1),4 );
```

(a) $(x-1)(x+1)$

```
(%i4) b : expand(a);
```

(b) x^2-1

```
(%i5) m*b/a;
```

(%o5) $-2x(x^2-1)e^{-x^2}$

(b) The simplest route to the desired result is to begin with the quantity under the radical, factor it, take its square root, and then construct the desired expression with the statements

```
(%i6) tmp : a^2*(a+3*x) + x^2*(3*a+x);
```

(tmp) $a^2(a+3x)+x^2(3a+x)$

```
(%i7) tmp1 : factor(tmp);
```

(tmp1) $(x+a)^3$

```
(%i8) sin(sqrt(tmp1)+y);
```

(%o8) $\sin(y+(x+a)^{3/2})$

If we start with the given expression *in toto*,specifically

```
(%i9) f1 : sin(sqrt( tmp ) + y );
```

(f1) $\sin(y+\sqrt{a^2(3x+a)+x^2(x+3a)})$

we can't avoid a dissection of the expression. For example, the statements

```
(%i10) tmp : part(f1, 1,2,1);
```

(tmp) $a^2(3x+a)+x^2(x+3a)$

```
(%i11) substpart(factor(tmp), f1, 1,2,1);
```

(%o11) $\sin(y+(x+a)^{3/2})$

extract the factor under the square root, reconstruct it, and substitute it back into the original expression.

(c) The statements

```
(%i12) eq1 : c*g+c*f+b^2*d+2*a*b*d+a^2*d+b^2*c+2*a*b*c+a^2*c;
```

(eq1) $cg+cf+b^2d+2abd+a^2d+b^2c+2bc+a^2c$

```
(%i13) tmp1 : c*g + c*f
```

(tmp1) $cg+cf$

```
(%i14) tmp2 : eq1 - tmp1
```

(tmp2) $b^2d+2abd+a^2d+b^2c+2bc+a^2c$

```
(%i15) factor(tmp1) + factor(tmp2);
```

(%o15) $c(g+f)+(a+b)^2(d+c)$

peel off the two terms cg and cf, then fact5or each piece and reassemble the original expression. The command `factor` applied to the full expression achieves nothing because the entire expression has no single common factor.

For example, all of the elements except those involving f and g were extracted, added, and factored in one step. A similar (although less complicated) procedure was used for the remaining two elements. Then these two simplified sections were added to create the desired result

(d) For this expression, we first factor out the common factor a and then manipulate the expression with the statements

```
(%i16) eq1 : a*exp((-b+%i*omega)*t)+a*exp((-b-%i*omega)*t);
```

(eq1) $ae^{(i\omega-b)t}+ae^{(-i\omega-b)t}$

```
(%i17) collectterms(eq1,a);
```

(%o17) $a(e^{(i\omega-b)t}+ae^{(-i\omega-b)t})$

```
(%i18) a*exp(-b*t)*expand(exp(b*t)*part(%,2));
```

(%o18) $ae^{-bt}(e^{i\omega t} + e^{-i\omega t})$

to reach the desired final result.

(e) The `demoivre` command, which converts exponentials to trigonometric or hyperbolic functions is specifically built for this transformation. The statements

```
(%i19) demoivre(eq1);
```

(%o19) $æ^{-b*t}\big(i\sin(\omega) + \cos(\omega t)\big) + æ^{-b*t}\big(\cos(\omega t) - i\sin(\omega)\big)$

```
(%i20) expand(%);
```

(%o20) $2ae^{-bt}\cos(\omega t)$

had the same original expression as part (d), but we had to simplify it in a different way. Therefore, the first section of part (e) is identical to part (d). In fact, only two extra commands were necessary for `MAXIMA` to convert the expression into the desired form. First, we used the `demoivre` command to convert the imaginary expression into one involving trigonometric functions. Then, using `expand`, we eliminated terms that were only superficially present. Note that, had part (e) come before part (d), we could then have goneback to the desired exponential form with the single command

```
(%i21) exponentialize(%);
```

(%o21) $ae^{-bt}(e^{i\omega t} + e^{-i\omega t})$

(f) We create the initial expression with the statement

```
(%i22) eq3 : x^2+y^2+z^2-2*a*(x+y)+2*a^2;
```

(eq3) $z^2 + y^2 - 2a(y + x) + x^2 + 2a^2$

```
(%i23) eq3a : expand(eq3);
```

(eq3a) $z^2 + y^2 - 2ay + x^2 - 2ax + 2a^2$

The command `factor` is needed but it can't be applied to this expression as a whole. Instead, we have to extract the two pieces that can be factored and then factor them with the statements

```
(%i24) pt1 : part(eq3a,2) + part(eq3a,3) + part(eq3a,6)/2;
```

(pt1) $y^2 - 2ay + a^2$

```
(%i25) pt2 : part(eq3a,4) + part(eq3a,5) + part(eq3a,6)/2;
```

(pt1) $x^2 - 2ax + a^2$

```
(%i26) pt3 : factor( pt1 );
```

(pt3) $(y - a)^2$

```
(%i27) pt4 : factor( pt2 );
```

(pt4) $(x - a)^2$

Finally, we assemble the pieces to construct the desired result with the statement

```
(%i28) res : pt4 + pt3 + part(eq3a,1);
```

(res) $z^2 + (y - a)^2 + (x - a)^2$

6.4 Damped Harmonic Oscillation

Exercise: In Section 6.7.8, we set up the differential equation for a damped harmonic oscillator but then pursued the solution only for the underdamped case. Starting with the dimensionless form[1]

$$\frac{d^2x}{dt^2} + 2\alpha\,\frac{dx}{dt} + x = 0 \quad ; \quad x(0) = 1 \quad ; \quad \left.\frac{dx}{dt}\right|_{t=0} = 0$$

of the equation deduced towards the end of Section 6.7.8, work out the solution for the underdamped case and then find also the solutions for the critically damped and overdamped cases defined, respectively by

$$\alpha > 0 \quad \text{and} \quad \alpha <, =, > 1$$

Describe the differences in the physical behavior for the three cases. For simplicity, take the initial conditions to be $x(0) = 1$ and $v(0) = 0$.

Solution: The problem we wish to solve is defined by the differential equation and the initial conditions

$$\frac{d^2x}{dt^2} + 2\alpha\,\frac{dx}{dt} + x = 0 \quad ; \quad x(0) = 1 \quad ; \quad \left.\frac{dx}{dt}\right|_{t=0} = 0$$

We wish to consider the three cases

$$\alpha > 0 \quad \text{and} \quad \begin{cases} \alpha < 1 & \text{underdamped} \\ \alpha = 1 & \text{critically damped} \\ \alpha > 1 & \text{overdamped} \end{cases}$$

We define the equation in MAXIMA with the statement

```
(%i1) depends( x, t )$
```

```
(%i2) deq : 'diff(x,t,2)+2*alpha*'diff(x,t) + x = 0;
```

(deq) $\dfrac{d^2x}{dt^2} + 2\alpha\dfrac{dx}{dt} + x = 0$

Underdamped Case: For the underdamped case $\alpha < 1$, we solve the equation and simplify the result with the statements

```
(%i3) solnunder : ode2( deq, x, t );
```

Is $(\alpha - 1)(\alpha + 1)$ positive, negative or zero? n;

(solnunder) $x = e^{-\alpha t}\left(\%k1\,\sin\left(\dfrac{\sqrt{4-4\alpha^2}\,t}{2}\right) + \%k2\,\cos\left(\dfrac{\sqrt{4-4\alpha^2}\,t}{2}\right)\right)$

```
(%i4) assume( omega > 0 )$
```

```
(%i5) under : subst(4*omega^2, 4-4*alpha^2,solnunder);
```

(under) $x = e^{-\alpha t}\big(\%k1\,\sin(\omega t) + \%k2\,\cos(\omega t)\big)!$

```
(%i6) sund : ic2(under, t = 0, x=1, 'diff(x,t) = 0 );
```

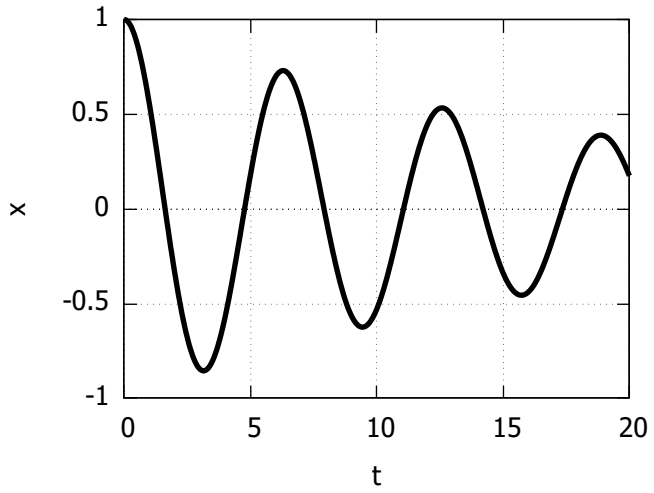
(sund) $x = e^{-\alpha t}\left(\dfrac{\alpha\,\sin(\omega t)}{\omega} + \cos(\omega t)\right)$

where $\omega = \sqrt{1-\alpha^2}$.

It is always prudent to verify a solution once it has been obtained. To check the initial conditions, we execute the statements

[1] The forms of the solution work out more easily if, in contrast to the choice in the text, we express the coefficient of the second term as *twice* a constant, so $\alpha = b/2m\omega_0$.

Figure E6.1: Underdamped motion for $\alpha = 0.05$ and $\omega = \sqrt{1-\alpha^2} = 0.99875$.



```
(%i7) ev(sund, t = 0);
```

(%o7) $x = 1$

```
(%i8) rhs( diff(sund,t) )$
(%i9) ev( %, t=0 );
```

(%o9) 0

and we confirm satisfaction of the equation with the statements[2]

```
(%i10) ev(deq,sund)$
(%i11) ev(%,diff)$
(%i12) expand(%/exp(-alpha*t))$
(%i13) subst(sqrt(1-omega^2),alpha, %)$
(%i14) expand(%)$
(%i15) factor(%);
(%o15)      0 = 0
```

This solution oscillates sinusoidally around the ultimate equilibrium position, sometimes above and sometimes below that position but also decreases exponentially in amplitude with each oscillation. The rate of oscillation is determined by ω and the rate of decay is determined by α. If, for example, $\alpha = 0.05$ and $\omega = \sqrt{1-\alpha^2} = 0.99875$, the statements

```
(%i16) alpha : 0.05$ omega : sqrt(1-alpha^2)$
(%i17) plot2d(rhs(sund), [t,0.0,20.0], [y,-1.0,1.0],
   [xlabel, "t"], [ylabel,"x"], [color,black], [grid2d,true],
   [style, [lines,4] ] )$
```

produce the graph in Fig. E6.1.

[2]Here and in the rest of this solution, intermediate output has sometimes been suppressed. In repeating this solution, you should replace terminating dollar signs with semicolons to view those intermediate results as a way to motivate each step.

Critically Damped Case: For the critically damped case in MAXIMA, we redefine the equation setting $\alpha = 1$ and then solve the equation and simplify the result with the statements

```
(%i18) deq1 : ev( deq, alpha = 1 );
```

(deq1) $\dfrac{d^2x}{dt^2} + 2\dfrac{dx}{dt} + x = 0$

```
(%i19) solncrit : ode2( deq1, x, t );
```

(solncrit) $x = (\%k2\, t + \%k1)\, e^{-t}$

```
(%i20) scrit : ic2(solncrit, t = 0, x=1, 'diff(x,t) = 0 );
```

(scrit) $x = (t+1)\, e^{-t}$

It is always prudent to verify a solution once it has been obtained. To check the initial conditions, we execute the statements

```
(%i21) ev(scrit, t = 0);
(%o21)     x = 1
(%i22) rhs( diff(scrit,t) )$
(%i23) ev( %, t=0 );
(%o23)     0
```

and we confirm satisfaction of the equation with the statements

```
(%i24) ev(deq1,scrit)$
(%i25) ev(%,diff)$
(%i26) expand(%/exp(-t));
(%o26)      0 = 0
```

Except for the first short interval, in which the solution starts off with a horizontal tangent, this solution is essentially exponentially decaying from the initial position. In contrast to the under damped case, the solution approaches the equilibrium position monotonically, never crossing that position as it moves towards it. Indeed, in the dimensionless casting, there are no parameters in the solution, a graph of which is obtained with the statements

```
(%i27) plot2d(rhs(scrit), [t,0.0,10.0], [y,0.0,1.0],
   [xlabel, "t"], [ylabel,"x"], [color,black], [grid2d,true],
   [style, [lines,4] ] )$
```

produce the graph in Fig. E6.2.

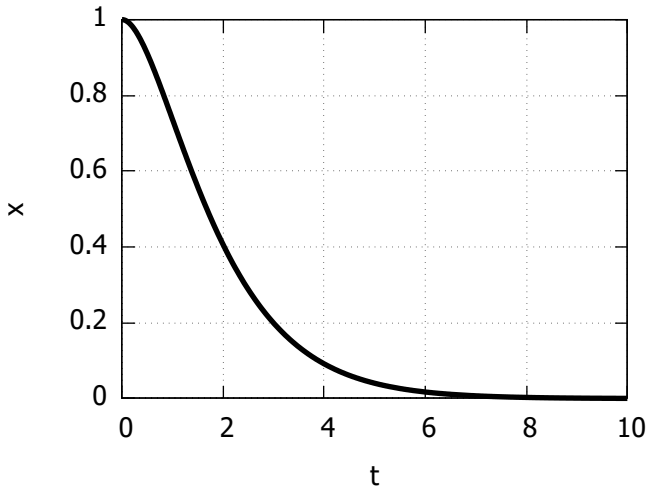
Overdamped Case: For the overdamped case $\alpha > 1$, we clear the previous assumptions and values on α and ω, assume $\alpha > 1$, and solve the equation and simplify the result with the statements

```
(%i28) remvalue( alpha, omega )$
(%i29) assume( alpha > 1 )$
(%i30) solnover : ode2( deq, x, t );
```

(solnover) $x = \%k1\, e^{(\sqrt{4\alpha^2-4}-2\alpha)\, t/2} + \%k2\, e^{(-\sqrt{4\alpha^2-4}-2\alpha)t/2}$

```
(%i31)
```

Figure E6.2: Critically damped motion.



```
(%i32) sover1 : subst( 4*omega^2, 4*alpha^2-4,solnover );
```

(sover1) $x = \%k1\, e^{(2\omega-2\alpha)t/2} + \%k2\, e^{(-2\omega-2\alpha)t/2}$

```
(%i33) sover2 : expand( sover1 );
```

(sover2) $x = \%k1\, e^{\omega t-\alpha t} + \%k2\, e^{-\omega t-\alpha t}$

```
(%i34) sover3 : subst( (omega-alpha)*t, omega*t-alpha*t, sover2);
```

sover3 $x = \%k2\, e^{-\omega t-\alpha t} + \%k1\, e^{(\omega-\alpha)t}$

```
(%i35) sover4 : subst(-(omega+alpha)*t, -omega*t - alpha*t, sover3);
```

sover4 $x = \%k1\, e^{(\omega-\alpha)t} + \%k2\, e^{(-\omega-\alpha)t}$

```
(%i36) sover : ic2(sover4, t = 0, x=1, 'diff(x,t) = 0 );
```

(sover) $x = \dfrac{(\omega+\alpha)\, e^{(\omega-\alpha)\, t}}{2\omega} + \dfrac{(\omega-\alpha)\, e^{(-\omega-\alpha)\, t}}{2\omega}$

where, this time, $\omega^2 = \alpha^2 - 1$.

It is always prudent to verify a solution once it has been obtained. To check the initial conditions, we execute the statements

```
(%i37) expand( ev(sover, t = 0) );
```

(%o38) $x = 1$

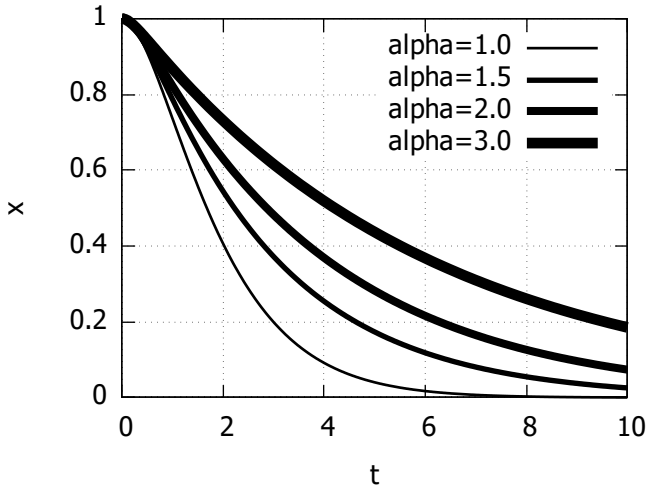
```
(%i39) rhs( diff(sover,t) )$
(%i40) expand( ev( %, t=0 ) );
```

(%o40) 0

and we confirm satisfaction of the equation with the statements

```
(%i41) ev(deq,sover)$
(%i42) ev(%,diff)$
```

Figure E6.3: Critically damped motion for $\alpha = 1.0$ and overdamped motion for for $\alpha = 1.5, 2.0, 3.0$ and $\omega = \sqrt{\alpha^2 - 1}$.




```
(%i43) expand(%/exp(-alpha*t))$
(%i44) ratsimp( subst(sqrt(omega^2+1),alpha, %) );
(%o44)      0 = 0
```

This solution in the variable `sover` consists of a superposition of two exponential terms with the factors

$$e^{(\omega-\alpha)t} = e^{(\sqrt{\alpha^2-1}-\alpha)t} \quad \text{and} e^{(-\omega-\alpha)t} = e^{(-\sqrt{\alpha^2-1}-\alpha)t}$$

Since $\sqrt{\alpha^2 - 1} < \alpha$,the first of these terms decays exponentially fairly slowly while the second term decays rather more rapidly. In the short term, both terms are significant; in the longer term, the more slowly decaying term dominates. If, for example,[3] $\alpha = 1.5, 2.0$ and 3.0, and $\omega = \sqrt{\alpha^2 - 1}$, the statements

```
(%i45) alpha : 1.5$ omega : sqrt(alpha^2-1)$
(%i46) f15 : ev( rhs(sover) )$
(%i47) alpha : 2.0$ omega : sqrt(alpha^2-1)$
(%i48) f20 : ev( rhs(sover) )$
(%i47) alpha : 3.0$ omega : sqrt(alpha^2-1)$
(%i48) f30 : ev( rhs(sover) )$
(%i49) plot2d([rhs(scrit),f15,f20,f30], [t,0.0,10.0], [y,0.0,1.0],
   [xlabel, "t"], [ylabel,"x"], [color,black], [grid2d,true],
   [style, [lines,2], [lines,4],[lines,6],[lines,8]],
   [legend, "alpha=1.0", "alpha=1.5", "alpha=2.0", "alpha=3.0"] )$
```

produce the graph in Fig. E6.3. For comparison, I have included the critically damped case simply copied from the solution obtained above for that case.

An Aside: The standard approach to the analytic solution of the differential equation

$$\frac{d^2x}{dt^2} + 2\alpha\,\frac{dx}{dt} + x = 0 \quad ; \quad x(0) = 1$$

[3] Remember $\alpha > 1$ in the overdamped case.

involves assuming a solution of the form $x(t) = e^{pt}$. Substituting this form into the equation yields the characteristic algebraic equation

$$p^2 + 2\alpha p + 1 = 1$$

which is a quadratic equation whose solutions are

$$p = \frac{-2\alpha \pm \sqrt{4\alpha^2 - 1}}{x} 2 = -\alpha \pm \sqrt{\alpha^2 - 1}$$

Clearly, the size of α has a significant influence on the character of the solutions. If $\alpha < 1$, the quantity under the square root is negative and the two roots are complex and are perhaps better written in the form

$$p = -\alpha \pm i\omega \quad \text{where} \quad \omega = \sqrt{1 - alpha^2}$$

In this case, with A and B undetermined constants (to be determined by the initial conditions), the solution

$$x(t) = Ae^{-\alpha + i\omega)t} + Be^{-\alpha - i\omega)t} = e^{\alpha t}\Big(Ae^{i\omega t} + Be^{-i\omega t}\Big)$$

decays exponentially while oscillating sinusoidally. This is the underdamped case.

In the overdamped case $\alpha > 1$, the quantity under the square root in p is positive and, because $\sqrt{\alpha^2 - 1} < \alpha$, both values of p are negative, yielding a solution that is a superposition of two exponential terms with different decay rates.

Finally, in the critically damped case, $\alpha = 1$, the square root in p is zero, and the characteristic equation has a double root, perhaps more easily seen in the factorization $p^2 + 2p + 1 = (p + 1)^2$. In this case, the general solution must be written in the form $x(t) = (At + B)e^{-t}$.

6.5 Legendre Polynomials

Exercise: The Legendre polynomials $P_n(x)$, which are valid and useful over the interval $-1 \leq x \leq 1$, can be defined in many ways. They emerge as the coefficients in the Taylor expansion of the generating function

$$g(x,t) = \frac{1}{\sqrt{1-2xt+t^2}} = \sum_{n=0}^{\infty} P_n(x)\, t^n$$

Alternatively, they can be determined from the recursion relationship

$$(2n+1)\, x\, P_n(x) = (n+1)\, P_{n+1}(x) + n\, P_{n-1}(x)$$

provided we include the first two $P_0(x) = 1$ and $P_1(x) = x$ to get started. Yet again, they can be found from application of multiple differentiation as implied by Rodrigues' formula

$$P_n(x) = \frac{1}{2^n\, n!} \frac{d^n}{dx^n}\Big((x^2-1)^n\Big)$$

However they are determined, the first half dozen of these polynomials will turn out to be

$$\begin{aligned} P_0(x) &= 1 & \qquad P_3(x) &= \tfrac{1}{2}(5x^3 - 3x) \\ P_1(x) &= x & \qquad P_4(x) &= \tfrac{1}{8}(35x^4 - 30x^2 + 3) \\ P_2(x) &= \tfrac{1}{2}(3x^2 - 1) & \qquad P_5(x) &= \tfrac{1}{8}(63x^5 - 70x^3 + 15x) \end{aligned}$$

a. Use the generating function and MAXIMA's capabilities for evaluating Taylor series to find the first half-dozen Legendre polynomials, extracting each as an expression bound to a variable.

b. Start by binding the value 1 to P[0] and the value x to P[1] and then, using the recursion relationship, find the next several Legendre polynomials. *Hint*: The MAXIMA statements would be

```
(%i1) P[0] : 1;
(%i2) P[1] : x;
```

and then

```
(%i3) P[n] := (2*n-1)*x*P[n-1]/n - (n-1)*P[n-2]/n;
```

(Verify the expression on the right by using MAXIMA to deduce this relationship from the standard form—the second equation in this exercise.) With these statements, you have set up $P_0(x)$ and $P_1(x)$ to start the recursion and then you have defined a *function* `P[n]` involving n that can be evaluated at any n. Once P_0 and P_1 have been defined, you can find P_2 and then P_3 and then ... with statements like

```
(%i4) P[2];
(%i5) P[3];
...
```

(Some simplifications may be necessary to cast the results in the standard form of the above table.)

c. Find the first half-dozen Legendre polynomials by using the command `diff` to evaluate Rodrigues' formula. *Hint*: You might find that using a loop would simplify your approach.

d. Be clever and, using either matrices or loops constructed in MAXIMA (see manuals for the structure of a `for` loop), find the values of *all* of the integrals

$$\int_{-1}^{1} P_n(x)\, P_m(x)\, dx$$

where n and m take on independently the values 0, 1, 2, 3, 4, 5. (There are 36 integrals to be evaluated. Try to be efficient in your coding.)

e. Within MAXIMA, obtain graphs of the first six Legendre polynomials over the interval $-1 \leq x \leq 1$.

f. It is known that a function $f(x)$ defined over the interval $-1 \leq x \leq 1$ can be expanded in a series of Legendre polynomials of the form $f(x) = \sum a_n\, P_n(x)$ where the coefficients a_n are given by

$$a_n = \frac{2n+1}{2} \int_{-1}^{1} f(x)\, P_n(x)\, dx$$

Find the first six coefficients for the expansion of the function $f(x) = 0$ when $-1 < x < 0$ and $f(x) = 1$ when $0 < x < 1$. Then, construct the (partial) series representing this function and obtain a graph of that approximation to compare with the graph of the original function.

g. MAXIMA actually knows quite a bit about many of the important special functions of mathematical physics. In particular, the package `orthopoly`, which can be loaded with the command `load( orthopoly )` and which is described fully in the MAXIMA manuals, provides two different ways to evaluate a very large number of these special functions. Take a look at that documentation and use the function `legendre_p` (syntax `legendre_p(n, x)`) in that package to determine the first several Legendre polynomials.

Solution: (a) One quick way to find the Legendre polynomials is through the Taylor expansion of the generating function,

$$g(x,t) = \frac{1}{\sqrt{1 - 2xt + t^2}} = \sum_{n=0}^{\infty} P_n(x)\, t^n$$

The Legendre polynomials are the coefficients of powers of t in this expansion. To find these coefficients using MAXIMA, we first define the generating function with the statement

```
(%i1) g : 1/sqrt(1 - 2*x*t + t^2);
```

(%o1) $\dfrac{1}{\sqrt{-2tx + t^2 + 1}}$

Next, we create the first several terms in the Taylor series expansion of the generating function by executing the statement

```
(%i2) legexp = taylor( g, t, 0, 5 );
```

(%o2)
/T/ $\dfrac{1}{8}(63x^5 + (-70)x^3 + 15x)\, t^5 + \dfrac{1}{8}(35x^4 + (-30)x^2 + 3)\, t^4 + \dfrac{1}{2}(5x^3 + (-3)x)\, t^2 + xt + 1 + \ldots$

The second argument, here t, to the `taylor` function identifies the variable to be expanded around, the third argument, here 0, identifies the point in t to expand about, and the fourth argument, here 5, stipulates the highest power of t to be included in the expansion. The Legendre polynomials $L_n(x), 0 \leq n \leq 5$ are evident in the coefficients of the several powers of t in this expansion.

A somewhat more transparent presentation, one from which the separate polynomials can be more easily extracted, follows if we collect the terms with the statement[4]

```
(%i3) legexpa :  collectterms( legexp, t );
```

$$\text{(\%o3)} \quad 1 + x\,t + \frac{3x^2-1}{2}\,t^2 + \frac{5x^3-3x}{2}\,t^3 + \frac{35x^4-30x^2+3}{8}\,t^4 + \frac{63x^5-70x^3+15x}{8}\,t^5$$

In particular, this rearrangement has converted the expression from a Taylor expansion into an ordinary polynomial. Finally, we invoke MAXIMA's `coeff` function to extract the coefficients from the polynomial with the loop

```
(%i4) for i from 0 thru 5 do P[i] : coeff( legexpa, t^i )$
```

thereby assigning the i-th polynomial to the i-th element of the array P. The polynomials can be individually accessed or displayed with statements like

```
(%i5) P[0];
(%o5)   1
(%i6) P[1];
(%o6)   x
```

$\vdots$

```
(%i10) P[5];
```

$$\text{(\%o10)} \quad \frac{63x^5-70x^3+15x}{8}$$

(b) Another way to find the Legendre polynomials is by using the recursion relationship

$$(2n+1)xP_n(x) = (n+1)P_{n+1}(x) + nP_{n-1}(x)$$

First, we must recast this relationship as an expression giving a particular Legendre polynomial in terms of the previous two, i.e., in an expression of the form

$$P_n(x) = A\,P_{n-1}(x) + B\,P_{n-2}(x)$$

where A and B potentially depend on n and x but not on the Legendre polynomials.

The statement in the exercise contends that the expression

$$P_n(x) = \frac{(2n-1)xP_{n-1}(x)}{n} - \frac{(n-1)P_{n-2}(x)}{n}$$

is a rewriting of the recursion relationship. We now prove this relationship. Accepting the initial recursion relationship, we first translate the indices so that the Legendre polynomial with the highest index is labeled P_n and then solve that result for P_n. We start by reinitializing MAXIMA and defining the recursion relationship with the statements

```
(%i11) kill(all)$ reset(all)$
(%i2) P[n] := (2*n-1)*x*P[n-1]/n - (n-1)*P[n-2]/n$
```

Now we set P_0 and P_1 and then use this newly defined recursion relationship to find the remaining polynomials with the statements

[4] We have here taken the liberty of reversing the order of the terms presented by MAXIMA and also of displaying the powers of t separately from the coefficients.

```
(%i3) P[0] :  1;
(%o3)  1
(%i4) P[1] :  x;
(%o4)  1
(%i5) P[2];
```

$$(\%o5) \quad \frac{3x^2}{2} - \frac{1}{2}$$

```
(%i6) expand(P[3]);
```

$$(\%o6) \quad \frac{5x^3}{2} - \frac{3x}{2}$$

```
(%i6) expand(P[4]);
```

$$(\%o6) \quad \frac{35x^4}{8} - \frac{15x^2}{4} + \frac{3}{8}$$

```
(%i7) expand(P[5]);
```

$$(\%o7) \quad \frac{63x^5}{8} - \frac{35x^3}{4} + \frac{15x}{8}$$

(c) The Legendre polynomials can also be defined with Rodrigues' formula. First, we enter Rodrigues' formula using the deferred differention `'diff` with the statement

```
(%i11) kill(all)$ reset(all)$
(%i2) Rod : 1/(2^n * n!)*'diff((x^2 - 1)^n, x, n);
```

$$(\%o2) \quad \frac{1}{2^n\, n!} \frac{d^n}{dx^n}(x^2-1)^n$$

The second argument of the function `diff` contains the variable, x, with respect to which the derivative is taken and the third argument specifies the number of times, n, that the differentiation is carried out. Now, we can find the Legendre polynomials by substituting different values of n into `Rod`. A single statement to effect this action and display the results is the loop

```
(%i3) for i from 0 thru 5 do (P[i] : expand( ev( Rod, n=i, diff) ), print( "P[", i, "] : ", P[i] ) );
```

$P[0] : 1$

$P[1] : x$

$$P[2] : \frac{3x^2}{2} - \frac{1}{2}$$

$$P[3] : \frac{5x^3}{2} - \frac{3x}{2}$$

$$P[4] : \frac{35x^4}{8} - \frac{15x^2}{4} + \frac{3}{8}$$

$$P[5] : \frac{63x^5}{2} - \frac{35x^3}{4} + \frac{15x}{8}$$

```
(%o3)  done
```

In this loop, we have used parentheses to block together two statements that comprise the body of the for-thru-do loop.

(d) To evaluate the requested integrals (and demonstrate the orthogonality of the Legendre polynomials with weight 1 on the interval $-1 \le x \le 1$), we could create a matrix containing the various integrands and then use the function `integrate` to integrate each element in that matrix. At

first, the task may look daunting. Luckily, MAXIMA provides straightforward tools to deal with this problem. We begin by creating a list of values for the first six Legendre polynomials as were determined in (a), (b), and (c).

We use the values for P[n] found in part (c) to create the matrix of products of $P_n\,P_m$. To create the matrix, we create a list of the polynomials, transpose the list, and then multiply the column list into the row list with the statements

```
(%i4) lst1 : makelist( P[i], i, 0, 5 )$
(%i5) lst2 : transpose( lst1 )$
(%i6) mat : lst2.lst1;
```

(%o6)
$$\begin{pmatrix} 1 & x & \frac{3x^2}{2}-\frac{1}{2} & \cdots \\ x & x^2 & x\left(\frac{3x^2}{2}-\frac{1}{2}\right) & \cdots \\ \frac{3x^2}{2}-\frac{1}{2} & x\left(\frac{3x^2}{2}-\frac{1}{2}\right) & \left(\frac{3x^2}{2}-\frac{1}{2}\right) & \cdots \\ \vdots & \vdots & \vdots & \vdots \end{pmatrix}$$

We now use the `integrate` function to integrate every element in the matrix with the statement

```
(%i7) integrate( mat,  x, -1, 1 ]
```

(%o7)
$$\begin{pmatrix} 2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 2/3 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2/5 & 0 & 0 & 0 \\ 0 & 0 & 0 & 2/7 & 0 & 0 \\ 0 & 0 & 0 & 0 & 2/9 & 0 \\ 0 & 0 & 0 & 0 & 0 & 2/11 \end{pmatrix}$$

Evidently, these integrals are 0 for all values of n and m except when $n = m$. Further, when $n = m$, it appears as if the value of the integral is $2/(2n + 1)$. Thus, we apparently can infer that

$$\int_{-1}^{1} P_n(x)\,P_m(x)\,dx = \frac{2}{2n+1}\,\delta_{nm}$$

(Here, δ_{nm}, the Kronecker delta, is 0 when $n \neq m$ and 1 when $n = m$.) This result is correct in general, though our evidence supports its correctness only when $0 \leq n, m \leq 5$.

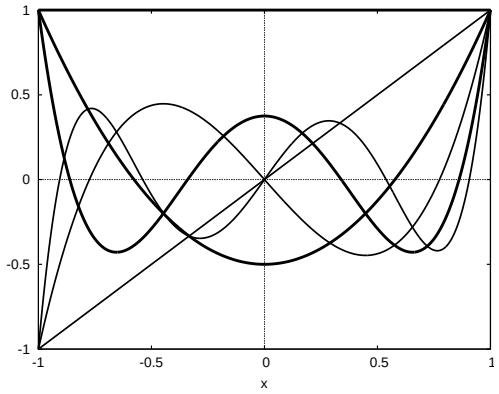
We can understand some of the zeroes in the above matrix by noting the parity of the Legendre polynomials: those for even n are even functions of x, those for odd n are odd functions of x. Thus, whenever we integrate a product of two polynomials, one of which is even and the other of which is odd, on symmetric limits, the integral (which is then an integral of an odd function on symmetric limits) will be zero. This parity argument, however, doesn't explain why the integration yields 0 when we integrate the product of two *different* even or two *different* odd polynomials, in which case the integrand is an *even* function. There appears to be something deeper at work that depends on 1 and -1 being the limits of integration. Let's evaluate the integral

$$\int_{-a}^{a} P_n(x)\,P_m(x)\,dx$$

with the statements

```
(%i8) integrate( mat, x, -a, a )$
(%i8) factor[%];
```

Figure E6.4: The First Six Legendre Polynomials. Note that $P_0(-1), P_2(-1)$, and $P_4(-1)$ have the value $+1$ and those functions are shown with heavier lines; $P_1(-1), P_3(-1)$, and $P_5(-1)$ have the value -1 and those functions are shown with lighter lines.



We suppress the results, but note that all entries when n and m have opposite parity are still zero. Those entries when n and m have the same parity (but are not equal) turn out to have a factor of $a - 1$, which will reduce those terms to zero when $a = 1$. Only the diagonal terms survive when a is set to the value 1.[5]

(e) To see what the Legendre polynomials look like graphically, we use the `plot2d` function in the statement

```
(%i9) plot2d( lst1, [x, -1.0, 1.0], [legend, false], [color, black],
      [style, [lines,6], [lines,4] ] );
```

The graph created with this command is included in Fig. E6.4. Notice that the functions are odd if n is odd and even if n is even. Also note that all the graphs start at $y = \pm 1$ and end at $y = +1$, i.e., that $P_n(-1) = (-1)^n$ and $P_n(1) = 1$.

(f) This section, the most involved of the exercise, will be addressed in three steps. First, we will evaluate the coefficients a_n for the given function.. Then we will combine these coefficients with the Legendre polynomials and evaluate the summation to form a polynomial approximation to our function. Finally, we will plot a graph of $f(x)$ created using the first six Legendre polynomials and compare it to the graph for the given values of $f(x)$.

First, we need to evaluate the coefficients a_n. Since $f(x) = 0$ in $-1 \le x \le 0$ and $f(x) = 1$ in $0 < x \le 1$, the defining integral reduces to the integral

$$a_n = \frac{2n+1}{2} \int_{-1}^{1} f(x)\, P_n(x)\, dx = \frac{2n+1}{2} \int_{0}^{1} P_n(x)\, dx \quad ; \quad n = 0, 1, 2, \ldots$$

Then, we perform the integration. We use the Legendre polynomials from part (c) to accomplish this integration. We integrate the list `lst1` already created with the statement

```
(%i10) lst2 : integrate( lst1, x, 0, 1 );
```

(%o10) $\left[1, \frac{1}{2}, 0, -\frac{1}{8}, 0, \frac{1}{16}\right]$

[5] We could also explore this issue by looking at the symmetry of a product of two Legendre polynomials of the same parity on the interval $0 \le x \le 1$ with respect to the point $x = 1/2$, but such an examination would entail invoking properties of the Legendre polynomials not assumed as background for this exercise.

After creating this list, we create a second list which evaluates $(2n+1)/2$ for n from zero to five with the statement

```
(%i11) lst3 :  makelist( (2*n+1)/2, n, 0, 5 );
```

(%o11) $$\left[\frac{1}{2},\frac{3}{2},\frac{5}{2},\frac{7}{2},\frac{9}{2},\frac{11}{2}\right]$$

Each element in this list corresponds to the element with the same n.

Now that we have the integral and $(2n+1)/2$ for each desired value of n, we multiply the corresponding elements to get a_n. We then multiply this product by $P_n(x)$ to get one piece of $f(x)$. Then, to get the partial sum, we add up all of these individual pieces of $f(x)$. All of this can be accomplished with a single statement that sums the products of the individual elements in our three lists. This statement is

```
(%i12) fct :  sum( lst3[i]*lst2[i]*lst1[i], i, 1, 6 );
```

(%o12) $$\frac{11\left(\frac{63x^5}{8}-\frac{35x^3}{4}+\frac{15x}{8}\right)}{32}-\frac{7\left(\frac{5x^3}{2}-\frac{3x}{2}\right)}{16}+\frac{3x}{4}+\frac{1}{2}$$

Now we simplify the answer with the `expand` functions in the statement

```
(%i13) fct1 : expand( fct );
```

(%o13) $$\frac{693\,x^5}{256}-\frac{525\,x^3}{128}+\frac{525\,x}{256}+\frac{1}{2}$$

Finally, we create the original function and then graph both that function and the approximation. We create the original step function with an `if` statement and then graph the two with the `plot2d` function. These statements are

```
(%i14) g(x) := if x < 0 then g : 0 else g : 1;
(%i15) plot2d( [fct1, g(x)], [x, -1.0, 1.0], [legend, false], [color, black],
[style, [lines, 4], [lines, 2] ] );
```

The resulting graph is shown in Fig. E6.5. Notice that, when carried out to the fifth order, the result is not as closely matching as one might expect. However, it is evident that over the region, the graph is trying to approximate that of the step function. A look at Fig. E6.6, which carries the expansion to tenth order, confirms this expectation. In the limit as the order approaches infinity, the Legendre approximation of $f(x)$ becomes exact and the plot will precisely match the defined function over the region.

(g) Since MAXIMA knows many of the important functions of mathematical physics, we can easily find the first six Legendre polynomials by executing the statement

```
(%i16) for i from 0 thru 5 do print( expand(legendre_p(i, x)) );
```

We do not print the list since we already know the answer.

Figure E6.5: Step function and its Legendre polynomial approximation using the first six Legendre polynomials.

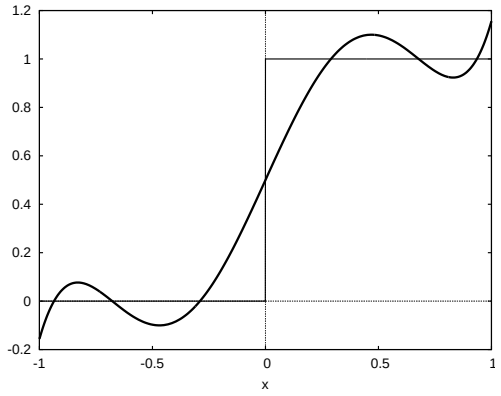
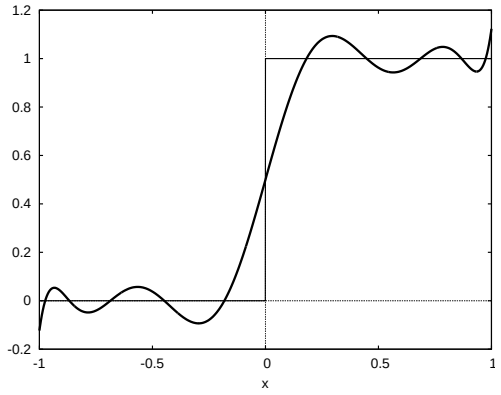


Figure E6.6: Step function and its Legendre polynomial approximation using the first eleven Legendre polynomials.



6.6 Wheatstone Bridge

Exercise: Figure E6.7 shows the circuit diagram for a Wheatstone bridge. Using Kirchoff's laws, set up the equations from which you could determine the currents in each branch of the circuit. Then, using MAXIMA, (a) solve the equations symbolically, (b) find conditions under which the current in the cross branch (through resistor R_5) will be zero, and (c) find the effective resistance seen by the battery. (The effective resistance is defined by the ratio V/I, where I is the current in the branch containing the battery.)

Solution: (a) To use Kirchoff's laws on the Wheatstone bridge, we set up the diagram explaining the currents and their directions as represented in Fig. E6.8.

With this definition, we can write six independent equations using Kirchoff's laws—three for the currents at nodes of the circuit and three for the voltages around loops of the circuit. We define the equations in MAXIMA with the statements:

Figure E6.7: Circuit for Wheatstone Bridge as described in Exercise 6.6.

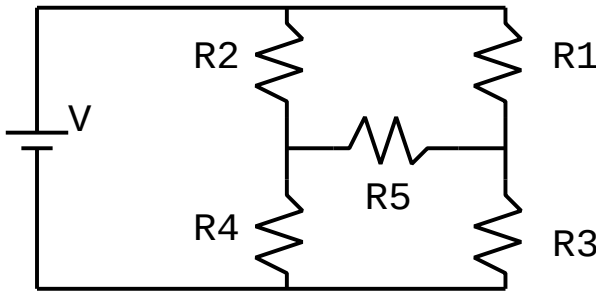
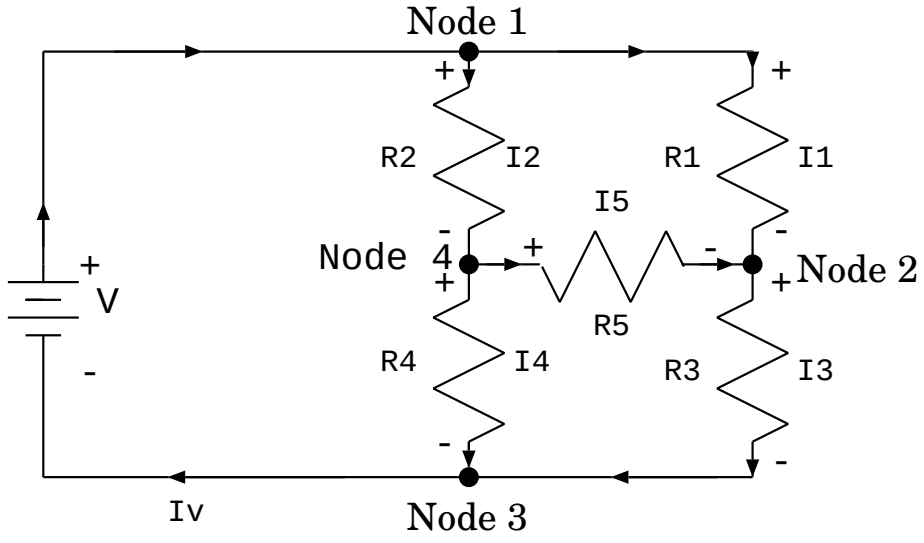


Figure E6.8: Wheatstone Bridge circuit with arbitrary current directions labeled.



```
(%i1)  IEqn1 : Iv - I2 - I1 = 0$
```
Set net current into Node 1 = 0.

```
(%i2)  IEqn2 : I1 + I5 - I3 = 0$
```
Set net current into Node 2 = 0.

```
(%i3)  IEqn3 : I3 + I4 - Iv = 0$
```
Set net current into Node 3 = 0.

```
(%i4)  VolEqn1 : V - I1*R1 - I3*R3 = 0$
```
Set voltage drop clockwise around loop containing V, R_1, and $R_3 = 0$.

```
(%i5)  VolEqn2 : V - I2*R2 - I4*R4 = 0$
```
Set voltage drop clockwise around loop containing V, R_2, and $R_4 = 0$.

```
(%i6) VolEqn3 : V - I2*R2 - I5*R5 -
         I3*R3 = 0$
```
Set voltage drop clockwise around loop containing V, R_2, R_5, and $R_3 = 0$.

We solve these (linear) equations for the unknown currents with the statement

```
(%i7) eqlst : [IEqn1, IEqn2, IEqn3, VolEqn1, VolEqn2, VolEqn3]$
(%i8) varlst : [I1, I2, I3, I4, I5, Iv]$
(%i9) soln : factor(  solve( eqlst, varlst ) );
```

Here, the function `solve` solves a system of linear equations for unknowns as long as the number of unknowns is equal to or less than the number of equations (and the equations are linearly independent) and the function `factor` separately simplifies the denominators and the numerators in the solution for each variable. We suppress display of this solution because it is a bit involved. You should, however, look at it and, in particular, note that the solutions for the six currents all share the same (complicated) denominator. To simplify the appearance of these solutions, let's substitute a single symbol for that denominator. We extract the denominator with the statement

```
(%i10) den :  denom( ev( I1, soln ) );
```

(%o10) $R_3R_4R_5 + R_1R_4R_5 + R_2R_3R_5 + R_1R_2R_5 + R_2R_3R_4 + R_1R_3R_4 + R_1R_2R_4 + R_1R_2R_3$

Then, we replace this denominator in `soln` with the single symbol `D` with the statement

```
(%i11) solna : factor( subst( D, den, soln ) );
```

%o11)
$$\left[\left[I_1 = \frac{(R_4R_5 + R_2R_5 + R_2R_4 + R_2R_3)V}{D}, I_2 = \frac{(R_3R_5 + R_1R_5 + R_1R_4 + R_1R_3)V}{D},\right.\right.$$
$$I_3 = \frac{(R_4R_5 + R_2R_5 + R_2R_4 + R_1R_4)V}{D}, I_4 = \frac{(R_3R_5 + R_1R_5 + R_2R_3 + R_1R_3)V}{D},$$
$$I_5 = \frac{(R_1R_4 - R_2R_3)V}{D},$$
$$\left.\left. I_v = \frac{(R_4R_5 + R_3R_5 + R_2R_5 + R_1R_5 + R_2R_4 + R_1R_4 + R_2R_3 + R_1R_3)V}{D}\right]\right]$$

(b) In particular, we can readily see from the solution for I_5 that, if $R_3R_2 = R_4R_1$, the current I_5 will be zero. Using MAXIMA explicitly, we can find the condition on R_1 under which I_5 will be zero with the statement

```
(%i12) cond = solve( ev( I5=0, solna ), R1 )
```

(%o12) $\left[R_1 = \frac{R_2R_3}{R_4}\right]$

We see that as long as R_1 is equal to R_2R_3/R_4 there will be no current through the bridge and we are free to choose R_5 to be anything without affecting the circuit. This relationship provides the

basis for the primary use of the Wheatstone bridge. Suppose R_1 is an unknown resistor and R_2, R_3, and R_4 can be adjusted. Suppose, further, that a galvanometer is placed in series with R_5. Then, by adjusting R_2, R_3, and R_4, we can *balance* the bridge so that the galvanometer shows no current. Finally, we can use the balance condition here deduced to calculate the value of R_1 from the known values of R_2, R_3, and R_4. Before the days of high-quality ohmmeters, the Wheatstone bridge was an essential piece of laboratory equipment.

(c) Finally, to find the effective resistance, we divide the current I_v passing through the branch with the battery by the voltage V of the battery. Since we have already solved for I_v the task involves the single statement

```
(%i13) effre = V / ev( Iv, solna )
```

$$\text{(\%o13)} \quad \frac{D}{R_4R_5 + R_3R_5 + R_2R_5 + R_1R_5 + R_2R_4 + R_1R_4 + R_2R_3 + R_1R_3}$$

To facilitate exploring this result, we insert the full value of `D` with the statement

```
(%i14) effrea : ev(effre, D = den )$
```

though we suppress display of the full result. With this result, however, the statement

```
(%i15) factor( ev( effrea, [R2=R1,R3=R1, R4=R1] );
```

(%o15) R_1

reveals that, if we choose $R_1 = R_2 = R_3 = R_4$, the effective resistance will be R_1. Further, if this is done, note that $R_1 = R_2R_3/R_4$ and, therefore, we are free to choose R_5, i.e., that I_v is zero regardless of the value of R_5. These two items are confirmed in MAXIMA through substitution. Confirmation of this assertion is provided by the MAXIMA statement

```
(%i16) factor( ev( effrea [R1=R, R2=R, R3=R, R4=R] ) )
```

(%o16) R

Here, R_5 does not appear in the effective resistance even though no value was given to that resistor. Changing R_5 in this situation does not change the effective resistance seen by the battery.

6.7 Quantum Wave Scattering

Exercise: In the scattering of a quantum wave from a rectangular barrier in particular circumstances, we find that the wave function must be expressed in three pieces in the form

$$\begin{aligned} \psi(x) &= Ae^{ikx} + Be^{-ikx} && x < 0 \\ &= D\cosh(\kappa x) + F\sinh(\kappa x) && 0 < x < w \\ &= Ce^{ikx} && x > w \end{aligned}$$

where k and κ are constants related to the energy of the particle and the height of the barrier, A is a constant reflecting the intensity of the incident beam, and B, D, F, and C are constants to be determined by imposing the requirement that the wave function *and* its first derivative be continuous both at $x = 0$ and at $x = w$, i.e., that

$$\psi(0^-) = \psi(0^+) \quad ; \quad \psi(w^-) = \psi(w^+) \quad ; \quad \left.\frac{d\psi}{dx}\right|_{0^-} = \left.\frac{d\psi}{dx}\right|_{0^+} \quad ; \quad \left.\frac{d\psi}{dx}\right|_{w^-} = \left.\frac{d\psi}{dx}\right|_{w^+}$$

where superscript plus and minus signs identify points slightly below and slightly above the indicated value of x, respectively. (While k and κ can be taken to be real for this barrier, the constants A, B, C, D, and F may—and probably will—be complex.) Use MAXIMA's abilities to manipulate expressions to

(a) Obtain the equations determining B, C, D, and F by imposing the stated boundary conditions on these solutions.
(b) Solve those equations for those constants (expressing each as a multiple of the constant A).
(c) Show that the reflection and transmission coefficients R and T defined by $R = |B/A|^2$ and $T = |C/A|^2$ are given by

$$R = \left|\frac{B}{A}\right|^2 = \frac{(\kappa^2 + k^2)^2 \sinh^2 \kappa w}{4\kappa^2 k^2 + (\kappa^2 + k^2)^2 \sinh^2 \kappa w} \quad ; \quad T = \left|\frac{C}{A}\right|^2 = \frac{4k^2\kappa^2}{4\kappa^2 k^2 + (\kappa^2 + k^2)^2 \sinh^2 \kappa w}$$

Here, the vertical bars symbolize the absolute value of the complex number enclosed by them.
(d) Verify that $R + T = 1$.

Solution: (a) In the scattering of a quantum wave from a rectangular barrier, the wave function is defined in three equations. One of these equations describes the wave on the side coming into the barrier, another the wave within the barrier, and the third the wave as it emerges from the barrier. For this problem, we begin by defining the three equations in MAXIMA with the statements

```
(%i1) psi1 : A*exp(%i*k*x) + B*exp(-%i*k*x)$
(%i2) psi2 : D*cosh(kap*x) + F*sinh(kap*x)$
(%i3) psi3 : C*exp(%i*k*x)$
```

Define the three wave functions, suppressing output.

(Recall that, within MAXIMA, the variable `%i` is equivalent to $\sqrt{-1}$.) Now, preparing to impose the boundary conditions, we evaluate the derivatives of these functions with the statements

```
(%i4) diff1 : diff( psi1, x )$
(%i5) diff2 : diff( psi2, x )$
(%i6) diff3 : diff( psi3, x )$
```

Evaluate the derivatives of ψ_1, ψ_2, and ψ_3 with respect to x, again suppressing output.

Physically, the wave function and its first derivative must be continuous at all points. In particular, at the points of discontinuity in the potential, i.e., at $x = 0$ and $x = w$, both the wave functions and their derivatives on either side of the boundary should be equivalent. Thus, we must require that the wave functions and their derivatives satisfy the four equations obtained with the statements

```
(%i7) eq1 :  ev( psi1 = psi2, x=0 );
(%o7)      A + B = D
(%i8) eq2 :  ev( psi2 = psi3, x=w );
(%o8)      D cosh(w κ) + F sinh(w κ) = C e^(i k w)
(%i9) eq3 :  ev( diff1 = diff2, x=0 );
(%o9)      i A k − i B k = F κ
(%i10) eq4 :  ev( diff2 = diff3, x = w);
(%o10)      F κ cosh(w κ) + D κ sinh(w κ) = i C e^(i k w) k
```

Establish the four equations expressing the required boundary conditions.

(b) We can now solve these four equations for the four coefficients B, C, D, and F by executing the statement

```
(%i11) soln :  solve( [eq1,eq2,eq3,eq4], [B,C,D,F] );
```

$$\texttt{(\%o11)} \qquad [[\, B = \text{mess}_B, C = \text{mess}_C, D = \text{mess}_D, F = \text{mess}_F \,]]$$

where the several messes are quite complicated—They are recorded in full detail in the appendix at the end of this solution.—but their details are not here particularly important, suffice it to note that each one has an overall factor of A.

(c) While these solutions have been quickly obtained, they are not the quantities of primary interest. From here, our task is to find the transmission and reflection coefficients and beat them into the form given in the statement of the exercise. Since the reflection and transmission coefficients are determined by B and C, respectively, we shall in what follows ignore D and F. We begin by extracting B and C and evaluating the *amplitudes* for reflection $r = B/A$ and transmission $t = C/A$ with the statements

```
(%i12) r :  ev( B, soln )/A;
```

$$\texttt{(\%o12)} \qquad -\frac{(\kappa^2 + k^2)\,\sinh(\kappa w)}{(\kappa^2 - k^2)\,\sinh(\kappa w) - 2\mathtt{i}k\,\kappa\cosh(\kappa w)}$$

```
(%i13) t :  ev( C, soln )/A;
```

$$\texttt{(\%o13)} \qquad \frac{2\mathtt{i}k\kappa\sinh^2(\kappa w) - 2\mathtt{i}k\kappa\cosh^2(\kappa w)}{(\kappa^2 - k^2)\,e^{\mathtt{i}kw}\,\sinh(\kappa w) - 2\mathtt{i}k\,\kappa\,e^{\mathtt{i}kw}\,\cosh(\kappa w)}$$

We are, of course, not interested in r and t but rather in $R = |r|^2$ and $T = |t|^2$. Since the factor e^{ikw} in both terms in the denominator of t does not affect the absolute value, we simplify t by removing that factor with the statements[6]

```
(%i14) den :  denom( t )$
(%i15) dena :  multthru( exp(-%i*k*w), den )$
(%i16) t :  subst( dena, den, t );
```

$$\texttt{(\%o16)} \qquad \frac{2\mathtt{i}k\kappa\sinh^2(\kappa w) - 2\mathtt{i}k\kappa\cosh^2(\kappa w)}{(\kappa^2 - k^2)\,\sinh(\kappa w) - 2\mathtt{i}k\,\kappa\cosh(\kappa w)}$$

Essentially, we extract the denominator of t, multiply each term in that denominator by e^{-ikw}, and finally substitute that new form for the original denominator in t.

We now must calculate the $|r|^2$ and $|t|^2$. Unfortunately, the function `abs` will achieve the desired objective only when the expressions submitted to `abs` have explicitly numeric values. Instead, we recognize that $|f|^2 = f * f^*$, i.e., f times the complex conjugate of f. We also need to make sure that some multiplications in the result are actually evaluated and that the identity $\cosh^2\theta - \sinh^2\theta = 1$ is used to simplify the expressions. The nested statements

[6] Note that it took quite awhile to find a procedure to to guide MAXIMA in accomplishing this quite simply task.

```
(%i17) R : trigsimp( map( expand, r*conjugate(r) ) );
```

(%o17) $$\frac{(\kappa^4+2k^2\kappa^2+k^4)\sinh^2(\kappa w)}{(\kappa^4+2k^2\kappa^2+k^4)\sinh^2(\kappa w)+4k^2\kappa^2}$$

```
(%i18) T : trigsimp( map( expand, t*conjugate(t) ) );
```

(%o18) $$\frac{4k^2\kappa^2}{(\kappa^4+2k^2\kappa^2+k^4)\sinh^2(\kappa w)+4k^2\kappa^2}$$

achieve all of these tasks efficiently.[7]

Finally, we can achieve a modest further simplification by noticing that the factor $\kappa^4+2k^2\kappa^2+k^4$, which appears in several places, can be factored. The statements

```
(%i19) a : kap^4 + 2*k^2*kap^2 + k^4$
(%i20) b : factor( a );
```

(%o20) $(\kappa^2+k^2)^2$

```
(%i21) R : subst( b, a, R );
```

(%o21) $$\frac{(\kappa^2+k^2)^2\,\sinh^2(\kappa w)}{(k^2+\kappa^2)^2\,\sinh^2(\kappa w)+4k^2\,\kappa^2}$$

```
(%i22) T : subst( b, a, T );
```

(%o22) $$\frac{4k^2\,\kappa^2}{(k^2+\kappa^2)^2\,\sinh^2(\kappa w)+4k^2\,\kappa^2}$$

At last (except for order of terms, about which we can really do nothing), we have the expressions given in the exercise.

(d) Now that we have the simplified equations for R and T we want to confirm that the sum of $R+T$ is indeed 1. The single statement

```
(%i23) combine( R + T );
```

(%o23) 1

confirms that $R+T=1$.

Appendix to Exercise 6.7

We record here the several messes that were represented by abbreviations in part (b) of this exercise:

$$B=-\frac{(\kappa^2+k^2)\sinh(\kappa w)A}{(\kappa^2-k^2)\sinh(\kappa w)-2\mathtt{i}k\kappa\cosh(\kappa w)},$$

$$C=\frac{(2\mathtt{i}k\kappa\sinh^2(\kappa w)-2\mathtt{i}k\kappa\cosh^2(\kappa w))Ae^{-\mathtt{i}kw}}{(\kappa^2-k^2)\sinh(\kappa w)-2\mathtt{i}k\kappa\cosh(\kappa w)}$$

$$D=\frac{(2k^2\sinh(\kappa w)+2\mathtt{i}\kappa\cosh(\kappa w))A}{(\kappa^2-k^2)\sinh(\kappa w)-2\mathtt{i}k\kappa\cosh(\kappa w)},$$

$$F=\frac{(2\mathtt{i}k\kappa\sinh(\kappa w)+2k^2\cosh(\kappa w))A}{(\kappa^2-k^2)\sinh(\kappa w)-2\mathtt{i}k\kappa\cosh(\kappa w)},$$

[7] Probably you should "un-nest" the statements to see the intermediate results, since each intermediate result provides defense for the next step.

6.8 Black Body Spectrum

Exercise: The distribution of wavelengths λ in the blackbody spectrum at (absolute) temperature T is given by

$$u(\lambda, T) = \frac{8\pi ch}{\lambda^5} \frac{1}{e^{ch/(\lambda kT)} - 1}$$

where c is the speed of light, h is Planck's constant, and k is Boltzmann's constant. In terms of the variable $y = ch/\lambda kT$, this function has the alternative expression

$$\frac{(ch)^4\, u(\lambda, T)}{8\pi (kT)^5} = f(y) = \frac{y^5}{e^y - 1}$$

(a) Verify this transformed form and then, using MAXIMA, (b) obtain a graph of $f(y)$ versus y, making sure to extend the graph over an interval that includes its peak and estimate the value of y at which that peak occurs; (c) show that the peak occurs for values of y satisfying

$$(y - 5)\, e^y + 5 = 0$$

(d) obtain a graph of *this* function versus y; (e) make another estimate of the value of y at which the original function has its maximum; and (f) show that the wavelength λ_m at which this maximum occurs satisfies

$$\lambda_m T = 0.28978 \times 10^{-2} \text{m K}$$

Hints: (1) Remember that maxima in a function occur where the derivative of that function with respect to the appropriate variable is zero. (2) Because e^y varies rapidly with y, you may have to play a bit to find a suitable range of values of y over which to plot these graphs.

Solution: (a) To verify the dimensionless transformation, we enter the original equation with the statement[8]

```
(%i1) lam : ⟨ESC⟩l⟨ESC⟩;
(%i2) u : (8*%pi*c*h/lam^5)*1/(exp(c*h/(lam*k*T))-1);
```

(u) $$\frac{8\pi ch}{\lambda^5 (e^{ch/\lambda kT} - 1)}$$

Then, with $y = ch/\lambda kT$, we prepare for substituting into this equation by noting that $\lambda = ch/ykT$. Thus the statements

```
(%i3) f1 : subst(h*c/(y*k*T), lam, u);
```

(f1) $$\frac{8\pi T^5 k^5 y^5}{c^4 h^4 (e^y - 1)}$$

```
(%i3) tmp : (c*h)^3/(8*pi*(k*T)^5)*u;
```

$$tmp := \frac{c^5 h^5}{k^5 T^5 \lambda^5 (e^{ch/\lambda kT} - 1)}$$

and, finally, effect the last transformation with the statement

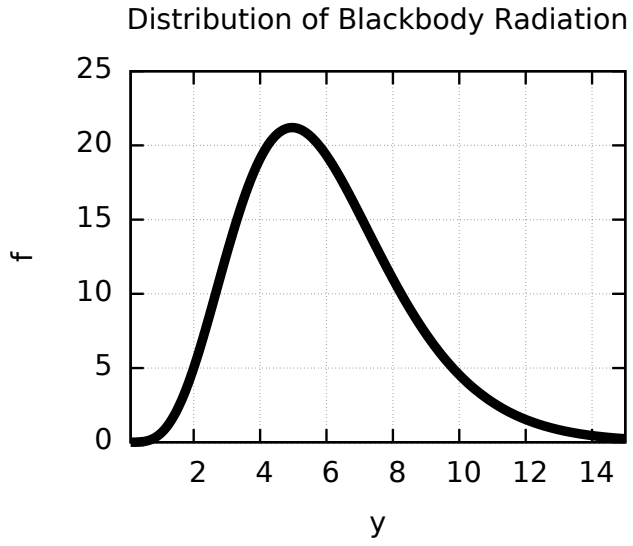
```
(%i4) f : c^4*h^4*f1/(8*%pi*k^5*T^5);
```

(f) $$\frac{y^5}{e^y - 1}$$

(b) We are then asked to graph $f(y)$ versus y. The `MAXIMA` statement

[8] Recall that λ is one of the few Greek letters not available in MAXIMA with the name `lambda`. The first statement here defines a symbol to facilitate entering that Greek character.

Figure E6.9: The distribution of wavelengths, λ, in the blackbody spectrum at an absolute temperature, T



```
(%i5) plot2d( f, [y,0.1,15.0], [color,black],[ylabel,"f"], [style, [lines,4]],
         [title,"Distribution of Blackbody Radiation"], [grid2d,true] );
```

used to create the graph shown in Fig. E6.9 specifies a y-range so that the coordinates of the peak could be estimated with ease. The peak appears to occur at approximately $y = 5$. We will find the exact coordinates of this point later on.

Note that the function involves a division by zero when $y = 0$, so we have to avoid that point in the calculation of values of f.

(c) Next, we are asked to verify that the peak occurs for the value of y satisfying the equation

$$(y - 5)e^y + 5 = 0$$

which we expect to be related to the derivative of our original function. We expect this because the maxima and minima of a function occur where the derivative of that function is equal to zero. The MAXIMA statementd

```
(%i6) dfdy : diff( f, y );
```

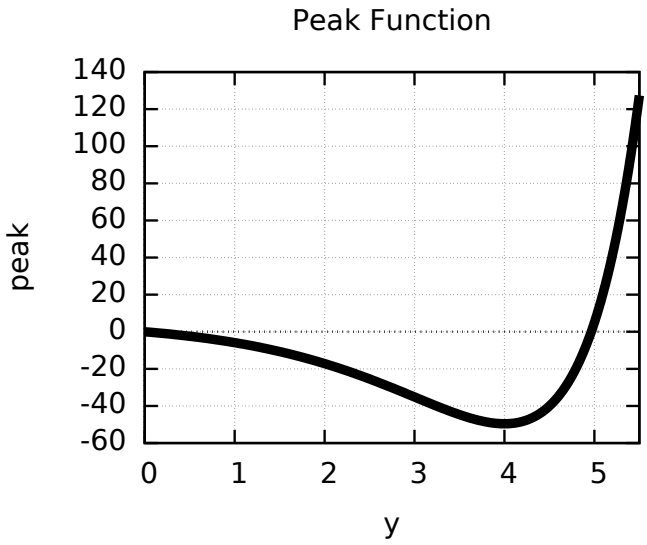
(dfdy) $\dfrac{5y^4}{e^y - 1} - \dfrac{y^5 e^y}{(e^y - 1)^2}$ `(%i7) dfdy : factor( dfdy );`

(dfdy) $-\dfrac{y^4(ye^y - 5e^y + 5)}{(e^y - 1)^2}$

evaluates the derivative and brings it all together over a common denominator. This derivative will be zero if and only if the part of the numerator in parentheses is zero. We extract that numerator with the statement

```
(%i8) peak : num(dfdy)/(-y^4);
```

Figure E6.10: The graph of the function that describes the maxima and minima of the distribution function.



(peak) $ye^y - 5e^y + 5$

```
(%i9) peak : collectterms( peak, exp(y) );
```

(peak) $(y-5)e^y + 5$

Note that we must reject the obvious root $y = 0$ because that would result in an indeterminate value for the radiation intensity itself.

(d) We graph this derivative with the statement

```
(%i10) plot2d( peak, [y,0.0,6.0], [color,black],[ylabel,"peak"], [style, [lines,4]],
         [title,"Peak Function"], [grid2d,true] );
```

This graph is shown in Fig. E6.10. Again it appears as if the one zero of significance in this derivative occurs very close to the value $y = 5.0$.

(e) We make a more refined determination of the value of y at which the original function peaks by solving the function `peak` for the value of y at that function is zero with the statement

```
(%i11) ypeak : find_root( peak, y, 4.5,5.5 );
```

(ypeak) 4.965114231744276

(f) Finally we verify that the maximum wavelength (λ_m) satisfies

$$\lambda_m T = 0.28978 \times 10^{-2}\,\mathrm{mK}$$

at the value of `ypeak` that we just found. Since $y = ch/\lambda kT = ch/xk$ ($x = \lambda T = ch/ky$), we set y equal to the value, $y = 4.965114232$ and introduce both Planck's constant $h = 6.62607015\times10^{-34}$ Js, the speed of light $c = 2.99792458\times10^{8}$ m/s), and Boltzmann's constant ($h = 1.380649\times10^{-23}$ JK^{-1}) with the statements

```
(%i12) x : c*h/(k*y);
```

(x) $\dfrac{ch}{ky}$

```
(%i13) ev( x, y = ypeak, c=2.99792458E8, h=6.62607015E-34, k = 1.380649E-23 );
```

```
(%o13)      0.002897771955185173
```

To work out the units of this result for $\lambda_m T$, we note that[9]

$$x = \lambda_m T = \frac{ch}{ky} \Longrightarrow [x] = \frac{[c][h]}{[k][y]} = \frac{(\text{m/s})(\text{Js})}{(\text{J/K})} = \text{mK}$$

We conclude that $\lambda_m T = 0.2897663979 \times 10^{-2}$ mK, in agreement with the assertion in the statement of the exercise. More specifically, at a temperature $T = 5778$ K (the approximate surface temperature of the sun), the peak of the radiated spectrum occurs at the wavelength

$$\lambda_m = \frac{0.2897771955185173 \times 10^{-2} \text{ mK}}{5778 \text{ K}} = 5.0152 \times 10^{-7} \text{ m} = 501.52 \text{ nm}$$

which is a wavelength in the green region of the visible spectrum. Clearly, the sun is not radiating as a black body.

[9] Square brackets around a symbol means the units of that symbol.

6.10 Laguerre Polynomials

Exercise: The Laguerre polynomials $L_n(x)$ can be defined in many ways. We might, for example, set $L_0(x) = 1$ and then generate each new polynomial in turn by requiring that $L_n(x)$ for $n > 0$ be a polynomial of order n that is orthogonal to all previous polynomials, i.e., that

$$\int_0^\infty e^{-x}\, L_n(x)\, L_m(x)\, dx = 0 \quad , \quad \text{all } m < n$$

and that the arbitrary overall sign remaining be resolved by requiring in addition that $L_n(0) = 1$. Alternatively, they can be determined from the recursion relationship

$$(n+1)L_{n+1}(x) = (2n+1-x)L_n(x) - nL_{n-1}(x)$$

provided we include the first two $L_0(x) = 1$ and $L_1(x) = 1 - x$ to get the recursion started. Yet again, they can be found from the application of multiple differentiation as implied by Rodrigues' formula

$$L_n(x) = \frac{e^x}{n!}\,\frac{d^n}{dx^n}(x^n\, e^{-x})$$

However they are determined, the first half dozen of these polynomials will turn out to be

$$\begin{aligned} L_0(x) &= 1 & \qquad L_3(x) &= 1 - 3x + \tfrac{3}{2}x^2 - \tfrac{1}{6}x^3 \\ L_1(x) &= 1 - x & \qquad L_4(x) &= 1 - 4x + 3x^2 - \tfrac{2}{3}x^3 + \tfrac{1}{24}x^4 \\ L_2(x) &= 1 - 2x + \tfrac{1}{2}x^2 & \qquad L_5(x) &= 1 - 5x + 5x^2 - \tfrac{5}{3}x^3 + \tfrac{5}{24}x^4 - \tfrac{1}{120}x^5 \end{aligned}$$

a. Determine the first half dozen Laguerre polynomials by constructing them one at a time to satisfy the requirements of orthogonality and normalization.

b. Start by binding the value 1 to `L[0]` and the value $1 - x$ to `L[1]`. Then, using the recursion relationship, find the next several Laguerre polynomials. *Hint*: The MAXIMA statements might be

```
(%i1) L[0] : 1;
(%i2) L[1] : 1-x;
```

and then

```
(%i3) L[n] := (2*n-1-x)*L[n-1]/n - (n-1)*L[n-2]/n;
```

(Verify the expression on the right.) With these statements, you have set up $L_0(x)$ and $L_1(x)$ to start the recursion and then you have defined a *function* `L[n]` involving n that can be evaluated at any n. Once L_0 and L_1 have been defined, you can find L_2 and then L_3 and then ... with statements like

```
(%i4) L[2];
(%i5) L[3];
...
```

(Some simplifications may be necessary to cast the results in the standard form of the above table.)

c. Find the first half-dozen Laguerre polynomials by using the command `diff` to evaluate Rodrigues' formula.

d. Be clever and, using either matrices or loops constructed in MAXIMA (see manuals for the structure of a `for` loop), find the values of *all* of the integrals

$$\int_0^\infty e^{-x}\, L_n(x)\, L_m(x)\, dx$$

where n and m take on independently the values 0, 1, 2, 3, 4, 5. (There are 36 integrals to be evaluated. Try to be efficient in your coding.)

e. MAXIMA actually knows quite a bit about many of the important special functions of mathematical physics. In particular, the package `orthopoly`, which can be loaded with the command `load( orthopoly )` and which is described fully in the MAXIMA manuals, provides two different ways to evaluate a very large number of these special functions. Take a look at that documentation and use the function `laguerre` (syntax `laguerre(n, x)`) in that package to determine the first several Laguerre polynomials.

Solution: (a) We will determine the first few Laguerre polynomials by constructing them one at a time using the conditions of orthogonality and normalization. Using the orthogonality integral,

$$\int_0^\infty e^{-x} L_n(x) L_m(x)\, dx$$

we will make sure each Laguerre polynomial is orthogonal to *all* of the polynomials preceding it. We will then use the equation,

$$\int_0^\infty L_n(x)^2\, dx$$

to make sure that each Laguerre polynomial is normalized. The MAXIMA coding used to check these conditions and create the first few Laguerre polynomials with the statements

```
(%i1) L(0) : 1;
```

(%o1) 1

```
(%i2) L[1] : a + b*x;
```

(%o2) $a + bx$

```
(%i3) f : integrate( exp(-x)*L[1]*L[0], x, 0, inf)=0;
```

(f) $b + a = 0$

```
(%i4) f1 : integrate(exp(-x)*(L[1]^2), x, 0 , inf)=1;
```

(f1) $2b^2 + 2ab + a^2 = 1$

```
(%i5) m : solve( [f,f1], [a,b] );
```

(m) $\left[[a = -1, b = 1], [a = 1, b = -1]\right]$

```
(%i6) L[1] : ev( L[1], m[2] );
```

(%o6) $1 - x$

```
(%i7) L[2] : a + b*x + c*x^2;
```

(%o7) $cx^2 + bx + a$

```
(%i8) f2 : integrate(exp(-x)*L[2]*L[1], x, 0, inf)= 0;
```

(f2) $-4c - b = 0$

```
(%i9) f3 : integrate(exp(-x)*L[2]*L[0], x, 0,inf)= 0;
```

(f3) $2c + b + a = 0$

```
(%i10) f4 : integrate(exp(-x)*L[2]^2, x, 0, inf) = 1;
```

(f4) $24c^2 + (12b + 4a)c + 2b^2 + 2ab + a^2 = 1$

```
(%i11) m2 : solve( [f2,f3, f4], [a,b,c] );
```

(m2) $\left[[a = 1, b = -2, c = \frac{1}{2}], [a = -1, b = 2, c = -\frac{1}{2}]\right]$

```
(%i12) L[2] : ev( L[2], m2[1] );
```

(%o12) $\frac{x^2}{2} - 2x + 1$

```
(%i13) L[3] : a + b*x + c*x^2 + d*x^3;
```

(%o13) $dx^3 + cx^2 + bx + a$

```
(%i14) f5 : integrate(exp(-x)*L[3]*L[2], x, 0, inf)= 0;
```

(f5) $18d + 2c = 0$

```
(%i15) f6 : integrate(exp(-x)*L[3]*L[1], x,  0, inf)= 0;
```

(f6) $-18\,d - 4\,c - b = 0$

```
(%i16) f7 : integrate(exp(-x)*L[3]*L[0], x, 0, inf)= 0;
```

(f7) $6d_2c + b + a = 0$

```
(%i17) f8 : integrate(exp(-x)*L[3]^2,x, 0, inf) = 1;
```

(f8) $720d^2 + (240c + 48b + 12a)d + (12b + 4a)c + 2b^2 + 2ab + a^2 = 1$

```
(%i18) m3 : solve( [f5,f6,f7,f8], [a,b,c,d] );
```

(m3) $\left[[a = -1, b = 3, c = -\frac{3}{2}, d = \frac{1}{6}],\ [a = 1, b = -3, c = \frac{3}{2}, d = -\frac{1}{6}]\right]$

```
(%i19) L[3] : ev( L[3], m3[2] );
```

(%o19) $-\frac{1}{6}x^3 + \frac{3}{2}x^2 - 3x + 1$

This coding is a little tedious and complicated. Since we used the variables a, b, etc. more than once, we had to be careful in assigning their values. Since we have squared quantities, there will be an overall sign ambiguity in this problem. When we solved for the various constants, a, b, etc, we were given two sets of solutions. Therefore, we gave these two sets one name and then evaluated the appropriate Laguerre polynomial at one of the elements of the solutions set. For conventions sake, we chose the solution set in which the constant term had the value +1. Although slightly tedious, this method of finding Laguerre polynomials demonstrates clearly what conditions each of them satisfies.

(b) We begin by binding `L[0]` and `L[1]` first, thereby providing the basis for exploiting the recursion relationship, and the MAXIMA coding

```
(%i20) L[0] : 1;
```

(%i20) 1

```
(%i21) L[1] : 1-x;
```

(%o21) $1 - x$

```
(%i22) LL : (2*n-1-x)*L[n-1]/n - (n-1)*L[n-2]/n;
```

(LL) $\frac{L_{n-1}(-x + 2n - 1)}{n} - \frac{L_{n-2}(n - 1)}{n}$

```
(%i23)  L[2] : expand(ev(LL,n=2));
```

(%o23) $$\frac{x^2}{2} - 2x + 1$$

```
(%i24) L[3] : expand(ev(LL,n=3));
```

(%o24) $$-\frac{x^3}{6} + \frac{3x^2}{2} - 3x + 1$$

```
(%i25) L[4] : expand(ev(LL,n=4));
```

(%o25) $$\frac{x^4}{24} - \frac{2c^3}{3} + 3x^2 - 4x + 1$$

```
(%i26) L[5] : expand(ev(LL,n=5));
```

(%o26) $$-\frac{x^5}{120} + \frac{5x^4}{24} - \frac{5x^3}{3} + 5x^2 - 5x + 1$$

is very straightforward.

(c) Find the first half-dozen Laguerre polynomials by using the command `diff` to evaluate Rodrigues' formula

$$L_n(x) = \frac{e^x}{n!}\,\frac{d^n}{dx^n}(x^n e^{-x})$$

This part poses a slight challenge in necessitating the use of `diff` to solve it. If we were to attempt to define a variable `LL` as above, but adapted for the above formula, MAXIMA would throw an error at us, since it has no way to evaluate the derivative without knowing *explicitly* what `n` is. As a circumvention, we can use the inert `'diff` (notice the initial tic) command, which won't evaluate the derivative but instead will simply display it. When we finally get around to evaluating the formula, we can use the `value` function, which returns all `Diff`s as `diff`s. The entire sequence of code

```
(%i27) LL : (exp(x)/n!)*'diff(x^n*exp(-x),x,n);
```

(LL) $$\frac{e^x\left(\dfrac{d^n}{dx^n}(x^n e^{-x})\right)}{n!}$$

```
(%i28) L[0] : 1;
```

(%o28) $$1$$

```
(%i29) L[1] : expand(ev(LL,n=1,diff));
```

(%o29) $$1 - x$$

```
(%i30) L[2] : expand(ev(LL,n=2,diff));
```

(%o30) $$\frac{x^2}{2} - 2x + 1$$

```
(%i31) L[3] : expand(ev(LL,n=3,diff));
```

(%o31) $$-\frac{x^3}{6} + \frac{3x^2}{2} - 3x + 1$$

```
(%i32) L[4] : expand(ev(LL,n=4,diff));
```

(%o32) $$\frac{x^4}{24} - \frac{2x^3}{3} + 3x^2 - 4x + 1$$

```
(%i33) L[5] : expand(ev(LL,n=5,diff));
```

(%o33) $$-\frac{x^5}{120} + \frac{5x^4}{24} - \frac{5x^3}{3} + 5x^2 - 5x + 1$$

could just as easily have been written using the loop

```
(%i34) for i from 0 thru 5 do (L[i] : expand(ev(LL,n=i,diff)), print(L[i]));
```

$$1$$

$$1 - x$$

$$\frac{x^2}{2} - 2x + 1$$

$$-\frac{x^3}{6} + \frac{3x^2}{2} - 3x + 1$$

$$\frac{x^4}{24} - \frac{2x^3}{3} + 3x^2 - 4x + 1$$

$$-\frac{x^5}{120} + \frac{5x^4}{24} - \frac{5x^3}{3} + 5x^2 - 5x + 1$$

```
(%o34)      done
```

The more detailed statements provides a level of clarity for the version involving a loop.

(d) To evaluate the orthogonality integral,

$$\int_0^\infty e^{-x} L_n(x) L_m(x)\, dx$$

where n and m take on independently the values 0, 1, 2, 3, 4, 5, we use the results for the polynomials already generated in a nested loop. The statements

```
(%i35) orth : zeromatrix(6,6)$
(%i36) for m from 0 thru 5 do for n from 0 thru 5 do
            orth[m+1,n+1] : integrate( exp(-x)*L[n]*L[m],x,0,inf);

(%o36)      done
(%i37) orth;
```

$$\text{(\%o37)} \qquad \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

that will give us our array. Notice that only the diagonals have values, a property of the orthogonality of these polynomials. That all diagonal values have the value 1 also means that the polynomials are normalized.

(e) To use the built-in function for the Laguerre polynomials, we invoke the statements

```
(%i38) load(orthopoly)$
(%i39) for n from 0 thru 5 do print( laguerre(n,x) );
```

The output from `(%i39)` is suppressed because it is identical to that produced at `(%i34)`.

6.12 Finding Eigenvectors/Values

Exercise:

(a) Find the eigenvalues and eigenvectors of the matrix

$$\begin{pmatrix} 2 & -1 & 0 & 0 \\ -1 & 2 & -1 & 0 \\ 0 & -1 & 2 & -1 \\ 0 & 0 & -1 & 2 \end{pmatrix}$$

(b) Use the function `part` to explore the structure of the entity returned by the function `eigenvectors` used in part (a).

Solution: (a) We begin this exercise by creating a matrix so that the `eigenvectors` function will be able to work on it later. We do this explicitly with the statement

```
(%i1) themat : matrix( [2, -1, 0 , 0 ], [-1, 2, -1, 0], [0, -1, 2, -1],
                [0, 0, -1, 2] );
```

(%o1)
$$\begin{pmatrix} 2 & -1 & 0 & 0 \\ -1 & 2 & -1 & 0 \\ 0 & -1 & 2 & -1 \\ 0 & 0 & -1 & 2 \end{pmatrix}$$

Now, we use the function `eigenvectors` to find the eigenvalues and eigenvectors of the matrix with the statements

```
(%i2) eig :  eigenvectors( themat );
```

(%o2)
$$\left[\left[\left[-\frac{\sqrt{5}-3}{2}, \frac{\sqrt{5}+3}{2}, -\frac{\sqrt{5}-5}{2}, \frac{\sqrt{5}+5}{2}\right], [1,1,1,1,]\right],\right.$$
$$\left[\left[\left[1, \frac{\sqrt{5}+1}{2}, \frac{\sqrt{5}+1}{2}, 1\right]\right], \left[\left[1, -\frac{\sqrt{5}-1}{2}, -\frac{\sqrt{5}-1}{2}, 1\right]\right],\right.$$
$$\left.\left.\left[\left[1, \frac{\sqrt{5}-1}{2}. -\frac{\sqrt{5}-1}{2}, -1\right]\right], \left[\left[1, -\frac{\sqrt{5}+1}{2}, \frac{\sqrt{5}+1}{2}, -1\right]\right]\right]\right]$$

The output here consists of two main lists, each enclosed in a pair of left-right, middle-sized brackets,[10] specifically,

```
(%i3) eig[1];
```

(%o3)
$$\left[\left[-\frac{\sqrt{5}-3}{2}, \frac{\sqrt{5}+3}{2}, -\frac{\sqrt{5}-5}{2}, \frac{\sqrt{5}+5}{2}\right], [1,1,1,1,]\right]$$

```
(%i4) eig[2];
```

(%o4)
$$\left[\left[\left[1, \frac{\sqrt{5}+1}{2}, \frac{\sqrt{5}+1}{2}, 1\right]\right], \left[\left[1, -\frac{\sqrt{5}-1}{2}, -\frac{\sqrt{5}-1}{2}, 1\right]\right],\right.$$
$$\left.\left[\left[1, \frac{\sqrt{5}-1}{2}. -\frac{\sqrt{5}-1}{2}, -1\right]\right], \left[\left[1, -\frac{\sqrt{5}+1}{2}, \frac{\sqrt{5}+1}{2}, -1\right]\right]\right]$$

[10] Those sizes are not used in the on-screen display; we have imposed them to clarify the structure of this expression.

The first of these top-level components, `eig[1]`, itself consists of two sublists, the first of which provides the eigenvalues and the second of which gives the degeneracy of each eigenvalue. Here, there are four distinct (non-degenerate) eigenvalues. The second of these top-level components `eig[2]` consists of several—here four—sublists, one for each *distinct* eigenvalue. These sublists provide the (unnormalized) eigenvectors. In the present case, there are four distinct eigenvalues and one eigenvector for each eigenvalue. The apparently redundant square brackets around each eigenvector would not be redundant had, say, one of the eigenvalues been, say, doubly degenerate, in which case the outer brackets for that eigenvector would have enclosed *two* lists, each providing one of the eigenvectors, i.e., would have had the structure

[[first eigenvector], [second eigenvector]]

Eigenvalues, degeneracies, and eigenvectors are given in the same order in their respective lists.

(b) Let us now use some of MAXIMA's commands, in particular `part`, to explore the structure of `eig` more explicitly, noting first that `part(eig, 1)` and `part(eig,2)` return the same expressions as `eig[1]` and `eig[2]`. To extract the first eigenvalue, its degeneracy, and the associated eigenvector(s), we would execute the statement

```
(%i5) [part(eig,1,1,1), part( eig,1,2,1), part(eig,2,1,1)];
```

(%o5) $$\left[-\frac{\sqrt{5}-3}{2}, 1, [1, \frac{\sqrt{5}+1}{2}, \frac{\sqrt{5}+1}{2}, 1]\right]$$

from which we learn that the first eigenvalue is $-(\sqrt{5}-3)/2$, the degeneracy of that eigenvalue is 1, and the corresponding eigenvector is $[1, (\sqrt{5}+1)/2, (\sqrt{5}+1)/2, 1]$.

Other eigenvalues, degeneracies, and eigenvectors can be extracted from the output of `eigenvectors` with similar statements.

Suppose we wanted to normalize the first eigenvector. We might, for example, extract the first eigenvector as a list, evaluate the square root of the dot product of that vector with its transpose to find magnitude of the vector, and then divide the vector by its magnitude with the statements

```
(%i6) eig1 :  part( eig, 2, 1, 1 )$
(%i7) mag :  sqrt( expand( eig1.transpose(eig1) ) );
```

(%o7) $$\sqrt{\sqrt{5}+5}$$

```
(%i8) normeig1 :  eig1/mag;
```

(%o8) $$\left[\frac{1}{\sqrt{5}+5}, \frac{\sqrt{5}+1}{\sqrt{5}+5}, \frac{\sqrt{5}+1}{\sqrt{5}+5}, \frac{1}{\sqrt{5}+5}\right]$$

We might also wish to verify the orthogonality of eigenvectors belonging to different eigenvalues. We might use `eig1` already extracted and execute the statements

```
(%i9) eig2 :  part( eig, 2, 2, 1 )$
(%i10) expand( eig1.transpose(eig2) );
(%o10) 0
```

to demonstrate that `eig1` and `eig2` are indeed orthogonal.

6.13 Stark Effect for $n = 2$ and $n = 3$

Exercise: When a (weak) constant external electric field of magnitude F—We reserve E for energy in this exercise.—is imposed on a hydrogen atom, the energy of the states with principal quantum number n shift from the energy given by the Bohr model by amounts determined by the eigenvalues of the matrix whose elements are $\langle nlm|eF\hat{z}|nl'm'\rangle$, where l, m, l', and m' range over all possible values of those quantum numbers allowed by the particular value of n. If the states by which the rows and columns are labeled are ordered $|2,0,0\rangle$, $|2,1,-1\rangle$, $|2,1,0\rangle$, and $|2,1,1\rangle$, then the matrix for the state $n = 2$ is

$$3ea_0F \begin{pmatrix} 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

where e is the magnitude of the charge on the electron and a_0 is the Bohr radius. Similarly, if the states by which the rows and columns are labeled are ordered $|3,2,2\rangle$, $|3,1,1\rangle$, $|3,2,1\rangle$, $|3,0,0\rangle$, $|3,1,0\rangle$, $|3,2,0\rangle$, $|3,1,-1\rangle$, $|3,2,-1\rangle$, and $|3,2,-2\rangle$, then the matrix for the state $n = 3$ is

$$3ea_0F \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -9/2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -9/2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -3\sqrt{6} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -3\sqrt{6} & 0 & -9/\sqrt{3} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -9/\sqrt{3} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -9/2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -9/2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

Find the eigenvalues and eigenvectors of these matrices. The eigenvalues give the energy shifts for the Stark effect for $n = 2$ and $n = 3$ and the eigenvectors give the linear combinations of the base states (i.e., the states in the absence of the external field) out of which the states in the presence of the field emerge as the field is turned on. *Hint*: Note that the command `zeromatrix` may be useful in speeding the creation of these matrices, many of whose elements *are* zero and the command `radcan` may be useful in simplifying results.

Solution: To find the eigenvalues and eigenvectors of the matrix for the $n = 2$ state, we must first construct the matrix with the statements

```
(%i1) mat :  zeromatrix(4,4)$
(%i2) mat[1, 3] :  -1$
(%i3) mat[3, 1] :  -1$
(%i4) mat;
```

Create a 4×4 matrix of zeros.
Set element in row 1, column 3.
Set element in row 3, column 1.
Show the matrix.

(%o4)
$$\begin{pmatrix} 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

Once our matrix has been created, we find the eigenvalues, degeneracies, and (unnormalized) eigenvectors using the MAXIMA function `eigenvectors` with the statements

```
(%i5) eig :  eigenvectors( mat );
```

(%o5)
$$\left[\left[[-1,1,0],[1,1,2] \right], \left[[\,[1,0,1,0]\,],[\,[1,0,-1,0]\,],[\,[0,1,0,0],[0,0,0,1]\,] \right] \right]$$

In the output on the screen, all brackets are the same size. We have here used different sizes in an effort to clarify the structure of that output. To clarify even further, we extract the pieces

```
(%i5) vals :  eig[1][1];                          Extract eigenvalues
(%o5)     [−1, 1, 0]
(%i6) deg :  eig[1][2];                           Extract degeneracies
(%o6)     [1, 1, 2]
(%i7) vecm1 :  eig[2][1][1];                      Extract vector for eigenvalue −1
(%o7)     [1, 0, 1, 0]
(%i8) vecp1 :  eig[2][2][1];                      Extract vector for eigenvalue +1
(%o8)     [1, 0, −1, 0]
(%i9) vec0a :  eig[2][3][1];                      Extract two vectors for eigenvalue 0
(%o9)     [0, 1, 0, 0]
(%i10) vec0b :  eig[2][3][2];
(%o10)     [ 0, 0, 0, 1 ]
```

The eigenvalues (-1, 1, and 0—which is doubly degenerate) represent the energy shifts for the Stark effect for the $n = 2$ state, and the eigenvectors give the linear combinations of the base states out of which the states in the presence of the electric field emerge as the field is turned on. Notice that the first and second eigenvectors are not normalized. The alternative MAXIMA command `uniteigenvectors` (or `ueivects` for short) outputs *normalized* eigenvectors, specifically,

```
(%i11) eign :  uniteigenvectors( mat );
```

$$\text{(\%o11)} \quad \left[\left[[-1,1,0], [1,1,2] \right], \left[\left[\left[\frac{1}{\sqrt{2}}, 0, \frac{1}{\sqrt{2}}, 0 \right] \right], \left[\left[\frac{1}{\sqrt{2}}, 0, -\frac{1}{\sqrt{2}}, 0 \right] \right], \left[[0,1,0,0], [0,0,0,1] \right] \right] \right]$$

The eigenvalues and degeneracies are, of course, the same as those first obtained; the eigenvectors, however, have been normalized.

We also know that the eigenvectors belonging to *different* eigenvalues should be orthogonal. To confirm that property, we begin by executing the statement

```
(%i12) matvecs :  matrix( eign[2][1][1], eign[2][2][1], eign[2][3][1],
               eign[2][3][2] );
```

$$\text{(\%o12)} \quad \begin{bmatrix} 1/\sqrt{2} & 0 & 1/\sqrt{2} & 0 \\ 1/\sqrt{2} & 0 & -1/\sqrt{2} & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

to create a matrix whose rows are the eigenvectors we have found. Then, we simply (matrix) multiply this matrix into its transpose,

```
(%i13) matvecs.transpose( matvecs );
```

to find the diagonal unit matrix

$$\text{(\%o13)} \quad \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

which demonstrates that each eigenvector is normalized (ones along the diagonal) and the each eigenvector is orthogonal to the other three eigenvectors. Indeed, the two eigenvectors that belong to the degenerate eigenvalue happen also to be orthogonal to each other, a property not necessarily guaranteed by the algorithms finding the eigenvectors.

For the $n = 3$ state, the method is identical but requires a bit more computational work. We start by creating our matrix with the statements

```
(%i14) mat2 :  zeromatrix(9,9)$                    Create a 9 × 9 matrix of zeros.
(%i15) mat2[2, 3] :  -9/2$                         Set element in row 2, column 3.
(%i16) mat2[4, 5] :  -3*sqrt(6)$                   Set element in row 4, column 5.
(%i15) mat2[5, 6] :  -9/sqrt(3)$                   Set element in row 5, column 6.
(%i16) mat2[7, 8] :  -9/2$                         Set element in row 7, column 8.
(%i17) mat2 :  mat2 + transpose( mat2 );           Construct and display result
```

$$\text{(\%o17)} \quad \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -9/2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -9/2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -3\sqrt{6} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -3\sqrt{6} & 0 & -3^{3/2} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -3^{3/2} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -9/2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -9/2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

Rather than entering all eight non-zero values explicitly, we have here recognized that the matrix of interest is symmetric and created it by first entering the four elements above the main diagonal and then adding that matrix to its transpose.

Now we find the eigenvalues and (normalized) eigenvectors as before, invoking again the command `uniteigenvectors`, specifically

```
(%i18) eig3 :  uniteigenvectors( mat2 );
```

$$\text{(\%o18)} \quad \left[\left[[-\frac{9}{2}, \frac{9}{2}, -9, 9, 0], [2, 2, 1, 1, 3] \right], \right.$$

$$\left[[\, [0, \frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}}, 0, 0, 0, 0, 0, 0], \ [0, 0, 0, 0, 0, 0, \frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}}, 0] \,], \right.$$

$$[\, [0, \frac{1}{\sqrt{2}}, -\frac{1}{\sqrt{2}}, 0, 0, 0, 0, 0, 0], [0, 0, 0, 0, 0, 0, \frac{1}{\sqrt{2}}, -\frac{1}{\sqrt{2}}, 0] \,],$$

$$[\, [0, 0, 0, \frac{1}{\sqrt{3}}, \frac{\sqrt{6}}{2\sqrt{3}}, \frac{1}{\sqrt{2}\sqrt{3}}, 0, 0, 0] \,],$$

$$[\, [0, 0, 0, \frac{1}{\sqrt{3}}, -\frac{\sqrt{6}}{2\sqrt{3}}, \frac{1}{\sqrt{2}\sqrt{3}}, 0, 0, 0] \,],$$

$$\left. \left. [\, [1, 0, 0, 0, 0, 0, 0, 0, 0], [0, 0, 0, \frac{1}{\sqrt{3}}, 0, -\frac{\sqrt{2}}{\sqrt{3}}, 0, 0, 0], [0, 0, 0, 0, 0, 0, 0, 0, 1] \,] \right] \right]$$

The eigenvalues are $9/2$ and $-9/2$ (both doubly degenerate), 9 and -9 (both non-degenerate), and 0 (triply degenerate). These eigenvalues and eigenvectors represent the energy shift and the linear combinations of the base states for the $n = 3$ state. Note that the groupings of the nine states in the eigenvectors are not as thoroughly jumbled as one might fear. The states for eigenvalue $\pm 9/2$ mix only the states $|3, 1, 1\rangle$ and $|3, 2, 1\rangle$ and, separately, the states $|3, 1, -1\rangle$ and $|3, 2, -1\rangle$; the states for

eigenvalues ± 9 mix only the states $|3,0,0\rangle$, $|3,1,0\rangle$, and $|3,2,0\rangle$; and two of the states for eigenvalue zero ($|3,2,2\rangle$ and $|3,2,-2\rangle$) are unchanged by the perturbation while the third is a mix of the states $|3,0,0\rangle$ and $|3,2,0\rangle$.

We can confirm the orthonormality of the nine eigenvectors for $n = 3$ in the same way we invoked for $n = 2$. The statement

```
%i19 matvecs : matrix( eig3[2][1][1], eig3[2][1][2], eig3[2][2][1],
        eig3[2][2][2], eig3[2][3][1], eig3[2][4][1],
        eig3[2][5][1], eig3[2][5][2], eig3[2][5][3] )$
```

creates a 9×9 matrix whose rows are the nine eigenvectors, and the statement. Then, we simply (matrix) multiply this matrix into its transpose,

```
(%i20) matvecs.transpose( matvecs );
```

to find the diagonal unit matrix

(%o13)
$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

again demonstrating that the eigenvectors here found are mutually orthonormal.

6.15 Lagrangian Approach to Equations of Motion

Exercise: As shown in Fig. E6.11, an object of mass m is connected to the center of each of the four sides of a square of sides 2ℓ with a spring of constant k and moves on a horizontal frictionless surface in the plane defined by the square. Take the equilibrium position of the object at the center of the square to be the origin of an xy coordinate system, and let the springs have unstretched length ℓ_0. For this situation, the kinetic and potential energies KE and PE of the object are given by

$$KE = \frac{1}{2}m\dot{x}^2 + \frac{1}{2}m\dot{y}^2 \quad ; \quad PE = \frac{1}{2}k\Big[\sqrt{(\ell - x)^2 + y^2} - \ell_0\Big] + \ldots$$

where PE will have three additional terms, one for each of the remaining springs. (The term shown applies to the spring connected to the right side of the square.) By definition, the Lagrangian function for this problem is given by $L = KE - PE$ and the equations of motion can be extracted from the Lagrangian by evaluating the expression

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{q}}\right) - \frac{\partial L}{\partial q} = 0$$

where q stands first for x and then for y. Write out the full potential energy and then use MAXIMA to deduce the equations of motion $m\ddot{x} = \ldots$ and $m\ddot{y} = \ldots$ for this system. The results are quite complicated. To simplify the problem, find in detail the equations only for the cases (a) $\ell_0 = 0$, i.e., the springs have unstretched length of 0—admittedly unrealistic springs, and (b) x and y remain small compared to ℓ throughout the motion, i.e., $x/\ell \ll 1$ and $y/\ell \ll 1$ for all time. *Note*: Statements like `diff( L, diff(x,t) )` are perfectly acceptable to MAXIMA, provided the dependence of `x` on `t` is indicated, either explicitly (`x(t)`) or through a prior execution of the statement `depends( x, t )`.

Solution: To derive the equations of motion for an object of mass m attached to a system of four identical springs and moving in a horizontal plane, we represent the system in terms of its energies and use the Lagrangian approach. The Lagrangian L is defined as the kinetic energy minus the potential energy. For this system the kinetic energy is easy to write down; it has the expression

$$KE = \frac{1}{2}m\dot{x}^2 + \frac{1}{2}m\dot{y}^2$$

The potential energy, however, is more difficult because it is determined by the extension (actual length minus unstretched length) of the springs when the object is at a general position (x, y). For example, if the unstretched length of each spring is ℓ_0 and the sides of the box are all 2ℓ, then the

Figure E6.11: Figure for Exercise 6.15.

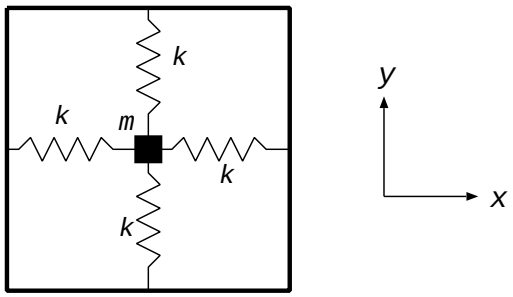
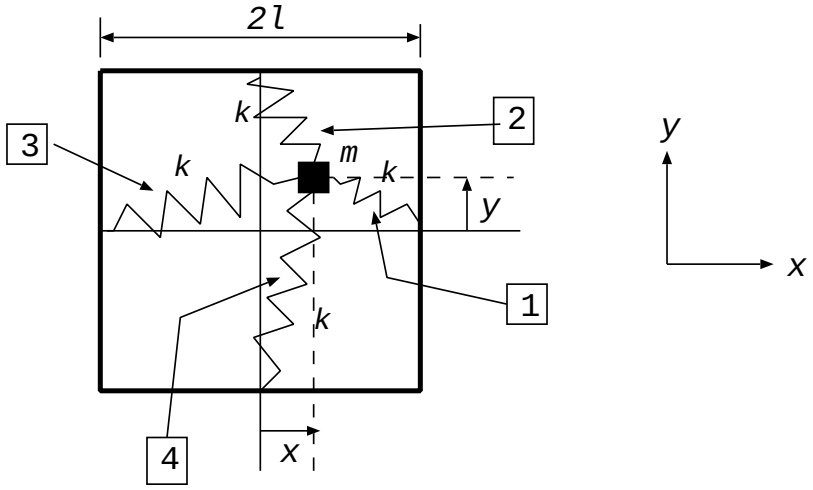


Figure E6.12: Object of mass m displaced from its equilibrium position at the origin to the point (x, y).



---

potential energy stored in the spring numbered $\boxed{1}$ is

$$PE_1 = \frac{1}{2}k\left[\sqrt{(\ell - x)^2 + y^2} - \ell_0\right]^2$$

Similarly, the potential energy stored in the remaining springs is given by

$$\begin{aligned} PE_2 &= \frac{1}{2}k\left[\sqrt{x^2 + (\ell - y)^2} - \ell_0\right]^2 \\ PE_3 &= \frac{1}{2}k\left[\sqrt{(\ell + x)^2 + y^2} - \ell_0\right]^2 \\ PE_4 &= \frac{1}{2}k\left[\sqrt{x^2 + (\ell + y)^2} - \ell_0\right]^2 \end{aligned}$$

With these expressions, the Lagrangian then is

$$L = KE - PE_1 - PE_2 - PE_3 - PE_4$$

and the equations of motion for x and y can be extracted using the relations

$$\begin{aligned} \frac{d}{dt}\left(\frac{\partial L}{\partial \dot{x}}\right) - \frac{\partial L}{\partial x} &= 0 \\ \frac{d}{dt}\left(\frac{\partial L}{\partial \dot{y}}\right) - \frac{\partial L}{\partial y} &= 0 \end{aligned}$$

Note here that the equations of motion for both x and y will be (essentially) identical, since rotating our coordinate system 90 degrees will not change how the problem looks. So, once the equation of motion for x is derived, the equation of motion for y can be obtained simply by replacing x with y.

Our first step, then, is to construct the Lagrangian, which we do in several steps with the statements[11]

[11] We suppress the output from the first five lines because it is already displayed earlier in this solution.

```
(%i1) KE : m* xdot^2/2 + m*ydot^2/2$                    Set kinetic energy.
(%i2) PE1 : k*(sqrt((l - x)^2 + y^2) - l0)^2/2$         Set potential energy in spring 1.
(%i3) PE2 : k*(sqrt((l - y)^2 + x^2) - l0)^2/2$         Set potential energy in spring 2.
(%i4) PE3 : k*(sqrt((l + x)^2 + y^2) - l0)^2/2$         Set potential energy in spring 3.
(%i5) PE4 : k*(sqrt((l + y)^2 + x^2) - l0)^2/2$         Set potential energy in spring 4.
(%i6) L : KE - PE1 - PE2 - PE3 - PE4;                   Evaluate Lagrangian.
```

$$\text{(\%o6)}\quad \frac{m\dot{y}^2}{2}-\frac{k\left(\sqrt{(y+l)^2+x^2}-l_0\right)^2}{2}-\frac{k\left(\sqrt{y^2+(x+l)^2}-l_0\right)^2}{2}-\frac{k\left(\sqrt{y^2+(l-x)^2}-l_0\right)^2}{2}-\frac{k\left(\sqrt{(l-y)^2+x^2}-l_0\right)^2}{2}+\frac{m\dot{x}^2}{2}$$

Following the suggestion in the statement of the exercise, we then find the equations of motion, starting with the statements

```
(%i7) dLdxdot : diff(L, xdot);
```

$$\text{(\%o7)}\quad m\dot{x}$$

```
(%i8)} dLdx : diff(L, x);
```

$$\text{(\%o8)}\quad -\frac{kx\left(\sqrt{(y+l)^2+x^2}-l_0\right)}{\sqrt{(y+l)^2+x^2}}-\frac{k\,(x+l)\left(\sqrt{y^2+(x+l)^2}-l_0\right)}{\sqrt{y^2+(x+l)^2}}+\frac{k\,(l-x)\left(\sqrt{y^2+(l-x)^2}-l_0\right)}{\sqrt{y^2+(l-x)^2}}-\frac{kx\left(\sqrt{(l-y)^2+x^2}-l_0\right)}{\sqrt{(l-y)^2+x^2}}$$

```
(%i9) dLdydot : diff(L, ydot);
```

$$\text{(\%o9)}\quad m\dot{y}$$

```
(%i10) dLdy : diff(L, y);
```

$$\text{(\%o10)}\quad -\frac{k\,(y+l)\left(\sqrt{(y+l)^2+x^2}-l_0\right)}{\sqrt{(y+l)^2+x^2}}-\frac{ky\left(\sqrt{y^2+(x+l)^2}-l_0\right)}{\sqrt{y^2+(x+l)^2}}+\frac{ky\left(\sqrt{y^2+(l-x)^2}-l_0\right)}{\sqrt{y^2+(l-x)^2}}-\frac{k\left(\sqrt{(l-y)^2+x^2}-l_0\right)(l-y)}{\sqrt{(l-y)^2+x^2}}$$

These results are far too complicated to support a simple treatment from here on. To illustrate the strategy that we would adopt if we *really* wanted to know the general equations of motion, we focus instead on two special cases.

Case 1: Suppose that the natural (unstretched) length of the springs is in fact 0, i.e., suppose that $\ell_0 = 0$. Then, somewhat surprisingly, the Lagrangian and its derivatives reduce to

```
(%i11) L1 : expand( ev(L, l0=0) );
```

$$\text{(\%o11)}\quad \frac{m\dot{y}^2}{2}-2ky^2+\frac{m\dot{x}^2}{2}-2kx^2-2kl^2$$

```
(%i12) dL1dxdot : diff(L1, xdot);
```

$$\text{(\%o12)}\quad m\dot{x}$$

```
(%i13) dL1dx : diff(L1, x);
```

(%o13) $-4kx$

```
(%i14) dL1dydot : diff(L1, ydot);
```

(%o14) $m\dot{y}$

```
(%i15) dL1dy : diff(L1, y);
```

(%o15) $-4ky$

That the complicated expressions should simplify so dramatically in this admittedly idealized situation is a bit of a surprise.

These equations are, of course, not yet the equations of motion. We need to bring the *real* independent variable t into greater prominence so we can take the next step. To that end, we make the replacements

```
(%i16) dL1dxdot : ev( dL1dxdot, x = x(t), xdot = diff(x(t), t) );
```

(%o16) $m\dfrac{dx(t)}{dt}$

```
(%i17) dL1dx : ev( dL1dx, x = x(t), xdot = diff(x(t), t) );
```

(%o17) $-4kx(t)$

```
(%i18) dL1dydot : ev( dL1dydot, y = y(t), ydot = diff(y(t), t) );
```

(%o18) $m\dfrac{dy(t)}{dt}$

```
(%i19) dL1dy : ev( dL1dy, y = y(t), ydot = diff(y(t), t) );
```

(%o19) $-4ky(t)$

Finally, we find the equations of motion themselves with the statements

```
(%i20) eq1 : diff(dL1dxdot, t) - dL1dx = 0;
```

(%o20) $m\dfrac{d^2x(t)}{dt^2} + 4kx(t) = 0$

```
(%i21) eq2 : diff(dL1dydot, t) - dL1dy = 0;
```

(%o20) $m\dfrac{d^2y(t)}{dt^2} + 4ky(t) = 0$

These equations are uncoupled and—though the exercise didn't ask for solutions—are readily solved and simplified with the statements. We begin by stipulating general initial conditions with the statements

```
(%i22) atvalue( x(t), t=0, x0 )$
(%i23) atvalue( diff(x(t),t), t=0, xdot0 )$
(%i24) atvalue( y(t), t=0, y0 )$
(%i25) atvalue( diff(y(t),t), t=0, ydot0 )$
```

Then we request the solution with the statement

```
(%i22) soln : desolve( [eq1, eq2], [x(t), y(t)] )$
```

Is km positive, negative, or zero? `p;`

Because the immediate solution is extensive, we suppress its immediate display. Looking at it on the screen, however, we realize that (1) replacing $\sqrt{km}$ with $m\omega$ (i.e., $\sqrt{k/m}$ with ω)[12] and then (2) expanding the solutions to cancel and m in the denominators with the same factor in the numerators with the statement

```
(%i23) solna : ev( soln, sqrt(k*m) = m*\omega  )$
(%i24) solnb :  expand( solna );
```

(%o24) $$\left[x(t) = \frac{\dot{x}_0 \sin(2\omega t)}{2\omega} + x_0 \cos(2\omega t), y(t) = \frac{\dot{y}_0 \sin(2\omega t)}{2\omega} + y_0 \cos(2\omega t)\right]$$

will simplify the expression. We conclude that, when the springs are identical *and have zero unstretched length*, the motion of the object consists of SHM independently in both the x and y directions and the frequency of both components of the motion is $2\omega = 2\sqrt{k/m}$.

Case 2: Suppose that the amplitude of the motion in each coordinate direction is small compared to ℓ so that we can linearize the equations of motion once we have them. That is, suppose we can ignore terms in the equations of motion that involve x^2, y^2, xy, and higher powers of the dependent variables. Equivalently, since the equations of motion are determined by differentiating the Lagrangian with respect to x and y, we can find the equations of motion in this limiting case not by finding the general equations of motion and then linearizing them but more easily by expanding the Lagrangian to include terms that are quadratic in x and y and then deducing the equations of motion. Indeed, for this case, the kinetic energy KE is already quadratic in the significant variables, so we need do nothing to simplify it. We determine the potential energy by itself and then simplify it with the statements

```
(%i25) PE : PE1 + PE2 + PE3 + PE4$
```
Find potential energy.

```
(%i26) PEa : taylor( PE, [x, y], 0, 3 )$
```
Evaluate successive Taylor expansions to include terms through x^2 and y^2.

```
(%i27) PEb : expand( collectterms(PEa, x, y) );
```
Simplify the result.

(%o27) $$-\frac{kl_0y^2}{l} + 2ky^2 - \frac{kl_0x^2}{l} + 2kx^2 + 2kl_0{}^2 - 4kll_0 + 2kl^2$$

In this special case, the Lagrangian and its derivatives reduce to

```
(%i28) L2 : KE - PEb;
```

(%o28) $$\frac{m\dot{y}^2}{2} + \frac{kl_0y^2}{l} - 2ky^2 + \frac{m\dot{x}^2}{2} + \frac{kl_0x^2}{l} - 2kx^2 - 2kl_0^2 + 4kll_0 - 2kl^2$$

```
(%i29) dL2dxdot : diff(L2, xdot);
```

(%o29) $m\dot{x}$

```
(%i30) dL2dx : diff(L2, x);
```

(%o30) $$\frac{2kl_0x}{l} - 4kx$$

```
(%i31) dL2dydot : diff(L2, ydot);
```

(%o31) $m\dot{y}$

```
(%i32) dL2dy : diff(L2, y);
```

(%o32) $$\frac{2kl_0y}{l} - 4ky$$

[12]The first form of the replacement is necessary because, in the structure of the result, $\sqrt{km}$ appears as an entity but $\sqrt{k/m}$ does not.

As in the first special case, these equations are also not yet the equations of motion. We need to bring the *real* independent variable t into greater prominence so we can take the next step. To that end, we make the replacements

```
(%i33) dL2dxdot : ev( dL2dxdot, x = x(t), xdot = diff(x(t), t) );
```

(%o33) $$m\frac{dx(t)}{dt}$$

```
(%i34) dL2dx : ev( dL2dx, x = x(t), xdot = diff(x(t), t) );
```

(%o34) $$\frac{2kl_0x(t)}{l} - 4kx(t)$$

```
(%i35) dL2dydot : ev( dL2dydot, y = y(t), ydot = diff(y(t), t) );
```

(%o35) $$m\frac{dy(t)}{dt}$$

```
(%i36) dL2dy : ev( dL2dy, y = y(t), ydot = diff(y(t), t) );
```

(%o36) $$\frac{2kl_0y(t)}{l} - 4ky(t)$$

Finally, we find the equations of motion themselves with the statements

```
(%i37) eq1 : diff(dL2dxdot, t) - dL2dx = 0;
```

(%o37) $$m\frac{d^2x(t)}{dt^2} - \frac{2kl_0x(t)}{l} + 4kx(t) = 0$$

```
(%i38) eq2 :  diff(dL2dydot, t) - dL2dy = 0;
```

(%o38) $$m\frac{d^2y(t)}{dt^2} - \frac{2kl_0y(t)}{l} + 4ky(t) = 0$$

These equations are still uncoupled and can be readily solved just as we did for the case $l_0 = 0$. We need not, however, find that solution explicitly to conclude, to no particular surprise, that the motion in each coordinate direction is simple harmonic and that the frequency of the motion is the same in both directions—a frequency given by

$$\Omega^2 = \frac{4k - 2kl_0/l}{m} \Longrightarrow \Omega = \sqrt{2\frac{k}{m}\left(2 - \frac{\ell_0}{\ell}\right)} = \omega\sqrt{2\left(2 - \frac{\ell_0}{\ell}\right)} = 2\omega\sqrt{1 - \frac{\ell_0}{2\ell}}$$

where, again, $\omega = \sqrt{k/m}$. In particular, this expression reduces to 2ω when $\ell_0 = 0$, a result consistent with that deduced in the previous example. More generally, the frequency of (small-amplitude) oscillations depends on the relationship between the unstretched length of the spring ℓ_0 and the side of the box 2ℓ. If, for example, $\ell_0 = \ell$, i.e., the spring is neither stretched nor compressed when the object is in its equilibrium position, then the frequency reduces to $\Omega = \sqrt{2}\,\omega$, a result consistent with the conclusion that, in those circumstances, only the two springs parallel to one side affect the motion in a direction parallel to that side.

6.17 Elliptic Integrals

Exercise: The complete elliptic integrals of the first and second kinds are given by

$$K(k) = \int_0^{\pi/2} \frac{d\phi}{(1 - k^2 \sin^2 \phi)^{1/2}} \quad ; \quad E(k) = \int_0^{\pi/2} (1 - k^2 \sin^2 \phi)^{1/2} \, d\phi$$

Use MAXIMA to evaluate these integrals to $O(k^6)$ by expanding the integrands in Taylor series *before* evaluating the integrals.

Solution: In this exercise, we will look at both the MAXIMA solutions to the elliptic integrals and the Taylor approximations of these integrals. To start we define variables and evaluate the integrals giving K and E by the statements

```
(%i1) Kint : 1/sqrt( 1 - k^2*sin(\phi)^2 );
```
Assign variable for K integrand.

(%o1) $$\frac{1}{\sqrt{1 - k^2 \sin^2 \phi}}$$

```
(%i2) Kser : taylor( Kint, k, 0, 5 );
```
Expand the K integrand in a Taylor Series.

(%o2) /T/ $$\frac{3}{8} k^4 \sin^4 \phi + \frac{1}{2} k^2 \sin^2 \phi + 1 + \ldots$$

```
(%i3) Kpoly : integrate(Kser, \phi, 0, %pi/2);
```
Integrate.

(%o3) $$\frac{9\pi k^4 + 16\pi k^2 + 64\pi}{128}$$

```
(%i4) Eint : sqrt(1 - k^2*sin( \Phi)^2 );
```
Assign variable for E integrand.

(%o4) $$\sqrt{1 - k^2 \sin^2 \phi}$$

```
(%i5) Eser : taylor( Eint, k, 0, 5 );
```
Expand the E integrand in a Taylor Series.

(%o5) /T/ $$-\frac{1}{8} k^4 \sin^4 \phi - \frac{1}{2} k^2 \sin \phi + 1 + \ldots$$

```
(%i6) Epoly : integrate(Eser, \phi, 0, %pi/2);
```
Integrate.

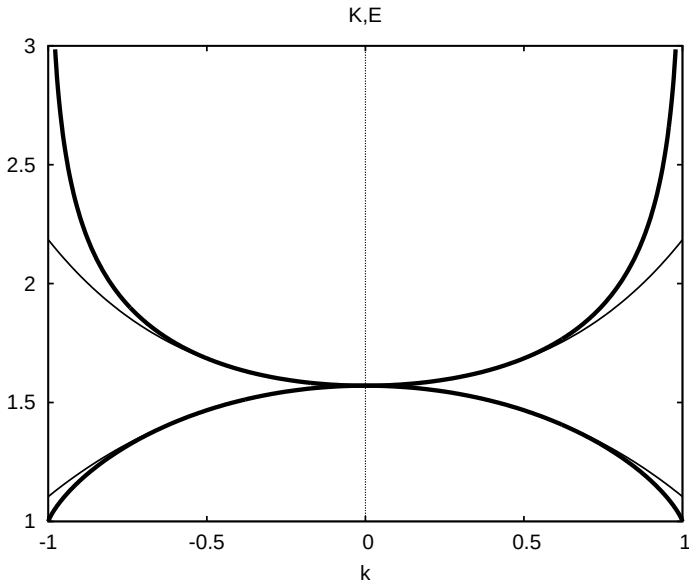
(%o6) $$-\frac{3\pi k^4 + 16\pi k^2 - 64\pi}{128}$$

Now we are ready to generate a graph showing not only the Taylor approximations but also the evaluations built in to MAXIMA. We invoke the single statement

```
(%i7) plot2d( [Kpoly, Epoly, elliptic_kc(k^2), elliptic_ec(k^2)], [k, -1.0, 1.0],
        [y, 1.0, 3.0], [legend, false], [title, "K,E"], [color, black],
        [style, [lines, 2], [lines, 2], [lines, 4], [lines, 4] ]  );
```

The display produced by this statement is shown in Fig. E6.13. It shows that the Taylor approximation doesn't quite agree with the more precise solution that MAXIMA provides for the integrals. This is because the Taylor series is built to approximate the equation very close to 0, so one shouldn't expect it to match for larger k. Even so, to the resolution of the graphs, the Taylor approximation to the order to which we have carried it appears to be good to about $k = 0.6$!

Figure E6.13: Complete elliptic integrals of the first and second kinds and their Taylor approximations. The full integrals are displayed with the heavier lines and the Taylor approximations are displayed with the lighter lines. The elliptic integral of the first kind K diverges at $k = \pm 1$ while that of the second kind E is well behaved at those critical points.



6.18 Laplace Transforms

Exercise: Find the Laplace transform of each of the functions

$$\text{(a)} \quad f(t) = t^n \quad \text{(b)} \quad f(t) = t\,e^{-at} \quad \text{(c)} \quad f(t) = \cosh(at) \quad \text{(d)} \quad \frac{df(t)}{dt}$$

and the inverse Laplace transform of each of the functions

$$\text{(e)} \quad \tilde{f}(s) = \frac{a + bs}{s^2 + \omega^2} \quad \text{(f)} \quad \tilde{f}(s) = \frac{a}{(s^2 + 9\omega^2)(s^2 + 4\omega^2)(s^2 + \omega^2)}$$

In (c), you may have to express the hyperbolic cosine in exponential form while in (f) you may have to help MAXIMA's `ilt` routine by first invoking the `partfrac` routine to expand the desired function in partial fractions.

Solution: (a) The Laplace transform of

$$f(t) = t^n$$

is obtained in MAXIMA with the statement

```
(%i1) laplace( t^n, t, s );
```

Is $n + 1$ positive, negative, or zero? `p;`

`(%o1)` $\Gamma(n+1)\,s^{-n-1}$

(b) For

$$f(t) = te^{-at}$$

the statement evaluating the Laplace transform is

```
(%i2) laplace(t*exp(-a*t), t, s );
```

`(%o2)` $\dfrac{1}{(s+a)^2}$

(c) For

$$f(t) = \cosh(at)$$

the statements evaluating the Laplace transform are

```
(%i3) laplace( cosh(a*t), t, s );
```

`(%o3)` $\dfrac{s}{s^2 - a^2}$

(d) The inverse Laplace transform of

$$\tilde{f}(s) = \frac{a + bs}{s^2 + \omega^2}$$

is obtained with the statements

```
(%i4) ilt( (a + b*s)/(s^2 + \omega^2), s, t );
```

Is ω zero or nonzero? `n;`

`(%o4)` $\dfrac{a\sin(\omega t)}{\omega} + b\cos(\omega t)$

(e) Finally, if

$$\tilde{f}(s) = \frac{a}{(s^2+9\omega^2)(s^2+4\omega^2)(s^2+\omega^2)}$$

we can obtain the inverse Laplace transform with the statements

```
(%i5) qs : a/((s^2 + 9*\omega^2)*(s^2 + 4*\omega^2)*(s^2 + \omega^2));
```

(%o5) $$\frac{a}{(s^2+9\omega^2)(s^2+4\omega^2)(s^2+\omega^2)}$$

```
(%i6) qt : ilt( qs, s, t );
```

Is ω zero or nonzero? `n;`

(%o6) $$\frac{a\sin(3\omega t)}{120\omega^5} - \frac{a\sin(2\omega t)}{30\omega^5} + \frac{a\sin(\omega t)}{24\omega^5}$$

Alternatively, we might obtain the result in a more conventional form by first expanding $\tilde{f}(s)$ in partial fractions in s^2, finding that

```
(%i7) qs1 : partfrac( qs, s );
```

(%o7) $$\frac{a}{40\omega^4(s^2+9\omega^2)} - \frac{a}{15\omega^4(s^2+4\omega^2)} + \frac{a}{24\omega^4(s^2+\omega^2)}$$

Then, remembering from the table in Section 2 of the chapter on solving ordinary differential equations that the inverse Laplace transform of $\omega/(s^2+\omega^2)$ is $\sin(\omega t)$, we recognize that each of the terms in this expression can be inverted separately to give a simple sine function with an appropriate (constant) multiplier and an appropriate argument. With this form of the transform, MAXIMA's inversion routine will automatically treat each term separately, and we find that

```
(%i8) qt1 : ilt( qs1, s, t );
```

Is ω zero or nonzero? `n;`

(%o8) $$\frac{a}{40\omega^4(s^2+9\omega^2)} - \frac{a}{15\omega^4(s^2+4\omega^2)} + \frac{a}{24\omega^4(s^2+\omega^2)}$$

6.19 Fitting Parabola Through Three Points

Exercise: Given the three points (x_i, y_i), $i = 1, 2, 3$, (a) find symbolic expressions for the coefficients a, b, and c of the parabola $y = ax^2 + bx + c$ that passes through these three points and then (b) find a symbolic expression for the value of x at which the extreme point of the parabola occurs. Finally, (c) determine numerically the angle at which the maximum range of a projectile occurs if the ranges at $\theta = 39°$, $40°$, and $41°$ are 0.7251744, 0.7259383, and 0.7258887, respectively.

Solution: (a) The equation $y = ax^2 + bx + c$, relates the five quantities x, y, a, b, and c. Given three points (x_i, y_i), $i = 1, 2, 3$, through which this parabola must pass, we can create three independent equations to solve for a, b, and c. The MAXIMA commands and their outputs used in accomplishing this are

```
(%i1) GenEq : y = a*x^2 + b*x + c$
```
Define the general equation.

```
(%i2) Eq1 : ev( GenEq, x = x1, y = y1 )$
```
Create an equation for point 1.

```
(%i3) Eq2 : ev( GenEq, x = x2, y = y2 )$
```
Create an equation for point 2.

```
(%i4) Eq3 : ev( GenEq, x = x3, y = y3 )$
```
Create an equation for point 3.

```
(%i5) soln : solve( [Eq1, Eq2, Eq3],
                    [a, b, c] );
```
Solve the system of linear equations for the coefficients a, b, and c.

(%o5)
$$\left[\left[a = \frac{x_1\,(y_3 - y_2) - x_2\,y_3 + x_3\,y_2 + (x_2 - x_3)\,y_1}{x_1\,(x_3^2 - x_2^2) - x_2\,x_3^2 + x_2^2\,x_3 + x_1^2\,(x_2 - x_3)},\right.\right.$$
$$b = -\frac{x_1^2\,(y_3 - y_2) - x_2^2\,y_3 + x_3^2\,y_2 + (x_2^2 - x_3^2)\,y_1}{x_1\,(x_3^2 - x_2^2) - x_2\,x_3^2 + x_2^2\,x_3 + x_1^2\,(x_2 - x_3)},$$
$$\left.\left.c = \frac{x_1\,(x_3^2\,y_2 - x_2^2\,y_3) + x_1^2\,(x_2\,y_3 - x_3\,y_2) + (x_2^2\,x_3 - x_2\,x_3^2)\,y_1}{x_1\,(x_3^2 - x_2^2) - x_2\,x_3^2 + x_2^2\,x_3 + x_1^2\,(x_2 - x_3)}\right]\right]$$

These parameters define the parabola passing through the three specified points.

(b) If we want to find where the maximum value of x is for the given parabola, we need only look at the singular point where the derivative of our parabolic equation is 0. We must assume of course that $a \neq 0$. Then we can solve this equation for x to find x_{max}. In MAXIMA the statements used for making the substitution and solving for x_{max} are

```
(%i6) NewGenEq : ev( GenEq, soln )$
```
Substitute the representations for the variables from `soln` obtained in part (a) to create the new equation.

```
(%i7) DiffGenEq : diff( NewGenEq, x )$
```
Differentiate that equation. (Here, since we did not specify y as a variable of x its derivative is 0).

```
(%i8) xmax : solve( DiffGenEq, x );
```
Solve the differentiated equation for x.

(%o8)
$$\left[\, x = \frac{(x_2^2 - x_1^2)\,y_3 + (x_1^2 - x_3^2)\,y_2 + (x_3^2 - x_2^2)\,y_1}{(2x_2 - 2x_1)\,y_3 + (2x_1 - 2x_3)\,y_2 + (2x_3 - 2x_2)\,y_1}\,\right]$$

(c) Upon inspecting the data given here, we expect the maximum range to occur at an initial angle somewhere between 39° and 41°. From (b) we know if we are given any three pairs of input and output we can find the extreme value of x on a parabola via direct substitution. We expect that, for the values listed, the maximum value will occur at an angle between 39° and 41°. To find the angle at which the maximum occurs for the values listed, we use the MAXIMA statement

```
(%i9) ev( rhs(xmax[1]), x1 = 39, x2 = 40, x3 = 41, y1 = 0.7251744, y2 = 0.7259383,
                        y3 = 0.7258887 );
(%o9)      40.43902888752648
```

The result tells us that, if we desire the maximum range, we should fire the projectile at an initial angle of 40.4390°. As we expected, the result falls between 39° and 41°.

6.20 Deducing Simpson's Rule

Exercise: Find symbolic expressions for the coefficients a, b, and c that will cause the parabola $y = ax^2 + bx + c$ to pass through the three points $(x_1 = x_2 - \delta x, y_1)$, (x_2, y_2), and $x_3 = x_2 + \delta x, y_3)$. Then integrate the parabola over the interval $x_1 \le x \le x_3$ and deduce Simpson's rule

$$\int_{x_1}^{x_3} y(x)\,dx = \frac{\delta x}{3}\Big(y_1 + 4y_2 + y_3\Big)$$

for (approximate) numerical integration.

Solution: In an expression containing five variables such as $y = ax^2 + bx + c$, we need at least three equations containing different explicit values for two of the variables. Here, we have three sets of points for x and y, so the task is to define the three equations for the three separate points, solve the system of linear equations for the variables a, b, and c, then integrate the new quadratic equation from x_1 to x_3. In MAXIMA this involves starting with the statements

```
(%i1) eqn1 : y1 = a * (x2 - dx)^2 + b*(x2 - dx) + c$
(%i2) eqn2 : y2 = a*x2^2 + b*x2 + c$
(%i3) eqn3 : y3 = a * (x2 + dx)^2 + b*(x2 + dx) + c$
(%i4) soln : solve( [eqn1, eqn2, eqn3], [a, b, c] );
```

$$\text{(\%o4)}\quad \left[\left[\, a = \frac{y_3 - 2\,y_2 + y_1}{2\,dx^2},\right.\right.$$
$$b = -\frac{x_2(\,2y_3 - 4y_2) - dx\,y_3 + (2x_2 + dx)\,y_1}{2\,dx^2},$$
$$\left.\left. c = \frac{x_2^2(\,y_3 - 2y_2) - dx\,x_2\,y_3 + 2\,dx^2\,y_2 + (x_2^2 + dx\,x_2)\,y_1}{2\,dx^2}\,\right]\right]$$

We have solved our system of linear equations. Now we need construct the approximating parabola $ax^2 + bx + c$ by subtituting the values given in `soln` for a, b, and c. To those ends, we invoke the statement

```
(%i5) y : ev( a*x^2 + b*x + c, soln )
```

$$\text{(\%o5)}\quad -\frac{x\,(x_2(2y_3 - 4y_2) - dx\,y_3 + (2x_3 + dx)\,y_1}{2\,dx^2}$$
$$+\frac{x_2^2(y_3 - 2y_2\,) - dx\,x_2\,y_3 + 2\,dx^2\,y_2 + (x_2^2 + dx\,x_2\,)\,y_1}{2\,dx^2}$$
$$+\frac{x^2(y_3 - 2\,y_2|y_1)}{2\,dx^2}$$

Finally, we integrate and recast this expression with the statements

```
(%i6) expand( integrate( y, x, x2 - dx, x2 + dx ) );
```

$$\text{(\%o6)}\quad \frac{dx\,y_3}{3} + \frac{4\,dx\,y_2}{3} + \frac{dx\,y_1}{3}$$

```
(%i7) factor( % );
```

$$\text{(\%o7)}\quad \frac{dx\,(\,y_1 + 4\,y_2 + y_3\,)}{3}$$

We have deduced Simpson's rule using MAXIMA.

6.21 Accuracy in Simpson's Rule

Exercise: The midpoint rule M and Simpson's rule S for evaluating integrals numerically start with the assumptions that

$$\int_a^b f(x)\,dx \approx M = (b-a)\,f\left(\frac{a+b}{2}\right) \quad ; \quad \int_a^b f(x)\,dx \approx S = \frac{b-a}{6}\left(f(a) + 4\,f\left(\frac{a+b}{2}\right) + f(b)\right)$$

respectively. To deduce the first of these expressions, we approximate $f(x)$ over the interval $a \le x \le b$ with a constant equal to the value of $f(x)$ at the midpoint of the interval while deducing the second entails approximating the function with a quadratic polynomial that passes through $f(x)$ at the endpoints and the midpoint of the interval. The midpoint rule will clearly be 100% accurate if $f(x)$ is a constant and Simpson's rule will be 100% accurate if $f(x)$ is a quadratic polynomial. Use symbolic manipulation to show that, surprisingly, the midpoint rule is in fact 100% accurate for the linear function $f(x) = mx + p$ and Simpson's rule is 100% accurate for the cubic polynomial $f(x) = cx^3 + dx^2 + ex + g$! *Optional*: Try to construct *geometric* arguments that would provide insight into the correctness of these analytic results.

Solution: The midpoint rule for evaluating integrals numerically is exact for linear functions, as we can show with the statements

```
(%i1) linf(x) := m*x + p$
(%i2) intab : expand( integrate( linf(x), x, a, b ) );
```

(%o2) $bp - ap + \dfrac{b^2 m}{2} - \dfrac{a^2 m}{2}$

```
(%i3) (b - a)*linf( (a + b)/2 )$
(%i4) intmid : expand( % );
```

(%o4) $bp - ap + \dfrac{b^2 m}{2} - \dfrac{a^2 m}{2}$

Clearly the result obtained by integrating the linear function is equivalent to the result obtained by applying the midpoint rule, though we could make this result even more evident either by subtracting one value from the other with the statement

```
(%i5) intmid - intab;
(%o5)     0
```

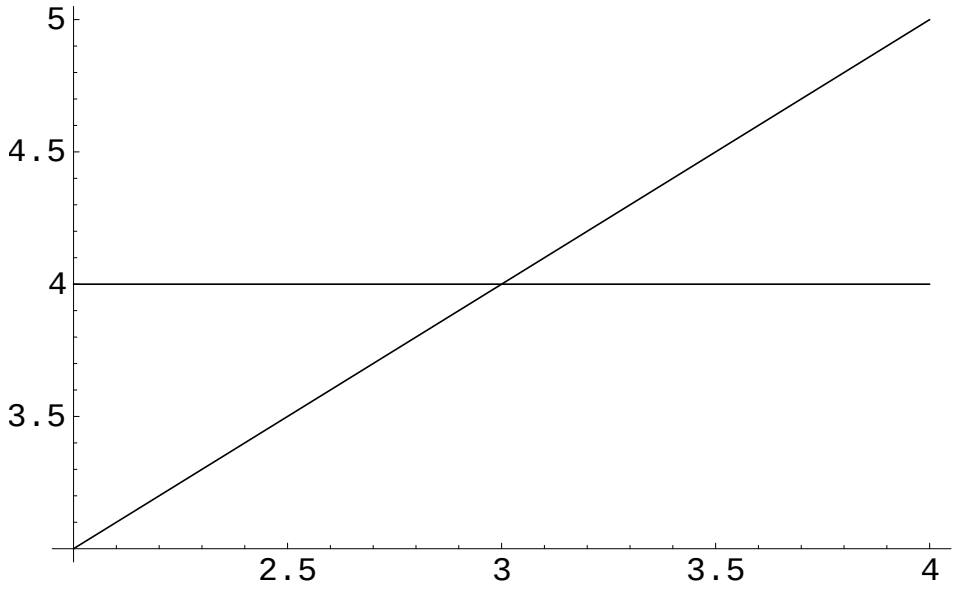
or by asking whether they are equal with the statement

```
(%i6) is( intmid = intab );
(%o6)     true
```

This result makes sense for a linear function. Geometrically, the value of the integral given by the midpoint rule is the area under the rectangle in Fig. E6.14 while the integral of the linear function is the area under the inclined line in that figure. The area under the horizontal line overestimates the area under the inclined line in the first half of the interval of integration and underestimates that area in the second half of the interval of integration, *but the overestimate in the first half of the interval is exactly the same as the underestimate in the second half of the interval.* The two errors cancel and the end result is *exactly* correct.

In the same way, Simpson's rule is exact for cubic polynomials, as we can check with the statements

Figure E6.14: Estimate of a linear function with the midpoint rule.




```
(%i7) cubf( x ) := c*x^3 + d*x^2 + e*x + g$
(%i8) intab : expand( integrate( cubf(x), x, a, b ) );
```

(%o8) $$bg - ag + \frac{b^2 e}{2} - \frac{a^2 e}{2} + \frac{b^3 d}{3} - \frac{a^3 d}{3} + \frac{b^4 c}{4} - \frac{a^4 c}{4}$$

```
(%i9) (b - a)*( cubf(a) + 4*cubf( (a + b)/2 ) + cubf(b))/6$
(%i10) intsimp : expand( % );
```

(%o10) $$bg - ag + \frac{b^2 e}{2} - \frac{a^2 e}{2} + \frac{b^3 d}{3} - \frac{a^3 d}{3} + \frac{b^4 c}{4} - \frac{a^4 c}{4}$$

and we see that the result found by integrating the polynomial is equivalent to the result obtained by applying Simpson's rule, though—again—we could make this result even more evident either by subtracting one value from the other with the statement

```
(%i11) intsimp - intab;
(%o11)     0
```

or by asking whether they are equal with the statement

```
(%i12) is( intsimp = intab );
(%o12)     true
```

Geometrically, Simpson's rule is 100% accurate for *cubic* polynomials for reasons similar to the reasons that the midpoint rule is 100% accurate for linear functions, but the property is much harder to "see" intuitively.

6.22 Potential Energy with a Bump

Exercise: Using MAXIMA, obtain graphs of the potential

$$U(x) = -\frac{U_0 a^2(a^2+x^2)}{8a^4+x^4}]$$

and the associated force. Note that the expression for the potential is simpler to plot if you plot not $U(x)$ versus x but rather $U(x)/U_0$ versus $\overline{x} = x/a$, i.e., rewrite the function in the form

$$\frac{U(x)}{U_0} = -\frac{1+\overline{x}^2}{8+\overline{x}^4}$$

You should be able to find a similar dimensionless version of the expression giving the force.

Solution: We first seek graphs of the potential energy

$$U(x) = -\frac{U_0 a^2(a^2+x^2)}{8a^4+x^4} \qquad \text{or} \qquad \frac{U(x)}{U_0} = -\frac{1+X^2}{8+X^4}$$

where, in the latter expression, we have introduced the dimensionless length $X = x/a$. The commands

```
(%i1) U : -(1+X^2)/(8+X^4)$
(%i2) plot2d(U,[X,-5.0,5.0])$
```

will ultimately produce a graph of $U(x)/U_0$ versus x/a, though that graph suggests the wisdom of plotting over a somewhat larger range of x and of wresting control of the y scaling particularly from the system. We try again with the command

```
(%i3) plot2d(U,[X,-10.0,10.0], [y,-0.3,0.0], [xlabel,"x/a"],
       [ylabel, "U/U0"], [grid2d,true], [color,black],
       [style, [lines,4]] );
```

The resulting graph is shown in Fig. E6.15.[13]

Going on now to find the force, we would take the negative derivative of the potential energy, $F_x = -dU/dx$. If, however, we introduce the quantities U/U_0 and $X = x/a$ in this derivative, we find that

$$F_x = -\frac{dU}{dx} = -\frac{U_0\, d(U/U_0)}{d(aX)} = -\frac{U_0}{a}\,\frac{d(U/U_0)}{dX} \qquad \Longrightarrow \qquad \frac{F_x}{U_0/a} = -\frac{d(U/U_0)}{dX}$$

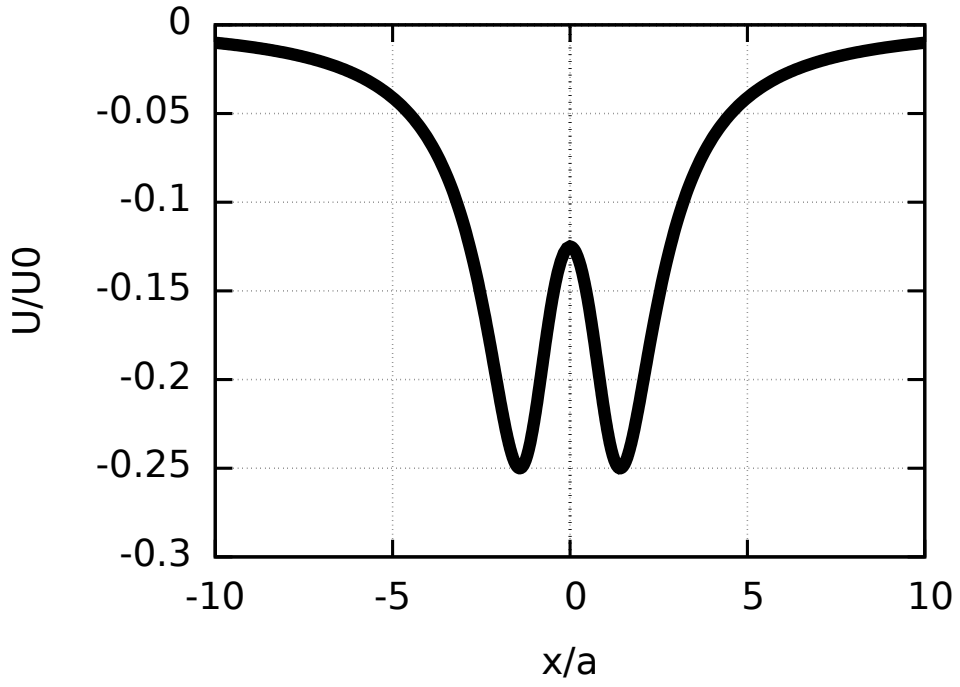
Evidently, if we measure force in units of U_0/a, then we can just differentiate the dimensionless version of the potential energy with respect to the dimensionless coordinate X. MAXIMA then yields the following results, though we must change the vertical scaling to plot the force in a way that reveals its behavior most easily.

```
(%i4) F: -diff(U,X);
```

(%o4) $$\frac{2X}{X^4+8} - \frac{4X^3(X^2+1)}{(X^4+8)^2}$$

[13] A description of the graph can be directed to a PDF or PostScript file by adding the option `[pdf_file, "figure.pdf"]` or `[ps_file, "figure.eps"]` to the `plot2d` command and re-executing the `plot2d` statement. These files will be placed in the default directory.

Figure E6.15: U/U_0 versus x/a. This graph was produced with MAXIMA.



```
(%i5) F : factor(F);
```

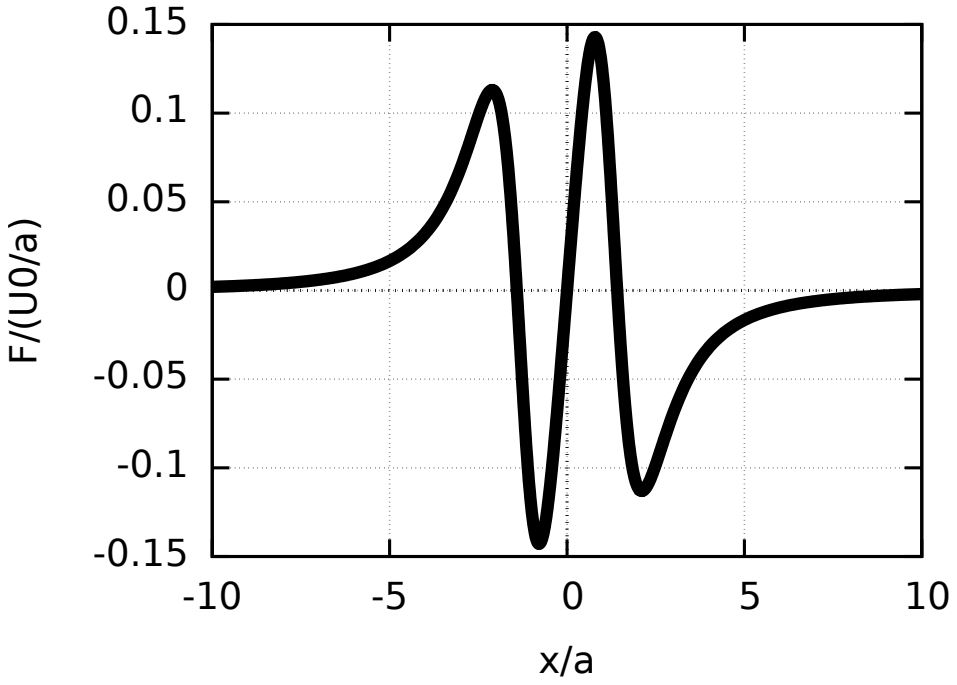
(%o5) $$-\frac{2X(X^2-2)(X^2+4)}{(X^4+8)^2}$$

```
(%i6) plot2d(F,[X,-10.0,10.0], [xlabel,"x/a"],
       [ylabel, "F/(U_0/a)"], [grid2d,true], [color,black],
       [style, [lines,4]] );
```

The result is shown in Fig. E6.16.[14]

[14] See previous footnote.

Figure E6.16: $F/(U_0 a)$ versus x/a. This graph was produced with MAXIMA.



6.23 Solving ODEs Symbolically

Exercise: Using MAXIMA, find complete (symbolic) solutions to each of the following problems, and use MAXIMA also to verify that the solutions you obtain actually *do* satisfy the original ODE and initial conditions.

a. $\dfrac{d^2x}{dt^2} = a$, $x(0) = x_0$, $v(0) = v_0$, where a is constant, i.e., find position as a function of time for a particle moving under the action of a constant force and launched with arbitrary initial conditions.

b. $m\dfrac{d^2x}{dt^2} = -eE_0\cos(\omega t + \theta)$, $x(0) = x_0$, $v(0) = v_0$, i.e., find position as a function of time for a charged particle moving under the action of a sinusoidal force and launched with arbitrary initial conditions.

c. $m\dfrac{d^2x}{dt^2} = -mg + b\left(\dfrac{dx}{dt}\right)^2$, $x(0) = 0$, $v(0) = 0$, i.e., find position as a function of time for a particle released from rest at the origin and allowed to fall freely under the action of gravity and a viscous retarding force proportional to the *square*of the speed.

d. the differential equations

$$m\frac{d^2x}{dt^2} = -b\frac{dx}{dt} \qquad ; \qquad m\frac{d^2z}{dt^2} = -mg - b\frac{dz}{dt}$$

for the motion of a projectile under gravity in a viscous medium when the motion starts at the origin with the initial velocity $\mathbf{v} = v_{x0}\,\hat{\mathbf{i}} + v_{z0}\,\hat{\mathbf{k}}$.

Solution: (a) The following MAXIMA statements will generate the solution to the problem defined by the differential equation and initial conditions

$$\frac{d^2x}{dt^2} = a \qquad ; \qquad x(0) = x_0 \qquad ; \qquad \frac{dx}{dt}(0) = v_0$$

Basically, in line `(%i1)`—output suppressed—we simply tell MAXIMA that x is to be viewed as a function of t; in line `%i2)` we define the equation, giving it the name `eqa`; in line `(%i3)` we request MAXIMA to solve the equation; and in line `(%i4)` we impose the initial conditions.

```
(%i1) depends(x,t)$
(%i2) eqa :  'diff(x,t,2) = a;
```

(eqa) $\dfrac{d^2}{dt^2}x = a$

```
(%i3) ode2(eqa,x,t);
```

(%o3) $x = \dfrac{at^2}{2} + \%\mathrm{k2}\,t + \%\mathrm{k1}$

```
(%i4) ic2(%o3, t = 0,x = x0,'diff(x,t) = v0);
```

(%o4) $x = x_0 + t\,v_0 + \dfrac{at^2}{2}$

The solution displayed in line `(%o4)` is readily recognized as satisfying both the original differential equation and the initial conditions.

(b) The following MAXIMA commands will generate the solution to the problem defined by the differential equation and initial conditions

$$m\frac{d^2x}{dt^2} = -eE_0\cos(\omega t + \theta) \qquad ; \qquad x(0) = x_0 \qquad ; \qquad \frac{dx}{dt}(0) = v_0$$

The steps are the same as those presented in part (a).[15] We define the equation, request its solution, and impose the initial conditions with the statements

```
(%i1) depends(x,t)$
(%i2) eqb : m*'diff(x,t,2) = -e*E0*cos(omega*t+theta);
```

(eqb) $$m\frac{d^2}{dt^2}x = -eE_0\cos(\theta+\omega t)$$

```
(%i3) ode2(eqb,x,t);
```

(%o3) $$x = \frac{eE_0\cos(\theta+\omega t)}{m\omega^2} + \%\mathrm{k2}\,t + \%\mathrm{k1}$$

```
(%i4) ic2(%,t = 0,x = x0,'diff(x,t) = v0);
```

(%o4) $$x = \frac{m\omega^2 x_0 - eE_0\cos\theta}{m\omega^2} + \frac{t(m\omega v_0 + eE_0\sin\theta)}{m\omega} + \frac{eE_0\cos(\theta+\omega t)}{m\omega^2}$$

Certainly the double integral of the original equation yields the result in line `(%o2)`, so we can mentally verify that piece of the solution. It is a bit harder to see directly that the solution in line `(%o3)` satisfies the given initial conditions, though the statements

```
(%i5) combine( ev(rhs(%o4),t=0) );
(%o5)     x0
(%i6) rhs( diff(%o4,t) )$
(%i7) combine( ev(%, t=0));
(%o7)     v0
```

reveal that satisfaction. Finally, we verify explicitly that the solution satisfies the differential equation with the statement

```
(%i8) m*diff(%o4,t,2);
```

(%o8) $$m\left(\frac{d^2}{dt^2}\right)x = -E_0 e\cos(\theta+\omega t)$$

Q.E.D.

(c) The problem defined by the differential equation and initial conditions[16]

$$m\frac{d^2x}{dt^2} = -mg + bv^2 \qquad ; \qquad x(0) = 0 \qquad ; \qquad \frac{dx}{dt}(0) = 0$$

is a bit more complicated. We again assert the statement of dependency—$x = x(t)$—and then define and solve the equation with the statements

```
(%i1) depends(x,t)$
(%i2) eqc : m*'diff(x,t,2) = -m*g+b*('diff(x,t))^2;
```

(eqc) $$m\frac{d^2}{dt^2}x = b\left(\frac{d}{dt}x\right)^2 - gm$$

```
(%i3) ode2(eqc,x,t);
Is b g m positive or negative?  p;
```

[15] We restart MAXIMA at the beginning of each new equation, so the statement labels start over again at `(%i1)` and we need to re-execute the `depends` statement.

[16] Note, incidentally, that v in this equation is the velocity, dx/dt, not a constant parameter.

(%o3)

$$x = \%k2 - \sqrt{bgm}\left(\frac{m\log\left(e^{\frac{2\sqrt{bgm}\,t}{m}+\frac{2\,\%k1\,\sqrt{bgm}}{m}}-1\right)}{b\sqrt{bgm}} - \frac{m\left(\frac{2\sqrt{bgm}\,t}{m}+\frac{2\,\%k1\,\sqrt{bgm}}{m}\right)}{2b\sqrt{bgm}}\right)$$

```
(%i4) ic2(%,t = 0,x = 0,'diff(x,t) = 0);
```

(%o4) []

Evidently, the attempt to impose the initial conditions has this time failed.

In an effort to make progress, let us simplify the expression a bit. First, recognize that the quantity $\alpha = \sqrt{bgm}/m = \sqrt{bg/m}$ appears in several places. The solution will have a much simpler form if we substitute a single symbol for that quantity with the statement

```
(%i5) subst(alpha*m,sqrt(b*g*m),%o3);
```

(%o5) $$x = \%k2 - \alpha m\left(\frac{\log(e^{2\alpha t+2\,\%k1\,\alpha}-1}{\alpha b} - \frac{2\alpha t+2\,\%k1\,\alpha}{2\alpha b}\right)$$

Here, α is dimensionally a frequency. In this form, it is now evident that $\%k1$ appears only in the combination $\%k1\,\alpha$. Nothing prevents us from declaring that the quantity $\%k1\,\alpha$ is the integration constant instead, since it is really only that combination that the initial conditions will determine anyway. That additional simplification is accomplished with the statements

```
(%i6) subst(2*%k1,2*%k1*alpha,%);
```

(%o6) $$x = \%k2 - \alpha m\left(\frac{\log(e^{2\alpha t+2\,\%k1}-1)}{\alpha b} - \frac{2\alpha t+2\,\%k1}{2\alpha b}\right)$$

```
(%i7) expand(%);
```

(%o7) $$x = -\frac{m\log(e^{2\alpha t+2\,\%k1}-1)}{b} + \frac{\alpha m t}{b} + \frac{\%k1\,\alpha\,m}{b} + \%k2$$

Next, let us introduce the simpler variable $\beta = m/b$ with the statement

```
(%o8) subst( beta*b, m, %);
```

(%o8) $$x = -\beta\log(e^{2\alpha t+2\,\%k1}-1) + \alpha\beta t + \%k1\,\beta + \%k2$$

Now, note that $\%k1\,\alpha\beta + \%k2$ is an unknown constant. nothing prevents us from renaming that constant to $\%k2$ with the statement

```
(%i9) subst(%k2 - %k1*beta, %k2, %);
```

(%o9) $$x = -\beta\log(e^{2\alpha t+2\,\%k1}-1) + \alpha\beta t + \%k2$$

We redefine $\%k1$ once more, setting the constant $2\,\%k1$ to $\log(\%k1)$ with the statement

```
(%i10) subst(log(%k1)/2, %k1,%);
```

(%o10) $$x = -\beta\log(\%k1\,e^{2\alpha t}-1) + \alpha\beta t + \%k2$$

Finally, we are ready to invoke `ic2` to determine the integration constants with the statement

```
(%i11) ic2(%,t = 0,x = 0,'diff(x,t) = 0);
```

(%o11) $x = -\beta \log(-e^{2\alpha t} - 1) + \alpha\beta t + \log(-2)\beta$

The logarithms of negative numbers look strange, though we note there are *two* of them. That problem will disappear if the two logarithmic terms are combined with the statement

```
(%i12) soln : logcontract(%);
```

(soln) $x = \beta \log\left(\frac{2}{e^{2\alpha t} + 1}\right) + \alpha\beta t$

We can then verify the satisfaction of the initial conditions, evaluate the velocity, and verify its limit as t goes towards ∞ with the statements

```
(%i13) ev(rhs(soln),t = 0);
(%o13)     0
(%i14) v : diff(rhs(soln),t);
```

(v) $\alpha\beta - \frac{2\alpha\beta e^{2\alpha t}}{e^{2\alpha t} + 1}$

```
(%i15) ev(v,t=0);
(%o15)     0
(%i16) limit(v, t, inf);
Is α positive, negative, or zero? p;
(%o16)     −αβ
```

As the statements

```
(%i17) subst(-alpha*beta, v, -m*g+b*v^2);
(%o17)     α²bβ² − gm
(%i18) subst(m/b, beta, %);
```

(%o18) $\frac{\alpha^2 m^2}{b} - gm$

```
(%i19) subst( sqrt(b*g/m), alpha, % );
(%o19)     0
```

show, the value $v_\infty = -\alpha\beta$ is exactly the limiting velocity that will result in the force $-mg + bv_\infty^2$ on the object assuming the value zero and the object experiences no further acceleration.

To verify that the solution `soln` in fact satisfies the differential equation, we evaluate the left `lside` and right `rside` sides separately, in the process casting all constants in terms of α, β, and b, recognizing that $m = b\beta$ and $g = \alpha^2\beta$. The statements

```
(%i20) lside : ev( factor( m*diff(rhs(soln),t,2) ), m=b*beta);
```

(lside) $-\frac{4\alpha^2 b\beta^2\, e^{2\alpha t}}{(e^{2\alpha t} + 1)^2}$

```
(%i21) rside : ev( factor(-m*g + b *diff(rhs(soln),t)^2), m=b*beta)$
(%i22) rside : ev( rside, g=alpha^2*beta );
```

(rside) $-\frac{4\alpha^2 b\beta^2\, e^{2\alpha t}}{(e^{2\alpha t} + 1)^2}$

Finally, we test the equality of the left and right sides with the statement

```
(%i23) is( lside = rside );
(%o23)     true
```

Q.E.D.

Rather simpler mathematics might have resulted had we begun by dividing the equation by m and made a simplifying substitution right at the beginning. Let us go back and redefine `eqc`, make a substitution that reduces the number of parameters, and then attempt a solution with `ode2`, using the statements

```
(%i24) expand(eqc/m);
```

$$\text{(\%o24)} \qquad \frac{d^2}{dt^2}x = \frac{b}{m}\left(\frac{d}{dt}x\right)^2 - g$$

```
(%i25) eqc: subst(beta*b,m,%);
```

$$\text{(\%o25)} \qquad \frac{d^2}{dt^2}x = \frac{1}{\beta}\left(\frac{d}{dt}x\right)^2 - g$$

```
(%i26) ode2(eqc,x,t);
Is βg positive or negative?  p;
```

$$\text{(\%o26)} \qquad x = \%k2 - \sqrt{\beta g}\left(\frac{\beta\log\left(\,e^{(2\sqrt{\beta g}\,t+2\,\%k1\,\sqrt{\beta g})/\beta}-1\right)}{\sqrt{\beta g}} - t - \%k1\right)$$

which is very similar to the result at line (%o5) and will require much the same manipulations to find a satisfactory final answer.

Perhaps we would be better off if we treated the equation as a pair of first order equations, leading MAXIMA a bit more by the hand. Let us introduce the *first* order equation

$$m\frac{d}{dt}v = -mg + bv^2 \qquad \Longrightarrow \qquad \frac{dv}{dt} = -g + \frac{v^2}{\beta}$$

and then, subsequently, treat the result for v as dx/dt. The commands would be something like the following:

```
(%i27) depends(v,t)$
(%i28) eqc1 : 'diff(v,t) = -g + v^2/beta;
```

$$\text{(\%o28)} \qquad \frac{d}{dt}v = \frac{v^2}{\beta} - g$$

```
(ı29) ode2(eqc1,v,t);
Is βg positive or negative?  p;
```

$$\text{(\%o29)} \qquad \frac{\beta\log\left(-\dfrac{\sqrt{\beta g}-v}{v+\sqrt{\beta g}}\right)}{2\sqrt{\beta g}} = t + \%c$$

```
(%i30) ic1(%,t=0,v=0);
```

$$\text{(\%o30)} \qquad \frac{\beta\log\left(-\dfrac{\sqrt{\beta g}-v}{v+\sqrt{\beta g}}\right)}{2\sqrt{\beta g}} = \frac{2\sqrt{\beta g}\,t + \log(-1)\beta}{2\sqrt{\beta g}}$$

```
(%i31) solve(%,v);
```

$$\text{(\%o31)} \qquad \left[v = -\frac{\sqrt{\beta g}\left(e^{2\sqrt{\beta g}\,t/\beta}-1\right)}{e^{2\sqrt{\beta g}\,t/\beta}+1}\right]$$

Finally, we construct the differential equation $dx/dt = v$ and solve it as a first order equation:

```
(%i32) depends(x,t)$
(%i33) 'diff(x,t)=rhs(part(%o30,1));
```

$$\text{(\%o33)} \qquad \frac{d}{dt}x = -\frac{\sqrt{\beta g}\left(e^{2\sqrt{\beta g}\,t/\beta}-1\right)}{e^{2\sqrt{\beta g}\,t/\beta}+1}$$

```
(%i34) ode2(%,x,t);
```

$$\text{(\%o34)} \qquad x = \%c - \sqrt{\beta g}\left(\frac{\beta\log(e^{2\sqrt{\beta g}\,t/\beta}+1)}{\sqrt{\beta g}} - t\right)$$

```
(%i35) ic1(%,t=0,x=0);
```

$$\text{(\%o35)} \qquad x = -\frac{\sqrt{\beta g}\left(\beta\log(e^{2\sqrt{\beta g}\,t/\beta}+1)-\log(2)\beta\right)-\beta g t}{\sqrt{\beta g}}$$

```
(%i36) logcontract(%);
```

$$\text{(\%o36)} \qquad x = \frac{\beta\sqrt{\beta g}\log\left(\dfrac{2}{e^{2\sqrt{\beta g}\,t/\beta}+1}\right)+\beta g t}{\sqrt{\beta g}}$$

```
(%i37) expand(%);
```

$$\text{(\%o37)} \qquad x = \beta\log\left(\frac{2}{e^{2\sqrt{\beta g}\,t/\beta}+1}\right)+\frac{\beta g t}{\sqrt{\beta g}}$$

Remember that $\beta = m/b$ and $\alpha = \sqrt{g/\beta}$. Then, making sure MAXIMA understands that certain quantities are positive, the statements

```
(%i38) assume(alpha>0, beta>0, g>0);
(%o38)      [α > 0, β > 0, g > 0]
(%i39) expand( subst(alpha^2*beta, g, %o36) );
```

$$\text{(\%o39)} \qquad x = \beta\log\left(\frac{2}{e^{2\alpha t}+1}\right)+\alpha\beta t$$

convert the present solution into exactly the solution at line `(%o12)`, though the line here is in fact labeled `(soln)`.

(d) The problem defined by the differential equations

$$m\frac{d^2x}{dt^2} = -b\frac{dx}{dt} \quad ; \quad m\frac{d^2z}{dt^2} = -mg - b\frac{dz}{dt}$$

and the initial conditions

$$x(0) = 0 \quad ; \quad z(0) = 0 \quad ; \quad \frac{dx}{dt}(0) = v_{x0} \quad ; \quad \frac{dz}{dt}(0) = v_{z0}$$

is much more straightforward than the problem of part (c). We simply define the equations, solve them, and determine the integration constants to match the initial conditions with the statements

```
(%i1) depends(x,t)$
(%i2) eqdx : m*'diff(x,t,2) = -b*'diff(x,t);
```

(eqdx) $$m\left(\frac{d^2}{dt^2}x\right) = -b\left(\frac{d}{dt}x\right)$$

```
(%i3) ode2(eqdx,x,t);
Is b zero or nonzero?  n;
```

(%o3) $$x = \%k2\, e^{-bt/m} + \%k1$$

```
(%i4) ic2(%,t = 0,x = 0,'diff(x,t) = vx0);
```

(%o4) $$x = \frac{mv_{x0}}{b} - \frac{me^{-bt/m}v_{x0}}{b}$$

```
(%i5) depends(z,t)$
(%i6) eqdz : m*'diff(z,t,2) = -m*g-b*'diff(z,t);
```

(%o6) $$m\left(\frac{d^2}{dt^2}z\right) = -b\left(\frac{d}{dt}z\right) - gm$$

```
(%i7) ode2(%,z,t);
Is b zero or nonzero?  n;
```

(%o7) $$z = \%k2\, e^{-bt/m} - \frac{bgmt - gm^2}{b^2} + \%k1$$

```
(%i8) ic2(%,t = 0,z = 0,'diff(z,t) = vz0);
```

(%o8) $$z = -\frac{e^{-bt/m}(bmv_{z0} + gm^2)}{b^2} + \frac{mv_{z0}}{b} - \frac{bgmt - gm^2}{b^2}$$

We verify explicitly that these solutions satisfy the initial conditions with the statements

```
(%i9) solnx : rhs(%o4)$
(%i10) solnz : rhs(%o8)$
(%i11) ev(solnx,t=0);
%o11)     0
(%i12) ev(solnz,t=0);
%o12)     0
(%i14) diff(solnx,t)$ ev(%,t=0);
(%o14)      v_x0
(%i16) diff(solnz,t)$ expand( ev(%,t=0) );
(%o16)      v_z0
```

and we verify that the solutions satisfy the original differential equations with the statements

```
(%i17) is( m*diff(solnx,t,2) = -b*diff(solnx,t) );
(%o17)      true
(%i18) is( expand(m*diff(solnz,t,2) = -m*g-b*diff(solnz,t)) );
(%o18)      true
```

Q.E.D.

6.25 Vector Derivatives in Cylindrical Coordinates

Exercise: Paralleling your development after Section 6.7.10, develop the command file `MAXvectcyl.mac` to provide functions `gradcyl`, `lapcyl`, `divcyl`, and `curlcyl` to return the derivatives

$$\begin{aligned}
\nabla f(r,\phi,z) &= \frac{\partial f}{\partial r}\,\hat{\mathbf{r}} + \frac{1}{r}\frac{\partial f}{\partial \phi}\hat{\boldsymbol{\phi}} + \frac{\partial f}{\partial z}\,\hat{\mathbf{k}} \\
\nabla^2 f(r,\phi,z) &= \frac{1}{r}\frac{\partial}{\partial r}\left(r\frac{\partial f}{\partial r}\right) + \frac{1}{r^2}\frac{\partial^2 f}{\partial \phi^2} + \frac{\partial^2 f}{\partial z^2} \\
\nabla \bullet \mathbf{V}(r,\phi,z) &= = \frac{1}{r}\frac{\partial (rV_r)}{\partial r} + \frac{1}{r}\frac{\partial V_\phi}{\partial \phi} + \frac{\partial V_z}{\partial z} \\
\nabla \times \mathbf{Q}(r,\phi,z) &= \left(\frac{1}{r}\frac{\partial Q_z}{\partial \phi} - \frac{\partial Q_\phi}{\partial z}\right)\hat{\mathbf{r}} + \left(\frac{\partial Q_r}{\partial z} - \frac{\partial Q_z}{\partial r}\right)\hat{\boldsymbol{\phi}} + \frac{1}{r}\left(\frac{\partial (rQ_\phi)}{\partial r} - \frac{\partial Q_r}{\partial \phi}\right)\hat{\mathbf{k}}
\end{aligned}$$

for evaluating these entities in cylindrical coordinates. Here, r, ϕ, and z are the (cylindrical) radial, azimuthal, and axial coordinates. Then,

a. Evaluate the gradient of (i.e., the negative of the force field associated with) each of the functions

$$V_1(r,\phi,z) = \frac{1}{(r^2+z^2)^{1/2}} \qquad V_2(r,\phi,z) = \frac{z}{(r^2+z^2)^{3/2}}$$

$$V_3(r,\phi,z) = \frac{2z^2-r^2}{(r^2+z^2)^{5/2}} \qquad V_4(r,\phi,z) = \frac{e^{-a(r^2+z^2)^{1/2}}}{(r^2+z^2)^{1/2}}$$

of the cylindrical variables.

b. Verify that each field obtained in part (a) is conservative by showing that the curl of each is zero.

c. With $\mathbf{r} = r\,\hat{\mathbf{r}} + z\,\hat{\mathbf{k}}$, show (1) that $\nabla \times \mathbf{r} = 0$ and (2) that $\nabla \bullet \mathbf{r} = 3$.

d. With $V_5(r,\phi,z) = r^2 + z^2$, show that $\nabla^2 V_5(r,\phi,z) = \nabla \bullet \nabla V_5(r,\phi,z) = 6$.

e. Show that $\nabla^2 V_1(r,\phi,z) = 0$. (*Note*: Except at $r = z = 0$.)

f. Evaluate the Laplacian of $V_4(r,\phi,z)$, $\nabla^2 V_4(r,\phi,z)$.

Solution: We begin by creating the command file `MAXvectcyl.mac` listed in Table 6.1 to evaluate the derivatives identified in the problem statement. Each line translates the corresponding definition about into the language of MAXIMA. Then, we start MAXIMA and, with the default directory set to the directory containing the command file, we execute the statement

```
(%i1) batchload( "Maxvectcyl.mac" ) $
```

to define the four functions for use in the reminder of this solution.

(a) We evaluate the several gradients with the statements

```
(%i1) gradV1 : gradcyl( 1/sqrt(r^2+z^2), r, phi, z );
```

(gradV1) $$\left[-\frac{r}{(z^2+r^2)^{3/2}}, 0, -\frac{z}{(z^2+r^2)^{3/2}}\right]$$

```
(%i2) gradV2 : gradcyl( z/(r^2+z^2)^(3/2), r, phi, z );
```

Table 6.1: The command file `MAXvectcyl.mac`.

```
/* MAXvectcyl.MAXIMA (UNIX) or MAXvectcyl.mac (Windows)

    Command file to define functions for evaluating the
    gradient and Laplacian of a scalar function and the
    divergence and curl of a vector function of three
    cartesian variables. */

gradcyl(f, r,phi,z ) := [diff(f,r), diff(f,phi)/r, diff(f,z)]$
lapcyl(f, r,phi,z ) := diff(r*diff(f,r),r)/r + diff(f, phi,2 )/r^2 +
                       diff(f,z,2) $
divcyl(V, r,phi,z) := diff(r*V[1],r)/r + diff(V[2],phi)/r +
                      diff(V[3],z) $
curlcyl(V, r,phi,z ) := [ diff(V[3],phi)/r - diff(V[2],z),
                          diff(V[1],z) - diff(V[3],r),
                          (diff(r*V[2],r) - diff(V[1],phi))/r ] $
```

(gradV2) $$\left[-\frac{3rz}{(z^2+r^2)^{5/2}}, 0, \frac{1}{(z^2+r^2)^{3/2}} - \frac{3z^2}{(z^2+r^2)^{5/2}}\right]$$

```
(%i3) gradV3 : gradcyl( (2*z^2-r^2)/(r^2+z^2)^(5/2), r, phi, z );
```

(gradV3) $$\left[-\frac{2r}{(z^2+r^2)^{5/2}} - \frac{5r(2z^2-r^2)}{(z^2+r^2)^{7/2}}, 0, \frac{4z}{(z^2+r^2)^{5/2}} - \frac{5z(2z^2-r^2)}{(z^2+r^2)^{7/2}}\right]$$

```
(%i4) gradV4 : gradcyl( exp(-a*sqrt(r^2+z^2))/sqrt(r^2+z^2), r, phi, z );
```

(gradV4) $$\left[-\frac{ar}{z^2+r^2} - \frac{r}{(z^2+r^2)^{3/2}}, 0, -\frac{az}{z^2+r^2} - \frac{z}{(z^2+r^2)^{3/2}}\right] e^{-a\sqrt{z^2+r^2}}$$

(b) Next, we evaluate the curl of each of the results in part (a) with the statements

```
(i5%) curlcyl(gradV1,r, phi, z );
(o5%)     [0,0,0]

(i6%) curlcyl(gradV2,r, phi, z );
(o6%)     [0,0,0]

(i7%) curlcyl(gradV3,r, phi, z );
(o7%)     [0,0,0]

(i8%) curlcyl(gradV4,r, phi, z );
(o8%)     [0,0,0]
```

Clearly, each of the potentials V_1, V_2, V_3 and V_4 is gives rise to a conservative force field.

(c) The curl and divergence of the position vector are evaluated with the statements

```
(i9%) curlcyl( [r,0,z], r, phi, z );
(o9%)     [0,0,0]

(i10%) divcyl( [r,0,z], r, phi, z );
(o10%)     3
```

(d) The Laplacian of V_5, which is the square of the distance of the point from the origin, is addressed with the statement

```
(i10%) lapcyl( r^2+z^2, r, phi, z );
(o10%)     6
```

or with the statement

```
(i11%) divcyl( gradcyl(r^2+z^2, r, phi, z ), r, phi, z );
(o10%)     6
```

(e) Since $\nabla^2 V_1 = \nabla \bullet \nabla V_1$ and ∇V_1 was evaluated as `gradV1` above, the single statement

```
(i12%) divcyl(gradV1, r, phi, z );
(o12%)      A bit of a mess
(i13%) factor(%);
(o13%)      0
```

Note that it took a few tries before finding that the command `factor` achieved the desired simplification.

(f) Similarly to (e), $\nabla^2 V_4 = \nabla \bullet \nabla V_4$ and ∇V_4 was evaluated as `gradV4` above, the single statement

```
(i14%) divcyl(gradV4, r, phi, z );
(o14%)      A bit of a mess
(%i15) factor(%);
```

$$\text{(o15\%)} \qquad \frac{a^2\, e^{-a\sqrt{(z^2+r^2}}}{\sqrt{(z^2+r^2)}}$$

Alternatively, we could use the function `lapcyl` with the statement

```
(i16%) V4 : exp(-a*sqrt(r^2+z^2))/sqrt(r^2+z^2)$
(i16%) lapcyl( V4, r, phi, z );
(o16%)      A bit of a mess
(%i17) factor(%);
```

$$\text{(o17\%)} \qquad \frac{a^2\, e^{-a\sqrt{(z^2+r^2}}}{\sqrt{(z^2+r^2)}}$$

6.27 Vector Product Identities

Exercise: Create four three-component vectors `A`, `B`, `C`, and `D` with statements like

```
A: [A[1], A[2], A[3]]$ B: [B[1],B[2],B[3]]$
C: [C[1],C[2],C[3]]$ D: [D[1],D[2],D[3]]$
```

Then, using the file `crossdot.mac` as described in Section 6.9, show that

a. $\mathbf{A}\times\mathbf{B} = -\mathbf{B}\times\mathbf{A}$, or, equivalently, that $\mathbf{A}\times\mathbf{B} + \mathbf{B}\times\mathbf{A} = 0$.

b. $\mathbf{A}\times\big(\mathbf{B}\times\mathbf{C}\big) = \big(\mathbf{A}\bullet\mathbf{C}\big)\mathbf{B} - \big(\mathbf{A}\bullet\mathbf{B}\big)\mathbf{C}$

c. $\big(\mathbf{A}\times\mathbf{B}\big)\times\mathbf{C} = \big(\mathbf{A}\bullet\mathbf{C}\big)\mathbf{B} - \big(\mathbf{B}\bullet\mathbf{C}\big)\mathbf{A}$

d. $\mathbf{A}\bullet\big(\mathbf{B}\times\mathbf{C}\big) = \big(\mathbf{A}\times\mathbf{B}\big)\bullet\mathbf{C}$

e. $\big(\mathbf{A}\times\mathbf{B}\big)\bullet\big(\mathbf{C}\times\mathbf{D}\big) = \big(\mathbf{A}\bullet\mathbf{C}\big)\big(\mathbf{B}\bullet\mathbf{D}\big) - \big(\mathbf{A}\bullet\mathbf{D}\big)\big(\mathbf{B}\bullet\mathbf{C}\big)$

Solution: After launching MAXIMA and setting the default directory to the directory containing the desired file, we load the MAXIMA command file `crossdot.mac` and define the four vectors with the statements[17]

```
(%i1) batchload( "crossdot.mac" );
(%i5) A: [A[1], A[2], A[3]]$ B: [B[1],B[2],B[3]]$
      C: [C[1],C[2],C[3]]$ D: [D[1],D[2],D[3]]$
```

a. We evaluate the two cross products with the statements

```
(%i6) cross1 : cross(A,B);
(cross1)     [A2B3 - A3B2, A3B1 - A1B3, A1B2 - A2B1]
(%i7) cross2 : cross(B,A);
(cross2)     [A3B2 - A2B3, A1B3 - A3B1, A2B1 - A1B2]
```

and test the assertion that $\mathbf{A}\times\mathbf{B} + \mathbf{B}\times\mathbf{A} = 0$ with the statement

```
(%i8) cross1 + cross2;
(%o8) [0,0,0]
```

clearly verifying that $\mathbf{A}\times\mathbf{B} = -\mathbf{B}\times\mathbf{A}$.

b. In a similar vein, the statements

```
(%i9) pt1 : cross( A, cross(B,C))$
(%i10) pt2 : dot(A,C)*B - dot(A,B)*C$
(%i11) expand( pt1 - pt2 );
(%o11) [0,0,0]
```

verify that $\mathbf{A}\times\big(\mathbf{B}\times\mathbf{C}\big) = \big(\mathbf{A}\bullet\mathbf{C}\big)\mathbf{B} - \big(\mathbf{A}\bullet\mathbf{B}\big)\mathbf{C}$

c. To verify that $\big(\mathbf{A}\times\mathbf{B}\big)\times\mathbf{C} = \big(\mathbf{A}\bullet\mathbf{C}\big)\mathbf{B} - \big(\mathbf{B}\bullet\mathbf{C}\big)\mathbf{A}$ we invoke the statements

```
(%i12) exp1 : cross (cross(A,B), C )$
(%i13) exp2 : dot(A,C)*B - dot(B,C)*A$
(%i14) expand( exp1 - exp2 );
(%o14) [0,0,0]
```

[17] The first line of `%i5` is terminated with a simple ⟨RETURN⟩ so MAXIMA sees the four definitions as a single line.

Comparing parts (b) and (c), we confirm that the position of the parentheses in this so-called triple cross product is important. This particular triple product is *not* associative, i.e., $\mathbf{A} \times (\mathbf{B} \times \mathbf{C}) \neq (\mathbf{A} \times \mathbf{B}) \times \mathbf{C}$.

d. To verify that $\mathbf{A} \bullet (\mathbf{B} \times \mathbf{C}) = (\mathbf{A} \times \mathbf{B}) \bullet \mathbf{C}$ we invoke the statements

```
(%i15) exp3 : dot( A, cross(B,C) )$
(%i16) exp4 : dot( cross(A,B), C )$
(%i17) expand( exp3 - exp4 );
(%o17) 0
```

Q.E.D.

e. Finally, to verify that $(\mathbf{A} \times \mathbf{B}) \bullet (\mathbf{C} \times \mathbf{D}) = (\mathbf{A} \bullet \mathbf{C})(\mathbf{B} \bullet \mathbf{D}) - (\mathbf{A} \bullet \mathbf{D})(\mathbf{B} \bullet \mathbf{C})$ we invoke the statements

```
(%i18) exp5 : dot( cross(A,B), cross(C,D) )$
(%i19) exp6 : dot(A,C)*dot(B,D) - dot(A,D)*dot(B,C)$
(%i20) expand( exp5 - exp6 );
(%o20) 0
```

Q.E.D. Note that this identity can alternatively be written in the form

$$(\mathbf{A} \times \mathbf{B}) \bullet (\mathbf{C} \times \mathbf{D}) = \begin{vmatrix} \mathbf{A} \bullet \mathbf{C} & \mathbf{A} \bullet \mathbf{D} \\ \mathbf{B} \bullet \mathbf{C} & \mathbf{B} \bullet \mathbf{D} \end{vmatrix}$$

Chapter 9

Introduction to Programming

9.2 Relationship between IF-THEN-ELSE and CASE

Exercise: Figure E9.1 shows three different alternative structures. Express each structure using (a) *only* `CASE` structures and (b) *only* `IF-THEN-ELSE` structures. In these figures, `T`, `F`, `C`, and `B` stand for true, false, condition, and block of statements, respectively. Use proper indentation as illustrated in the examples.

Solution: (a) In the pseudocode of Chapter 9, this figure translates directly into the nested IF-THEN-ELSE control structure

```
IF C1 THEN B1
      ELSE IF C2 THEN B2
                 ELSE B3
           END_IF
END_IF
```

Equivalently, using the CASE structure, we would write

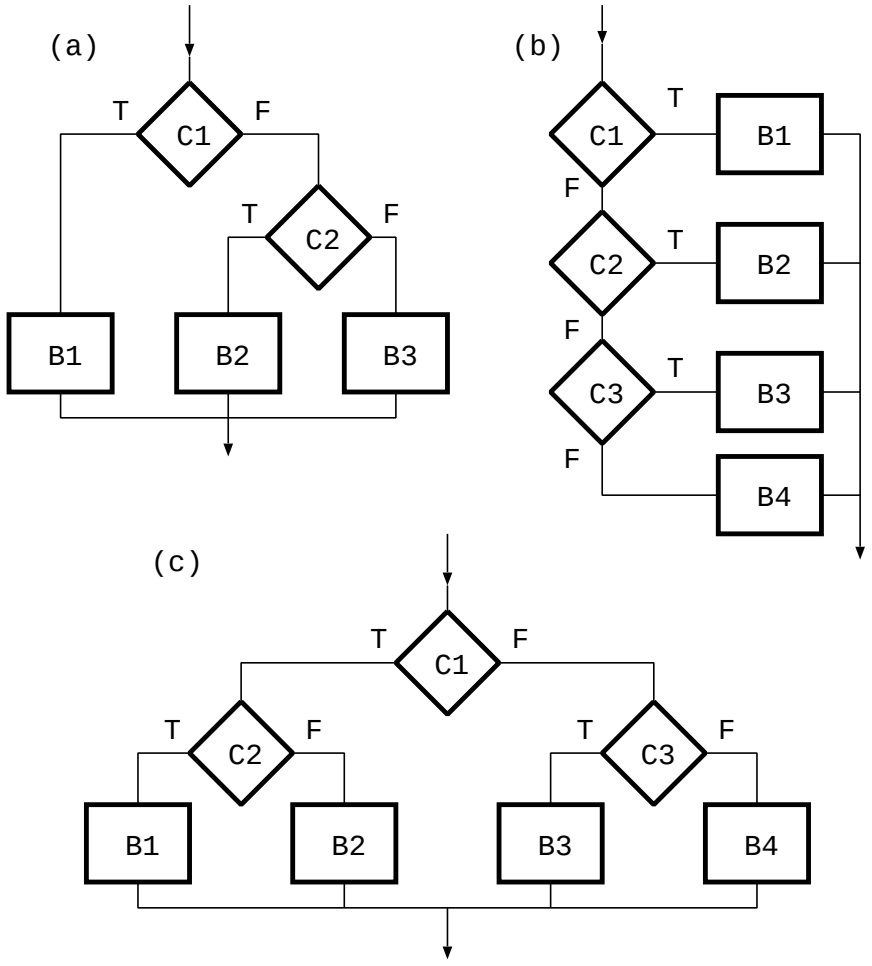
```
CASE
     OF C1 DO B1
     OF C2 DO B2
     OF OTHERS DO B3
END_CASE
```

(b) In the pseudocode of Chapter 9, this figure translates directly into the CASE structure

```
CASE
     OF C1 DO B1
     OF C2 DO B2
     OF C3 DO B3
     OF OTHERS DO B4
END_CASE
```

Equivalently, using the IF-THEN-ELSE structure, we would write

Figure E9.1: Figure for Exercise 9.2.

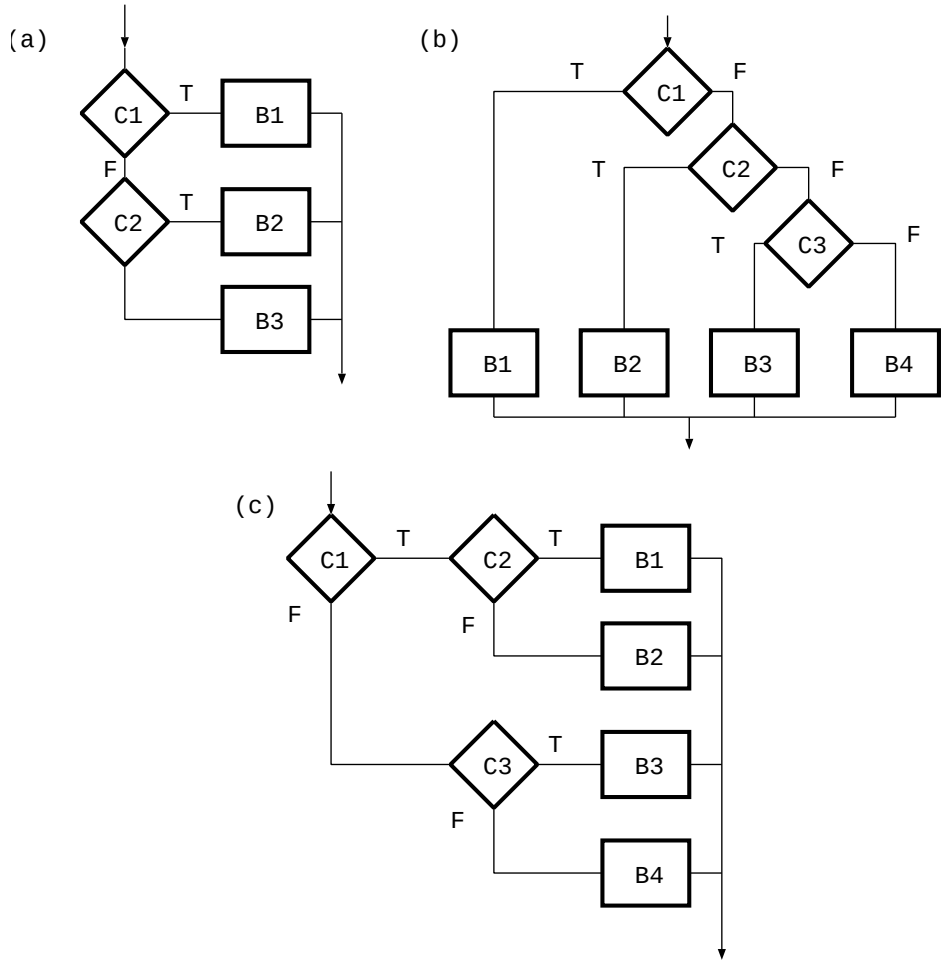


```
IF C1 THEN B1
      ELSE IF C2 THEN B2
                  ELSE IF C3 THEN B3
                              ELSE B4
                        END_IF
            END_IF
END_IF
```

(c) In the pseudocode of Chapter 9, this figure translates directly into the IF-THEN-ELSE structure

```
IF C1 THEN IF C2 THEN B1
                  ELSE B2
            END_IF
      ELSE IF C3 THEN B3
                  ELSE B4
            END_IF
END_IF
```

Figure E9.2: Figure for Exercise 9.2.



Equivalently, using the CASE structure, we would write

```
CASE
     OF C1 DO CASE
                   OF C2 DO B1
                   OF OTHERS DO B2
              END_CASE
     OF OTHERS DO CASE
                       OF C3 DO B3
                       OF OTHERS DO B4
              END_CASE
END_CASE
```

In Fig. E9.2, we redraw each of the diagrams in the statement of the exercise in the alternative form that leads more directly to a casting of the coding in the other of the two possible structures. Recognize that the two diagrams for each part of this exercise are simply topological rearrangements of each other.

9.7 Tracking Both Extremes and their Positions

Exercise: Basing your work on Algorithm (6) of Section 9.2,specifically

```
SENTINEL@ ⟵ ⟨agreed-upon special value⟩
READ (first) ITEM@ from list
EXTREME@ ⟵ ITEM@
LOOP
    READ (next) ITEM@ from list
    EXIT_LOOP WHEN ITEM@ = SENTINEL@
    IF ITEM@ and EXTREME@ are out of order
        THEN EXTREME@ ⟵ ITEM@
    END_IF
END_LOOP
WRITE "The extreme value is "; EXTREME@
```

write an algorithm that will obtain words one at a time and ultimately report (1) the word that would appear last if the list were alphabetized, (2) the word that would appear first if the list were alphabetized, (3) the total number of words given, and (4) the position of each extreme word in the original list. Only one pass through the list is permitted.

Solution: Algorithm (6) in Section 9.2 involves scanning through a list of values, keeping track of the extreme value in the list and ultimately displaying that extreme value. This exercise asks that, in scanning the list, we keep track of the earliest value, its position in the list, the latest value, and its position in the list, and that we count the entries in the list as we go so we can report the total number of values entered. For the last item, we will need to count values as they are entered, not only to have the total count at the end but also to have the information along the way that will permit us to note the position of items in the list. In broad outline, we need to execute the statements shown in Table 9.1.

While this program will do the job, note that

- the algorithm will not recognize it if the ultimate earliest or latest entry occurs more than once in the list. If, say the earliest item occurs twice, which occurrence will this algorithm report? How would you modify the algorithm to report the other occurrence?
- the algorithm does not recognize that, if a particular item *is* earlier than the current earliest item, it is certainly not later than the current latest item. Thus, the separate IF-THEN structures in the loop could be combined into a nested structure in which the algorithm didn't bother to make the second test if the first test happens to be satisfied.
- the incrementation of `CNT%` is positioned in the loop so that the sentinel is *not* counted as a valid entry. In broad terms, one must pay careful attention to the point in the sequence of instructions at which a counter is incremented; off-by-one errors are extremely common in counting operations. Generally speaking, counters should be incremented *immediately* after the task being counted has been completed and only if the completion represents a *bona fide* countable occurrence.

Table 9.1: Algorithm for Part (a) of Exercise 9.7.

```
SENTINEL$ <-- ???                    ! Set sentinel to mark end of list
CNT% <-- 0                           ! Initialize counter to count entries
READ ITEM$ from list                 ! Get first item
CNT% <-- CNT% + 1%                   ! Count item entered
EARLIEST$ <-- ITEM$                  ! First item is earliest item
LATEST$ <-- ITEM$                    ! First item is also latest item
EARLCNT% <-- CNT%                    ! Note position of current earliest item
LATCNT% <-- CNT%                     ! Note position of current latest item
LOOP
   READ ITEM$ from list              ! Get next item
   EXIT_LOOP WHEN ITEM$ = SENTINEL$
   CNT% <-- CNT% + 1%                ! Count newly entered item
   IF ITEM$ < EARLIEST$ THEN BEGIN_BLOCK            ! Adjust earliest
                               EARLIEST$ <-- ITEM$  !   if necessary
                               EARLCNT% <-- CNT%
                             END_BLOCK
   END_IF
   IF ITEM$ > LATEST$ THEN BEGIN_BLOCK              ! Adjust latest
                             LATEST$ <-- ITEM$      !   if necessary
                             LATCNT% <-- CNT%
                           END_BLOCK
   END_IF
END_LOOP
WRITE "The earliest item is"; EARLIEST$; "in position"; EARLCNT%
WRITE "The latest item is"; LATEST$; "in position"; LATCNT%
WRITE "The total number of items is"; CNT%
```

9.9 Mystery Procedure 1

Exercise: Suppose you have N% cards laid out in a row on a table. On each card is a single word. Determine the end result of applying the mystery procedure laid out in Table 9.2 to that array of cards and choose a suitable name for the procedure.

Table 9.2: Procedure for Exercise 9.9.

```
PROCEDURE ???????
SCANEND% ⟵ N%
LOOP
  CARD% ⟵ 1%
  Obtain word on card CARD% and store in WORD$
  LATEST_WORD$ ⟵ WORD$
  LATEST_CARD% ⟵ CARD%
  LOOP
    CARD% ⟵ CARD% + 1%
    Obtain word on card CARD% and store in WORD$
    IF WORD$ occurs after LATEST_WORD$
      THEN BEGIN_BLOCK
        LATEST_WORD$ ⟵ WORD$
        LATEST_CARD% ⟵ CARD%
      END_BLOCK
    END_IF
    EXIT_LOOP WHEN CARD% = SCANEND%
  END_LOOP
  Exchange card LATEST_CARD% with card SCANEND%
  SCANEND% ⟵ SCANEND% - 1%
  EXIT_LOOP WHEN SCANEND% = 1%
END_LOOP
END_PROCEDURE
```

Solution: To determine the function of the procedure in Table 9.2, we begin by looking at what happens in the outermost loop, which begins by

1. setting the counter CARD% to 1, i.e., by setting our attention on the first card on the table,
2. fetching the word on that card and storing it in memory location WORD$,
3. copying that word into a memory location labeled LATEST_WORD$, and
4. recording the card number in a memory location labeled LATEST_CARD%.

These several actions *prime* the inner loop, in each pass through which we

1. move our sights to the next card on the table,
2. copy the word on that card into WORD$,
3. adjust the value of LATEST_WORD$ and LATEST_CARD% to be the content and position of the newly examined card, but only if its contents WORD$ occurs later in the alphabet than the current contents of LATEST_WORD$.

The loop continues until we have examined the SCANEND%-th card on the table, at which point LATEST_WORD$ contains the particular word on the cards from the first through the SCANEND%-th

that occurs latest in the alphabet and `LATEST_CARD%` contains the position on the table of the card containing that word.

Upon exit from the inner loop, we exchange the `SCANEND%`-th card with the `LATEST_CARD%`-th card, thereby moving to the `SCANEND%`-th position the card in the subgroup of cards from 1 thru `SCANEND%`, i.e., we move the card in that subgroup that will be latest in the alphabet to the last position in the subgroup!

Then, we decrement `SCANEND%` and go back to the beginning of the outermost loop.

Note that the outer loop starts with `SCANEND%` set to the number of cards on the table. Thus, in the first pass through that outer loop, we move the card in the entire stack that is latest in the alphabet to the last position in the stack. Then we ignore that last card by decrementing the counter on the outer loop. At the end of the next pass through the outer loop, we move the "latest-in-the-alphabet" card in the first `N%-1%` cards to the bottom of that stack. Then, we move the "latest-in-the-alphabet" card in the first `N%-2%` cards to the bottom of that stack, ... Every time we move a card, we move the card displaced by the "latest-in-the-alphabet" card to a position earlier in the stack—so it will continue to be examined in each pass through the loops until such time as it, itself, becomes the "latest-in-the-alphabet" card, moves to the bottom, and ceases to be further examined.

By the time this mystery procedure has completed execution, the cards on the table have been arranged in alphabetic order.

Chapter 11

Solving ODEs

11.1 Driven Oscillator (MAXIMA)

Exercise: Find the motion of a driven, damped oscillator satisfying the differential equation

$$m\frac{d^2x}{dt^2} + b\frac{dx}{dt} + kx = mf\cos\omega t$$

subject to the general initial conditions

$$x(0) = x_0 \qquad ; \qquad \frac{dx}{dt}(0) = v_0$$

and then determine the particular initial values that might be imposed so that the transient part of the solution is wiped out from the beginning, i.e., so that the motion is identically the steady-state motion from the moment the oscillator is set into motion. Assume that the oscillator is *under*damped. (Note that, so that m will ultimately appear only in conjunction with b and k, we have chosen to define f as a force *per unit mass* rather than simply a force.)

Solution: To solve the given differential equation with MAXIMA, we begin by defining the differential equation with the statements

```
(%i1) depends( x, t )$
(%i2) odeqn : m*'diff( x, t, 2 ) + b*'diff( x, t, 1 ) + k*x =
          m*f*cos( \omega*t );
```

(%o2) $$m\frac{d^2x}{dt^2} + b\frac{dx}{dt} + k\,x = fm\cos(\omega t)$$

and then seek the solution with the statement

```
(%i3) soln :  ode2( odeqn, x, t )$
Is 4km - b^2 positive, negative, or zero? p;
```

where requiring that $4km - b^2 > 0$ stipulates that we seek the case when the motion is *under*damped. The complicated result—which you should look at to motivate the next steps—is suppressed but will be simplified if we make the replacements

$$k \longrightarrow \omega_0 m^2 \quad \text{and} \quad b \longrightarrow 2m\beta$$

in `soln` with the statement

```
(%i4) solna : ev( soln, k = \omega[0]^2*m, b = 2*m*\beta )$
```

and then the replacement

$$\sqrt{4\omega_0^2 - 4 * \beta^2} \longrightarrow 2\Omega$$

with the statement

```
(%i5) solnb : ev( solna, sqrt(4*\omega[0]^2 - 4*\beta^2) = 2*\Omega );
```

The result at this point is

(%o5) $$x = e^{-\beta t}\Big(\%k_1 \sin(\Omega t) + \%k_2 \cos(\Omega t)\Big) + \frac{2\beta f m^2 \omega \sin(\omega t) + (\omega_0^2 f m^2 - f m^2 \omega^2)\cos(\omega t)}{m^2\omega^4 + (4\beta^2 m^2 - 2\omega_0^2 m^2)\,\omega^2 + \omega_0^4 m^2}$$

Before attempting further simplification, we pause to note that the solution is clearly a superposition of a piece that oscillates sinusoidally with frequency ω_0, which is the natural frequency of the unforced oscillation, and a secone piece that oscillates with the frequency ω, which is the frequency of the driving force. As time unfolds, the exponential decay of the *transient* term oscillating at frequency Ω ultimately disappears, leaving only the piece oscillating at the driving frequency ω.

Various simplifications are evident from a close examination of this expression. Persuading MAXIMA to recognize those simplifications, however, is quite another matter. Note, for example, that there is a factor of m^2 in every term in both the numerator and the denominator of the term at the end of the above result and a factor of f in each term in that numerator. The single (nested) statement,

```
(%i6) solnc :  substpart( factor(part(solnb,2,2)), solnb, 2, 2);
```

(%o6) $$x = e^{-\beta t}\Big(\%k_1 \sin(\Omega t) + \%k_2 \cos(\Omega t)\Big) + \frac{f\Big(2\beta\omega \sin(\omega t) - \omega^2 \cos(\omega t) + \omega_0^2 \cos(\omega t)\Big)}{\omega^4 + 4\beta^2\omega^2 - 2\omega_0^2\omega^2 + \omega_0^4}$$

uses `part` to extract the second term in `solnb`, uses `factor` to factor both the numerator and denominator in that part (which cancels the factors of m^2 and isolates the factor f), and finally uses `substpart` to replace the original part of `solnb` with the factored form of that part.

An even cleaner form of the solution emerges if we collect the terms involving $\cos(\omega t)$ in the numerator and recognize the square of $(\omega^2 - \omega_0^2)$ in the denominator of the second term in this solution. The statements

```
(%i7) tmpa : collectterms( part( solnc, 2, 2, 1, 2), cos(\omega*t) );
```

(%o7) $2\beta\omega \sin(\omega t) + (\omega_0^2 - \omega^2)\cos(\omega t)$

```
(%i8) solnd :  substpart( tmpa, solnc,2,2,1,2 );
```

(%o8) $$x = e^{-\beta t}\Big(\%k_1 \sin(\Omega t) + \%k_2 \cos(\Omega t)\Big) + \frac{f\Big(2\beta\omega \sin(\omega t) + (\omega_0^2 - \omega^2)\cos(\omega t)\Big)}{\omega^4 + 4\beta^2\omega^2 - 2\omega_0^2\omega^2 + \omega_0^4}$$

will rearrange the numerator in the second term. The statements

```
(%i8) tmpb : part(solnd, 2, 2, 2) - 4*\beta^2*\omega^2;
```

(%o8) $\omega^4 - 2\omega_0^2\omega^2 + \omega^4$

```
(%i9) tmpc : ev( tmpb, \omega^4=a^2, \omega^2=a, \omega[0]^2=b, \omega[0]^4=b^2);
```

(%o9) $b^2 - 2ab + a^2$

```
(%i10) tmpd : factor(tmpc);
```

(%o10) $(b-a)^2$

```
(%i11) tmpe : ev( tmpd, b=\omega[0]^2, a= \omega^2 ) + 4*\beta^2*\omega^2;
```

(%o11) $\left(\omega_0^2-\omega^2\right)^2+4\beta^2\omega^2$

```
(%i12) solne :  substpart( tmpe, solnd,2,2,2);
```

(%o12) $$x=e^{-\beta t}\Big(\%k_1\sin(\Omega t)+\%k_2\cos(\Omega t)\Big)+\frac{f\Big(2\beta\omega\sin(\omega t)+(\omega_0^2-\omega^2)\cos(\omega t)\Big)}{\left(\omega_0^2-\omega^2\right)^2+4\beta^2\omega^2}$$

At last, we have beaten the solution into a suitable form, from which we readily see in particular that the response to the driving force resonates when $\omega=\omega_0$,

We can now move to imposing general initial conditions $x(0)=x_0$, $v(0)=v_0$ with the statements

```
(%i13) solnf : ic2( solne, t=0, x=x[0], 'diff(x,t)=v[0] );
```

(%o13) $$x=e^{-\beta t}\big(A\sin(\Omega t)+B\cos(\Omega t)\big)+\frac{f\Big(2\beta\omega\sin(\omega t)+(\omega_0^2-\omega^2)\cos(\omega t)\Big)}{\left(\omega_0^2-\omega^2\right)^2+4\beta^2\omega^2}$$

We have suppressed explicit inclusion of the values MAXIMA gives for the coefficients A and B. As we have already concluded, the term multiplied by the decaying exponential will disappear as t becomes large, leaving only the steady state part that oscillates with the driving frequency and resonates when the driving frequency is close to the natural frequency.

To be sure, MAXIMA indeed returns explicit values for the constants A and B. For the sake of completeness, we use MAXIMA as follows to extract and simplify these four coefficients. For the coefficient A, we would execute the statement[1]

```
(%i14) A : part( solnf,2,1,2,1) / sin(\Omega*t);
```

(%o14) $$\frac{\left(\begin{array}{c}\big(x_0\beta+v_0\big)\omega^4+\big(-\beta f+4x_0\beta^3+4v_0\beta^2-2\omega_0^2x_0\beta-2\omega_0^2v_0\big)\omega^2\\ -\omega_0^2\beta f+\omega_0^4x_0\beta+\omega_0^4v_0\end{array}\right)}{\omega^2\big(4\beta^2\Omega-2\omega_0^2\Omega\big)+\omega^4\Omega+\omega_0^4\Omega}$$

```
(%i15) B : part( solnf,2,1,2,2) / cos(\Omega*t);
```

(%o15) $$\frac{x_0\omega^4+\big(f+4x_0\beta^2-2\omega_0^2x_0\big)\omega^2-\omega_0^2f+\omega_0^4x_0}{\omega^4+\big(4\beta^2-2\omega_0^2\big)\omega^2+\omega_0^4}$$

where we find the first and second terms in the parentheses following the exponential function in the solution with the `part` function and then divide out the part of those terms that are not part of the coefficient.

To achieve the objective of the problem, we want to find initial conditions that will result in the transient terms having zero coefficients so that, in fact, the transient term is gone *from the beginning*. Evidently, because the sine and cosine functions are linearly independent of one another, we cannot "wipe out" the transient term as required unless the coefficients of the sine and cosine terms are separately zero. Thus, to eliminate the transient term from the beginning, we must require that

$$A=0 \qquad \text{and} \qquad B=0$$

[1] *Warning*: MAXIMA does not in all circumstances present the terms in this solution in the same order. Thus, the numbers needed to extract the coefficients of the terms with sine or cosine factors may be different in your session than they are here. *Beware*!

Our remaining task in MAXIMA then is simply to solve these two equations for the initial values, x_0 and v_0. Each of the coefficients, however, is a fraction whose denominator is *not* zero and involves neither x_0 nor v_0. Thus, the required coefficients will be zero only if the numerators are zero, and we find that the equations determining x_0 and v_0 are

```
(%i16) eq1 : num(A) = 0$
(%i17) eq2 : num(B) = 0$
```

Finally, we determine the desired initial conditions with the statements

```
(%i18) notran : solve( [eq1, eq2], [ x[0], v[0] ] );
```

$$(\%o18) \qquad \left[\left[x_0 = -\frac{f\omega^2 - \omega_0^2 f}{\omega^4 + (4\beta^2 - 2\omega_0^2)\omega^2 + \omega_0^4}, v_0 = \frac{2\beta f\omega^2}{\omega^4 + (4\beta^2 - 2\omega_0^2)\omega^2 + \omega_0^4} \right] \right]$$

Note that the factor f appears in both terms in the numerator of x_0 and that the two denominators are the same (and are the same as the expression in the denominator of the expression at `(%o8)`. We can recast the numerator of x_0 with the statements

```
(%i19) tmpf : collectterms( part(notran, 1, 1, 2, 1, 1), f );
```

$$(\%o19) \qquad f(\omega^2 - \omega_0^2)$$

```
(%i20) notrana: substpart( tmpa, notran, 1, 1, 2, 1, 1 );
```

$$(\%o20) \qquad \left[\left[x_0 = -\frac{f(\omega^2 - \omega_0^2)}{\omega^4 + (4\beta^2 - 2\omega_0^2)\omega^2 + \omega_0^4}, v_0 = \frac{2\beta f\omega^2}{\omega^4 + (4\beta^2 - 2\omega_0^2)\omega^2 + \omega_0^4} \right] \right]$$

Then, we simplify the denominators with the substitutions[2]

```
(%i21) substpart( tmpe, notrana, 1,1,2,1,2)$
(%i22) notranb : substpart( tmpe, %, 1,2,2,2)$
```

$$(\%o22) \qquad \left[\left[x_0 = -\frac{f(\omega^2 - \omega_0^2)}{(\omega_0^2 - \omega^2)^2 + 4\beta^2\omega^2}, v_0 = \frac{2\beta f\omega^2}{(\omega_0^2 - \omega^2)^2 + 4\beta^2\omega^2} \right] \right]$$

Finally, to verify that transients disappear with these initial values, we substitute these values into the original solution with the statement

```
(%i23) solng : ev( solnf, notranb );
```

$$(\%o23) \qquad x = e^{-\beta t}\left(\omega^2 \frac{L\sin(\omega t)}{\omega^2(4\beta^2\Omega - 2\omega_0^2\Omega) + \omega^4\Omega + \omega_0^4\Omega} + \frac{M\cos(\Omega t)}{\omega^4 + (4\beta^2 - 2\omega_0^2)\omega^2 + \omega_0^4}\right) + \frac{f\left(2\beta\omega\sin(\omega t) + (\omega_0^2 - \omega^2)\cos(\omega t)\right)}{\left(\omega_0^2 - \omega^2\right)^2 + 4\beta^2\omega^2}$$

The quantities L and M are both given explicitly on the screen but are too complicated to display here. The structure here, however, reveals that

```
(%i24) L : part(solng, 2,1,2,1,1,1 )$
(%i25) M : part(solna, 2,1,2,2,1,1 )$
```

The values of L and M are still quite involved. Exploration with several functions (`expand`, `combine`, `multthru`, `factor`) that might simplify these expressions ultimately leads to the results that

[2] Note that the minus sign preceding the solution for x_0 counts as a part.

```
(%i26) factor( L );
(%o26)     0
(%i27) factor( M );
(%o27)     0
```

and we confirm that, with the identified initial conditions, terms with L and M in `(%o19)` disappear, the transient part of the general solution is gone from the beginning, and the steady state solution describes the motion right from time $t = 0$.

Note also that, if we set the frequency ω of the driving force equal to the natural frequency ω_0 of the undamped oscillator, Unfortunately, the simple statement

```
(%i28) ev( notranb, \omega = \omega[0] );
```

replaces ω with ω_0 but also replaces the characters `\omega` in `\omega[0]` which then becomes ω_{00}, which, not surprisingly, MAXIMA views to be different from ω_0. Instead, we achieve the objective in two steps, specifically

```
(%i29) notranc : subst( b, \omega[0], notranb )$
(%i30) ev( notranc, \omega = b );
```

`(%o30)` $$\left[\left[x_0 = 0, v_0 = \frac{f}{2\beta}\right]\right]$$

While this solution is perfectly appropriate if we carefully match the driving frequency to the natural frequency, the necessary initial conditions to wipe out the transient terms from the beginning are more complicated if the driving frequency differs from the natural frequency.

11.2 Two Masses, One Spring (MAXIMA)

Exercise: Take away the walls and the two springs connecting the blocks to the walls in the system of Fig. E11.1, let the masses be different (m_1 and m_2, say) and denote the constant of the one spring by k. Suppose the blocks are constrained to move along a straight line. Measuring from an arbitrarily selected origin on that line, let the coordinates of the particles be x_1 and x_2, respectively. The equations of motion for this system are

$$m_1 \frac{d^2x_1}{dt^2} = k(x_2 - x_1) \qquad ; \qquad m_2 \frac{d^2x_2}{dt^2} = -k(x_2 - x_1)$$

Let the system be put into motion with arbitrary initial conditions

$$x_1(0) = x_{10} \quad ; \quad x_2(0) = x_{20} \quad ; \quad \frac{dx_1(0)}{dt} = v_{10} \quad ; \quad \frac{dx_2(0)}{dt} = v_{20}$$

Solve this initial-value problem for $x_1(t)$ and $x_2(t)$ and then examine the behavior of the particular quantities

$$X(t) = \frac{m_1 x_1(t) + m_2 x_2(t)}{m_1 + m_2} \qquad \text{and} \qquad Y(t) = x_2(t) - x_1(t)$$

which are, respectively, the position of the center of mass of the system and the position of the second block relative to the first block.

Solution: Let us start by defining the differential equations and initial conditions for the system of interest with the statements[3]

```
(%i1) eqn1 : m1*'diff( x1(t), t, 2 ) = k*(x2(t) - x1(t));
```

(%o1) $$m_1 \frac{d^2x_1(t)}{dt^2} = k\left(x_2(t) - x_1(t)\right)$$

```
(%i2) eqn2 : m2*'diff( x2(t), t, 2 ) = -k*(x2(t) - x1(t));
```

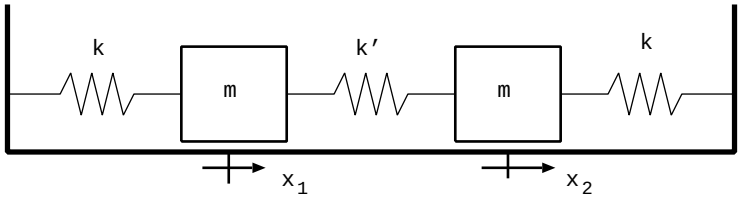
(%o2) $$m_2 \frac{d^2x_2(t)}{dt^2} = -k\left(x_2(t) - x_1(t)\right)$$

```
(%i3) atvalue( x1(t), t=0, x10 )$
(%i4) atvalue( x2(t), t=0, x20 )$
(%i5) atvalue( 'diff( x1(t), t ), t=0, v10 )$
(%i6) atvalue( 'diff( x2(t), t  ), t=0, v20 )$
```

Then, we find the solution with the statement

[3] Remember that dependence on the independent variable must be explicitly indicated when `desolve` is to be invoked.

Figure E11.1: A system of two coupled objects.



```
(%i7) soln : desolve( [eqn1, eqn2], [x1(t), x2(t)]);
```

Is $km_1m_2(m_2+m_1)$ *positive, negative, or zero?* `p;`

(%o7) $\left[\, x_1(t) = \ldots, x_2(t) = \ldots \,\right]$

where, of course, the ellipses stand for the actual (elaborate and extensive) expressions that MAXIMA supplies. The resulting expressions, however, include some terms that vary sinusoidally with the frequency

$$\Omega = \frac{\sqrt{km_1m_2(m_2+m_1)}}{m_1m_2} = \sqrt{k\frac{m_1+m_2}{m_1m_2}} = \sqrt{k\left(\frac{1}{m_1}+\frac{1}{m_2}\right)} = \sqrt{\frac{k}{m_{\rm red}}}$$

where $m_{\rm red}$, defined by

$$\frac{1}{m_{\rm red}} = \frac{1}{m_1} + \frac{1}{m_2} \quad \text{or} \quad m_{\rm red} = \frac{m_1m_2}{m_1+m_2}$$

is called the reduced mass of the two-body system.

With this recognition, we can simplify the appearance of the solution just a little by substituting Ω for $\sqrt{km_1m_2(m_2+m_1)}/(m_1m_2)$ with the statement

```
(%i8) solna : subst( m1*m2*\Omega, sqrt(k*m1*m2*(m2+m1)), soln );
```

(%o8) $\left[\, x_1(t) = \ldots, x_2(t) = \ldots \,\right]$

where the ellipses still stand for complicated expressions, though they are somewhat less complicated than at `(%o7)`. In an effort to simplify the expressions further, let us extract $x_1(t)$ and $x_2(t)$ and manipulate them a bit with the statements

```
(%i9) solnx1 : combine( expand( part( solna, 1, 2 ) ) );
```

(%o9)
$$\frac{-m_2\cos(\Omega t)\,x_{20} + m_2\,x_{20} + m_2\cos(\Omega t)\,x_{10} + m_1\,x_{10} + m_2\,t\,v_{20} + m_1\,t\,v_{10}}{m_2+m_1} + \frac{m_2\sin(\Omega t)\,v_{10} - m_2\sin(\Omega t)\,v_{20}}{m_2\Omega + m_1\Omega}$$

```
(%i10) solnx2 : combine( expand( part( solna, 2, 2 ) ) );
```

(%o10)
$$\frac{m_1\cos(\Omega t)\,x_{20} + m_2\,x_{20} - m_1\cos(\Omega t)\,x_{10} + m_1\,x_{10} + m_2\,t\,v_{20} + m_1\,t\,v_{10}}{m_2+m_1} + \frac{m_1\sin(\Omega t)\,v_{20} - m_1\sin(\Omega t)\,v_{10}}{m_2\Omega + m_1\Omega}$$

After some exploration with a variety of MAXIMA commands, we discover that further simplification occurs with the statements

```
(%i11) solnx1a : collectterms( expand(solnx1), t );
```

(%o11)
$$\cos(\Omega t)\left(\frac{m_2\,x_{10}}{m_2+m_1} - \frac{m_2\,x_{20}}{m_2+m_1}\right) + \frac{m_2\,x_{20}}{m_2+m_1} + \frac{m_1\,x_{10}}{m_2+m_1} + \sin(\Omega t)\left(\frac{m_2\,v_{10}}{m_2\Omega+m_1\Omega} - \frac{m_2\,v_{20}}{m_2\Omega+m_1\Omega}\right) + t\left(\frac{m_2\,v_{20}}{m_2+m_1} + \frac{m_1\,v_{10}}{m_2+m_1}\right)$$

```
(%i12) solnx2a : collectterms( expand(solnx2), t );
```

(%o12)
$$\cos(\Omega t)\left(\frac{m_1\,x_{20}}{m_2+m_1} - \frac{m_1\,x_{10}}{m_2+m_1}\right) + \frac{m_2\,x_{20}}{m_2+m_1} + \frac{m_1\,x_{10}}{m_2+m_1}$$

$$+\sin(\Omega t)\left(\frac{m_1\,v_{20}}{m_2\Omega+m_1\Omega}-\frac{m_1\,v_{10}}{m_2\Omega+m_1\Omega}\right)+t\left(\frac{m_2\,v_{20}}{m_2+m_1}+\frac{m_1\,v_{10}}{m_2+m_1}\right)$$

Finally, it would be grand if we could combine the two terms in each set of large parentheses over their common denominator. Again after much trial and error, we find that the statements

```
(%i13) combine( solnx1a )$
(%i14) solnx1b :  map( factor, % );
```

(%o14)
$$\frac{m_2\,x_{20}+m_1\,x_{10}}{m_2+m_1}-\frac{m_2\cos(\Omega t)\,(x_{20}-x_{10})}{m_2+m_1}$$
$$+\frac{t\,(m_2\,v_{20}+m_1\,v_{10})}{m_2+m_1}-\frac{m_2\sin(\Omega t)\,(v_{20}-v_{10})}{(m_2+m_1)\Omega}$$

```
(%i15) combine( solnx2a )$
(%i16) solnx2b :  map( factor, % );
```

(%o16)
$$\frac{m_2\,x_{20}+m_1\,x_{10}}{m_2+m_1}+\frac{m_1\cos(\Omega t)\,(x_{20}-x_{10})}{m_2+m_1}$$
$$+\frac{t\,(m_2\,v_{20}+m_1\,v_{10})}{m_2+m_1}+\frac{m_1\sin(\Omega t)\,(v_{20}-v_{10})}{(m_2+m_1)\Omega}$$

achieve that objective.

Now, we are ready to look at the center of mass of the system, $X(t)$, as well as the position of the second block relative to the first, $Y(t)$. To find the center of mass, we invoke the statement

```
(%i17) X : collectterms( expand(factor((m1*solnx1b + m2*solnx2b)/(m1 + m2))), t );
```

(%o17)
$$\frac{m_2\,x_{20}}{m_2+m_1}+\frac{m_1\,x_{10}}{m_2+m_1}+t\left(\frac{(m_2\,t\,v_{20}}{m_2+m_1}+\frac{m_1\,t\,v_{10}}{m_2+m_1}\right)$$

This interesting result has the simple form $X = X_0 + V_{\rm cm}t$, where

$$X_0=\frac{m_1x_{10}+m_2x_{20}}{m_1+m_2}\qquad;\qquad V_{\rm cm}=\frac{m_1v_{10}+m_2v_{20}}{m_1+m_2}$$

In other words, the center of mass of the moving system starts at the initial position X_0 of the center of mass and moves along a straight line at *constant velocity* $V_{\rm cm}$. This result, of course, confirms our expectation that, in the absence of external forces, the linear momentum of the composite system must be conserved. However, as the spring stretches and contracts, the center of mass will motor along at a constant velocity. Note also that the velocity of the center of mass is independent of the spring constant.

To find the motion of the second object relative to the first, we would use instead the MAXIMA statement

```
(%i18) V : collectterms( expand(factor(solnx2b - solnx1b)), t );
```

(%o18)
$$\cos(\Omega t)\,(x_{20}-x_{10})+\sin(\Omega t)\left(\frac{v_{20}}{\Omega}-\frac{v_{10}}{\Omega}\right)$$

Since the coefficients of the sine and cosine terms are *constants*, this expression for $Y(t)$ can be generalized as $A\cos(\Omega t) + B\sin(\Omega t)$. We conclude, then that the motion of the second object relative to the first is simple harmonic and is given by an oscillating function that could, if we chose, be written as a single sinusoidal function with an appropriate amplitude and phase. The main point is that the separation of the two objects oscillates quite simply as the two objects individually move closer to and further away from the equilibrium position (so long, of course, as we remain in

our frictionless dreamworld). Note that, in terms of the reduced mass m_{red}, the frequency of the oscillation has the simple relationship

$$\Omega = \sqrt{\frac{k}{m_{\text{red}}}}$$

to the parameters describing the system. Indeed, the relative motion is identical to that of a *single* object of mass m_{red} on a spring of constant k.

Taken together, then, we have a system of two blocks moving across a surface, with the center of mass moving at constant velocity, and, as the motion ensues, with m_2 pulling m_1, then m_1 catching up and pushing block m_2, then .... This pushing and pulling process is repeated indefinitely. The behavior, of course, follows directly from the forces and Newton's laws of motion.

11.4 Compound Pendulum (MAXIMA)

Exercise: The system called the double pendulum shown in Fig. E11.2 consists of a ball of mass m_1 hanging from a rigid and massless rod of length l_1 attached to the ceiling and a second ball of mass m_2 hanging from a rigid and massless rod of length l_2 attached to the first ball. The balls swing in a plane, and the configuration of the system is specified by giving two angles, the first of which, θ, gives the angle that the upper string makes with the vertical and the second of which, ϕ, gives the angle that the lower string makes with the vertical. The motion *can* be very complicated and at times will be chaotic. For small amplitudes, however, things are much more sedate. When the amplitudes of the motion of both balls are small and—to simplify a little bit—when the strings are both the same length ($l_1 = l_2$, which we will symbolize with the letter l), the equations of motion turn out to be

$$\frac{d^2\theta}{dt^2} + \frac{m_2}{m_1+m_2}\frac{d^2\phi}{dt^2} + \frac{g}{l}\theta = 0$$

$$\frac{d^2\phi}{dt^2} + \frac{d^2\theta}{dt^2} + \frac{g}{l}\phi = 0$$

Find the normal modes of oscillation of this system and determine the initial conditions that will cause the system to oscillate exclusively in one or the other of these modes.

Solution: We begin by introducing the symbol β to stand for the ratio $m_2/(m_1+m_2)$ and the symbol Ω to stand for the quantity $\sqrt{g/l}$, finding that the equations we seek to solve are now

$$\frac{d^2\theta}{dt^2} + \beta\frac{d^2\phi}{dt^2} + \Omega^2\theta = 0 \quad ; \quad \frac{d^2\phi}{dt^2} + \frac{d^2\theta}{dt^2} + \Omega^2\phi = 0$$

which we enter into a MAXIMA notebook with the statements

```
(%i1) depends( [\theta, \phi], t )$
(%i2) deq1 : 'diff(\theta, t, 2) + \beta*'diff(\phi,t,2) +\Omega^2*\theta=0$
(%i3) deq2 : 'diff(\phi, t, 2) + 'diff(\theta,t,2) +\Omega^2*\phi=0$
```

Next we suppose solutions of the form

$$\theta(t) = A\cos(\omega t) \qquad ; \qquad \phi(t) = B\cos(\omega t)$$

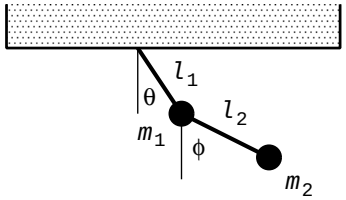
and substitute them into `deq1` and `deq2`. We define ϕ and θ as expressions depending on t with the statements

```
(%i4) \theta : A*cos(\omega*t)$
(%i5) \phi : B*cos(\omega*t)$
```

Then, we effect the desired substitution into `deq1` and `deq2` with the statements

---

Figure E11.2: Figure for Exercise 11.4.



```
(%i6) eq1 : ev( deq1, diff );
```

(%o6) $-\beta\omega^2\cos(\omega t)\,B-\Omega^2\cos(\omega t)\,A+-\omega^2\cos(\omega t)=0$

```
(%i7) eq2 : ev( deq2, diff );
```

(%o7) $\Omega^2\cos(\omega t)\,B-\omega^2\cos(\omega t)\,B-\omega^2\cos(\omega t)\,A=0$

Then, dividing out the factor $\cos(\omega t)$ with the statements

```
(%i8) eq3 : expand( eq1/cos(\omega*t) );
```

(%o8) $-\beta\omega^2\,B+\Omega^2\,A-\omega^2\,A=0$

```
(%i9) eq4 : expand( eq2/cos(\omega*t) );
```

(%o9) $\Omega^2\,B-\omega^2\,B-\omega^2\,A=0$

we arrive at two algebraic equations (`eq3` and `eq4`), for which we seek conditions under which non-trivial solutions for A and B can be found. Writing these two equations in the form

$$\begin{pmatrix} -\omega^2+\Omega^2 & -\beta\omega^2 \\ -\omega^2 & -\omega^2+\Omega^2 \end{pmatrix}\begin{pmatrix} A \\ B \end{pmatrix}=\begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

we see that, in essence, we want to extract the coefficient matrix with respect to the constants (A, B) and then find the values of ω^2 which will cause the determinant of that matrix to be zero. The statements

```
(%i10) mat : coefmatrix( [Eq3, eq4], [A, B] );
```

(%o10) $$\begin{bmatrix} \Omega^2-\omega^2 & -\beta\omega^2 \\ -\omega^2 & \Omega^2-\omega^2 \end{bmatrix}$$

will extract the coefficient matrix. Then, the statement

```
(%i11)  detm : expand( determinant(mat) );
```

(%o11) $\Omega^4-2\omega^2\Omega^2-\beta\omega^4+\omega^4$

finds the determinant, the statement

```
(%i12) det1 : ev( detm, \omega = sqrt(alpha) );
```

(%o12) $\Omega^4-2\alpha\Omega^2-\beta\alpha^2+\alpha^2$

temporarily replaces ω^2 with α so MAXIMA will recognize the expression as a quadratic equation (in α). The statement

```
(%i13) soln : factor( solve( det1 = 0, \alpha ) );
```

(%o13) $$\left[\alpha=-\frac{(\sqrt{\beta}+1)\,\Omega^2}{\beta-1},\alpha=\frac{(\sqrt{\beta}-1)\,\Omega^2}{\beta-1}\right]$$

generates the two solutions for α. An even simpler form results if we note that $\beta-1=(\sqrt{\beta}+1)(\sqrt{\beta}-1)$ and introduce the substitution

```
(%i14) solna : ev( soln, (\beta - 1) = (\sqrt(\beta)+1)*(\sqrt(\beta) - 1) );
```

(%o14) $$\left[\alpha=-\frac{\Omega^2}{\sqrt{\beta}-1},\alpha=\frac{\Omega^2}{\sqrt{\beta}+1}\right]$$

and—at last—-the statements

```
(%i15) assume( \Omega > 0, \beta < 1 );
```

(%o15) $[\ \Omega > 0, \beta < 1\ ]$

```
(%i16) w1 : sqrt( part(solna, 1, 2 ));
```

(%o16) $$\frac{\Omega}{\sqrt{1-\sqrt{\beta}}}$$

```
(\%i17) w2 : sqrt( part(solna, 2, 2 ));
```

(%o17) $$\frac{\Omega}{\sqrt{\sqrt{\beta}+1}}$$

yield the two separate solutions for ω. (We don't bother with the negative square roots, since the sign of ω is irrelevant to the solution. Further, we note that $0 \leq \beta \leq 1$ and the quantities under the square roots are therefore always positive; the solutions indeed are oscillatory, not exponentially growing or decaying.)

We must, however, also find the initial amplitudes that will start the system oscillating in the normal mode corresponding to each of these frequencies. To do so, we return to the algebraic equations `eq3` and `eq4`. We set ω in `eq3` to `w1` and solve for B—denote it B_1—as a multiple of A with the statements

```
(%i18) ev( eq3, \omega = w1 )$
(%i19) B1 : solve( %, B );
```

(%o19) $$\left[B = -\frac{A}{\sqrt{\beta}}\right]$$

Then, we verify that `eq4` is satisfied as well with the statement

```
(%i20) ratsimp( ev( eq4, %, \omega = w1 ) );
```

(%o20) $0 = 0$

Parallel statements applied to the second frequency yield the results

```
(%i21) ev( eq3, \omega = w2 )$
(%i22) solve( %, B );
```

(%o22) $$\left[B = \frac{A}{\sqrt{\beta}}\right]$$

```
(%i23) ratsimp( ev( eq4, %, \omega = w2 ) );
```

(%o23) $0 = 0$

Intriguingly, one normal mode is obtained by displacing the upper pendulum by some amount A in one direction and the lower pendulum by an amount $A/\sqrt{\beta}$—which is *larger* than A because $\beta < 1$—in the *same* direction, while the other normal mode is obtained by displacing the upper pendulum by some amount A in one direction and the lower pendulum by an amount $A/\sqrt{\beta}$ in the *other* direction. The interesting feature is that the displacement of the second pendulum in the two cases has the same *magnitude* even though the direction is different. In both cases the pendula are released from rest.

We should note also that the mode in which the two pendula are displaced in the same direction will oscillate with the frequency

$$\omega_2 = \frac{\Omega}{\sqrt{1+\sqrt{\beta}}} = \frac{\sqrt{g/l}}{\sqrt{1+\sqrt{\beta}}}$$

while the mode in which the two pendula are displaced in opposite directions will oscillate with the frequency

$$\omega_1 = \frac{\Omega}{\sqrt{1-\sqrt{\beta}}} = \frac{\sqrt{g/l}}{\sqrt{1-\sqrt{\beta}}}$$

From these relationships, we note that $\omega_2 < \omega_1$. As we would expect (and as has turned out to be the case, the mode in which the pendula oscillate *in phase* would be the lower frequency mode and the mode in which the pendula oscillate *out of phase* would be the higher frequency mode. Note also that the unit $\sqrt{g/l}$ in which these frequencies are expressed is the frequency of small amplitude oscillations of a single pendulum of length l.

11.6 Series Solution (MAXIMA)

Exercise: Among the simplest of differential equations is the equation

$$\frac{d^2x}{dt^2} + \omega^2 x = 0$$

that describes a simple harmonic oscillator. Generate a series solution to this equation and then verify that the solution thus generated agrees with the known solution

$$x(t) = A\cos\omega t + B\sin\omega t$$

where A and B are constants determined by the initial conditions.

Solution: Solving differential equations using series in MAXIMA is easy if one knows the right statements. We begin by assigning the left-hand side of the differential equation

$$\frac{d^2x}{dt^2} + \omega^2 x = 0$$

to the variable `eq1` with the statement

```
(%i1) exp1 : 'diff( x, t, 2 ) + \omega^2*x;
```

(%o1) $$\frac{d^2x}{dt^2} + \omega^2\, x$$

Now, supposing a solution of the form

$$x = \sum_i^{\infty} a_i t^{i+\beta}$$

we create a (truncated) series with the statement

```
(%i2) soln :  sum( a[i]*t^(i+\beta), i, 0, 10 )$
```

Then, we substitute this expression into the differential equation with the statement

```
(%i3) exp2 : ev( exp1, x=soln, diff );
```

(%o3) (result too long to record)

Each term in this result, however, has a factor of t^β, which we can remove by division. To simplify the expression, we also multiply by t^2, which adjusts every power of t upward by 2 and creates an expression in which the lowest power of t is t^0. (We can perform these adjustments because the result is set equal to zero. Since t is not identically zero, we can multiply the overall expression by t to any power without destroying the "zero-ness" of the result.) Thus, we we invoke the statements

```
(%i4) soln1 : expand( t^2*exp2/t^\beta )$
(%i5) soln2 : collectterms( soln1, t );
```

(%o5) $$a_{10}\omega^2 t^{12} + a_9\omega^2 t^{11} + \left(a_8\omega^2 + a_{10}\beta^2 + 19a_{10}\beta + 90a_{10}\right)t^{10}$$
$$+\left(a_6\omega^2 + a_9\beta^2 + 17a_9\beta + 72a_9\right)t^9 + \ldots + \left(a_1\beta^2 + a_1\beta\right)t + a_0\beta^2 - a_0\beta$$

Following the pattern described in the text for the method of Frobenius, we now extract the coefficients with the statement

```
(%i6) for i from 0 thru 12 do c[i] : coeff( soln2, t, i )$
(%i7) for i from 0 thru 12 do print("i = ", i, "c[i] is    ", c[i] )$
```

$i = 0$ c[i] is $a_0\beta^2 - a_0\beta$
$i = 1$ c[i] is $a_1\beta^2 - a_1\beta$
$i = 2$ c[i] is $a_0\omega^2 + a_2\beta^2 + 3a_2\beta + 2a_2$
$i = 3$ c[i] is $a_1\omega^2 + a_3\beta^2 + 5a_3\beta + 6a_3$
$i = 4$ c[i] is $a_2\omega^2 + a_4\beta^2 + 7a_4\beta + 12a_4$
$i = 5$ c[i] is $a_3\omega^2 + a_5\beta^2 + 9a_5\beta + 20a_5$
$i = 6$ c[i] is $a_4\omega^2 + a_6\beta^2 + 11a_6\beta + 30a_6$
$i = 7$ c[i] is $a_5\omega^2 + a_7\beta^2 + 13a_7\beta + 42a_7$
$i = 8$ c[i] is $a_6\omega^2 + a_8\beta^2 + 15a_8\beta + 56a_8$
$i = 9$ c[i] is $a_7\omega^2 + a_9\beta^2 + 17a_9\beta + 72a_9$
$i = 10$ c[i] is $a_8\omega^2 + a_{10}\beta^2 + 19a_{10}\beta + 90a_{10}$
$i = 11$ c[i] is $a_9\omega^2$
$i = 12$ c[i] is $a_{10}\omega^2$

Each coefficient in this list must itself be zero. To find the possible values of β, we require the *first* coefficient to be zero, discovering with the statement

```
(%i8) bet : solve( c[0]=0, \beta );
(%o8)      [ β = 0, β = 1 ]
```

that β can be either 0 or 1. (We assume that $a_0 \neq 0$.)

Suppose that we take $\beta = 1$ first. To find the coefficients in this case, we would execute the statements[4]

```
(%i9) for i from 0 thru 10 do c1[i] : ev( c[i], \beta = 1 );
(%i10) for i from 0 thru 10 do print("i = ", i, "c1[i] is    ", c1[i] )$
```

$i = 0$ c1[i] is 0
$i = 1$ c1[i] is $2a_1$
$i = 2$ c1[i] is $a_0\omega^2 + 6a_2$
$i = 3$ c1[i] is $a_1\omega^2 + 12a_3$
$i = 4$ c1[i] is $a_2\omega^2 + 20a_4$
$i = 5$ c1[i] is $a_3\omega^2 + 30a_5$
$i = 6$ c1[i] is $a_4\omega^2 + 42a_6$
$i = 7$ c1[i] is $a_5\omega^2 + 56a_7$
$i = 8$ c1[i] is $a_6\omega^2 + 72a_8$
$i = 9$ c1[i] is $a_7\omega^2 + 90a_9$
$i = 10$ c1[i] is $a_8\omega^2 + 110a_{10}$

```
(%i11) for i from 1 thru 10 do s[i] : solve( c1[i]=0, a[i] );
{%i12) s[0] : [ a[0] = a[0] ]$
(%i13)  for i from 0 thru 10 do print( s[i] );
```

[4] We stop at $i = 10$ because the truncation of the original series at $n = 10$ renders the values of c_i for $i = 11, 12$ are incomplete.

$$\left[\, a_0 = a_0 \,\right]$$
$$\left[\, a_1 = 0 \,\right]$$
$$\left[\, a_2 = -\frac{a_0\omega^2}{6} \,\right]$$
$$\left[\, a_3 = -\frac{a_1\omega^2}{12} \,\right]$$
$$\left[\, a_4 = -\frac{a_2\omega^2}{20} \,\right]$$
$$\left[\, a_5 = -\frac{a_3\omega^2}{30} \,\right]$$
$$\left[\, a_6 = -\frac{a_4\omega^2}{42} \,\right]$$
$$\left[\, a_7 = -\frac{a_5\omega^2}{56} \,\right]$$
$$\left[\, a_8 = -\frac{a_6\omega^2}{72} \,\right]$$
$$\left[\, a_9 = -\frac{a_7\omega^2}{90} \,\right]$$
$$\left[\, a_{10} = -\frac{a_8\omega^2}{110} \,\right]$$

Next, we need to recast all of these coefficients so that they depend only on a_0 with the statements

```
(%i14) s1[0] : s[0]$ s1[1] : s[1]$ s1[2] : s[2] $
(%i15) for i from 3 thru 10 do s1[i] : ev( s[i], s1[i-2] );
(%i16) for i from 0 thru 10 do print(s1[i] );
```

$$\left[\, a_0 = a_0 \,\right]$$
$$\left[\, a_1 = 0 \,\right]$$
$$\left[\, a_2 = -\frac{a_0\omega^2}{6} \,\right]$$
$$\left[\, a_3 = 0 \right.$$
$$\left[\, a_4 = \frac{a_0\omega^4}{120} \,\right]$$
$$\left[\, a_5 = 0 \right.$$
$$\left[\, a_6 = -\frac{a_0\omega^6}{5040} \,\right]$$
$$\left[\, a_7 = 0 \right.$$
$$\left[\, a_8 = \frac{a_0\omega^8}{362880} \,\right]$$
$$\left[\, a_9 = 0 \right.$$
$$\left[\, a_{10} = -\frac{a_0\omega^{10}}{39916800} \,\right]$$

Finally, to obtain the series for the solution corresponding to $\beta = 1$, we substitute these values for the coefficients into the original series with the statements

```
(%i17) vals : makelist( part(s1[i],1), i, 0, 10 );
```

(%o17) $$\left[a_0 = a_0, a_1 = 0, a_2 = -\frac{a_0\omega^2}{6}, \ldots,\ a_9 = 0, a_{10} = -\frac{a_0\omega^{10}}{39916800}\ \right]$$

```
(%i18) ev( soln, vals, \beta=1 );
```

(%o18) $$-\frac{a_0\omega^{10}t^{11}}{39916800} + \frac{a_0\omega^8 t^9}{3672880} - \frac{a_0\omega^6 t^7}{5040} + \frac{a_0\omega^4 t^5}{120} - \frac{a_0\omega^2 t^3}{6} + a_0 t$$

Note that there is a factor of a_0 in each term, which we should extract from the expression. Further, the series will be more recognizable if we multiply and divide by ω. The statement

```
(%i19) (a[0]/\omega)*multthru(\omega/a[0], % );
```

(%o19) $$\frac{a_0}{\omega}\left(-\frac{\omega^{11}t^{11}}{39916800} + \frac{\omega^9 t^9}{3672880} - \frac{\omega^7 t^7}{5040} + \frac{\omega^5 t^5}{120} - \frac{\omega^3 t^3}{6} + \omega t\right)$$

achieves those objectives. If we now recognize that $3! = 6$, $5! = 120$, $7! = 5040$, $9! = 362880$, and $11! = 39916800$, we can recast this result as the series

$$\frac{a_0}{\omega}\left(\omega t - \frac{t^3\omega^3}{3!} + \frac{t^5\omega^5}{5!} - \frac{t^7\omega^7}{7!} + \frac{t^9\omega^9}{9!} - \frac{t^{11}\omega^{11}}{11!}\right)$$

which we recognize as the beginning of the series for

$$\frac{a_0}{\omega}\sin(\omega t)$$

though we need not come to that recognition for the series solution to be valid or useful in its own right.

The solution when $\beta = 0$ can be found in a similar way. We begin by finding the coefficients when $\beta = 0$ with the statement

```
(%i20) for i from 0 thru 10 do c1[i] : ev( c[i], \beta = 0 );
(%i21) for i from 0 thru 10 do print("i = ", i, "c1[i] is    ", c1[i] )$
```

$i = 0$ c1[i] is 0

$i = 1$ c1[i] is 0

$i = 2$ c1[i] is $a_0\omega^2 + 2a_2$

$i = 3$ c1[i] is $a_1\omega^2 + 6a_3$

$i = 4$ c1[i] is $a_2\omega^2 + 12a_4$

$i = 5$ c1[i] is $a_3\omega^2 + 20a_5$

$i = 6$ c1[i] is $a_4\omega^2 + 30a_6$

$i = 7$ c1[i] is $a_5\omega^2 + 42a_7$

$i = 8$ c1[i] is $a_6\omega^2 + 56a_8$

$i = 9$ c1[i] is $a_7\omega^2 + 72a_9$

$i = 10$ c1[i] is $a_8\omega^2 + 90a_{10}$

```
(%i22) for i from 2 thru 10 do s[i] : solve( c1[i]=0, a[i] );
{%i23) s[0] : [ a[0] = a[0] ]$ s[1] : [ a[1] = a[1] ]$
(%i24)  for i from 0 thru 10 do print( s[i] );
```

$\left[\ a_0 = a_0\ \right]$

$\left[\ a_1 = a_1\ \right]$

$$\left[\ a_2 = -\frac{a_0\omega^2}{2}\ \right]$$
$$\left[\ a_3 = -\frac{a_1\omega^2}{6}\ \right]$$
$$\left[\ a_4 = -\frac{a_2\omega^2}{12}\ \right]$$
$$\left[\ a_5 = -\frac{a_3\omega^2}{20}\ \right]$$
$$\left[\ a_6 = -\frac{a_4\omega^2}{30}\ \right]$$
$$\left[\ a_7 = -\frac{a_5\omega^2}{42}\ \right]$$
$$\left[\ a_8 = -\frac{a_6\omega^2}{56}\ \right]$$
$$\left[\ a_9 = -\frac{a_7\omega^2}{72}\ \right]$$
$$\left[\ a_{10} = -\frac{a_8\omega^2}{90}\ \right]$$

Next, we need to recast all of these coefficients so that they depend only on a_0 or a_1 with the statements

```
(%i25) s1[0] : s[0]$ s1[1] : s[1]$
(%i26) for i from 2 thru 10 do s1[i] : ev( s[i], s1[i-2] );
(%i27) for i from 0 thru 10 do print(s1[i] );
```

$$\left[\ a_0 = a_0\ \right]$$
$$\left[\ a_1 = a_1\ \right]$$
$$\left[\ a_2 = -\frac{a_0\omega^2}{2}\ \right]$$
$$\left[\ a_3 = -\frac{a_1\omega^2}{6}\ \right]$$
$$\left[\ a_4 = -\frac{a_0\omega^4}{24}\ \right]$$
$$\left[\ a_5 = -\frac{a_1\omega^4}{120}\ \right]$$
$$\left[\ a_6 = -\frac{a_0\omega^6}{720}\ \right]$$
$$\left[\ a_7 = -\frac{a_1\omega^6}{5040}\ \right]$$
$$\left[\ a_8 = -\frac{a_0\omega^8}{40320}\ \right]$$
$$\left[\ a_9 = -\frac{a_1\omega^8}{362880}\ \right]$$
$$\left[\ a_{10} = -\frac{a_0\omega^{10}}{3628800}\ \right]$$

Neither a_0 nor a_1 is fixed by these conditions. Finally, substituting these solutions into the series with which we began, we find that

```
(%i28) vals : makelist( part(s1[i],1), i, 0, 10 );
```

```
(%o28)
```
$$\left[a_0 = a_0, a_1 = a_1, a_2 = -\frac{a_0\omega^2}{2}, \ldots,\ a_9 = \frac{a_1\omega^8}{362880}, a_{10} = -\frac{a_0\omega^{10}}{3628800}\ \right]$$

```
(%i29) ev( soln, vals, \beta=0 );
```

```
(%o29)
```
$$-\frac{a_0\omega^{10}t^{10}}{3628800} + \frac{a_1\omega^8t^9}{362880} + \frac{a_0\omega^8t^8}{40320} + \ldots - \frac{a_1\omega^2t^3}{6} - \frac{a_0\omega^2t^2}{2} + a_1t + a_0$$

Note that some of the terms have a factor of a_0 and others have a factor of a_1, which we should extract from the expression. We can collect terms with each factor by invoking the statement

```
(%i30) ser : collectterms(%, a[1], a[0] );
```

```
(%o30)
```
$$a_0\left(-\frac{\omega^{10}t^{10}}{3628800} + \frac{\omega^8t^8}{40320} - \frac{\omega^6t^6}{720} + \frac{\omega^4t^4}{24} - \frac{\omega^2t^2}{2} + 1\right)$$
$$+a_1\left(\frac{\omega^8t^9}{362880} - \frac{\omega^6t^7}{5040} + \frac{\omega^4t^5}{120} - \frac{\omega^2t^3}{6} + t\right)$$

The second term in this expression is, of course, the same series we obtained with $\beta = 1$, specifically

$$\frac{a_1}{\omega}\left(\frac{\omega^9t^9}{9!} - \frac{\omega^7t^7}{7!} + \frac{\omega^5t^5}{5!} - \frac{\omega^3t^3}{3!} + \omega t\right) = \frac{a_1}{\omega}\sin(\omega t)$$

(or at least the first few terms in the series expansion of $\sin(\omega t)$). Similarly, recognizing that $2 = 2!$, $24 = 4!$, $720 = 6!$, $40320 = 8!$, and $362880 = 10!$, we conclude that the first term in this expression is

$$a_1\left(-\frac{\omega^{10}t^{10}}{3628800} + \frac{\omega^8t^8}{40320} - \frac{\omega^6t^6}{720} + \frac{\omega^4t^4}{24} - \frac{\omega^2t^2}{2} + 1\right) = a_1\cos(\omega t)$$

(or at least the first few terms in the series expansion of $\cos(\omega t)$). Finally, we conclude that we can write the general solution of the original equation in the simple form

$$x(t) = a_0\cos(\omega t) + \frac{a_1}{\omega}\sin(\omega t)$$

Finally, we want to compare the (truncated) series with the analytic solution, i.e., we want to compare

$$x_{\text{series}}(t) = \left(1 - \frac{\omega^2t^2}{2} + \frac{\omega^4t^4}{24} - \frac{\omega^6t^6}{720} + \frac{\omega^8t^8}{40320} - \frac{\omega^{10}t^{10}}{3628800}\right)a_0$$
$$+\left(t - \frac{\omega^2t^3}{6} + \frac{\omega^4t^5}{120} - \frac{\omega^6t^7}{5040} + \frac{\omega^8t^9}{362880}\right)a_1$$

with

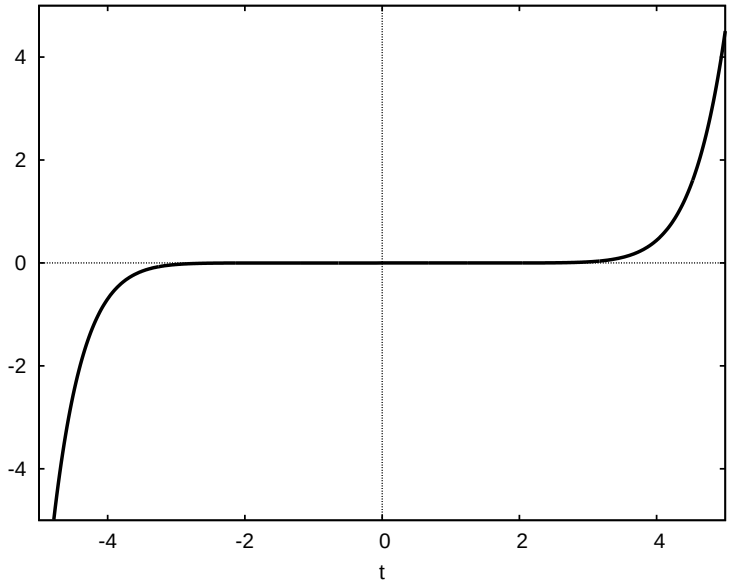
$$x_{\text{analytic}}(t) = a_0\cos(\omega t) + \frac{a_1}{\omega}\sin(\omega t)$$

To simplify the graphing (and avoid the need to specify ω, we introduce the new variable $\tau = \omega t$. If, however, we then replace τ with t, we end up with the same expression we would have had had we simply set $\omega = 1$ in the original expressions, and we elect the latter course, understanding that, after the evaluation, we must interpret t as ωt. That simplification is accomplished with the statement

```
(%i31) xanalytic : a[0]*cos(t) + a[1]*sin(t)$
(%i32) xseries : ev( ser, \omega = 1 )$
```

To compare the two solutions, we now subtract the analytic solution from the series solution and plot a graph of the difference. To create the graph, we randomly select numbers to substitute for a_0 and a_1, say $a_0 = 4$ and $a_1 = 6$. To these ends, we execute the statements

Figure E11.3: Plot of the difference between the analytic solution and the series solution. This plot shows that the two solutions are similar around the origin.



```
(%i33) comparison : ev( xseries - xanalytic, a[0] = 4, a[1] = 6 )$
(%i34) plot2d( comparison, [t, -5, 5], [y,-5.0,5.0], [color, black],
          [style, [lines, 4] );
```

From the resulting graph in Fig. E11.3, we see that the two solutions are the same for a short time. Indeed, they are closely the same within the interval $-3.0 \leq \omega t \leq 3.0$. This result is expected because the polynomial is just an approximation of the known solution. Therefore, since the two solutions are the same for a short time, we conclude that the series solution agrees with the analytic solution.

11.7 Projectile in Linear Air Resistance (MAXIMA)

Exercise: Find a symbolic solution for all three components for the motion of a projectile in a linear, viscous medium, when the initial conditions are general, i.e., solve

$$m\frac{d^2x}{dt^2} = -b\frac{dx}{dt} \quad ; \quad m\frac{d^2y}{dt^2} = -b\frac{dy}{dt} \quad ; \quad m\frac{d^2z}{dt^2} = -mg - b\frac{dz}{dt}$$

subject to the initial conditions

$$x(0) = x_0 \quad ; \quad y(0) = y_0 \quad ; \quad z(0) = 0 \quad ; \quad \frac{dx}{dt}(0) = v_{x0} \quad ; \quad \frac{dy}{dt}(0) = v_{y0} \quad ; \quad \frac{dz}{dt}(0) = v_{z0}$$

Since the equations are uncoupled, you can solve each individually. Alternatively, you can solve the three equations simultaneously as a system. Solve them *both* ways. Once you have the solutions in hand, verify that they satisfy the original equations and initial conditions. Finally, explore their limits for small b.

Solution: The problem defined by the above equations and initial conditions is fairly straightforward. We simply define the equations, solve them, and determine the integration constants to match the initial conditions. These equations are clearly uncoupled, so we can treat the set as three independent second-order ODEs. Appropriate statements to MAXIMA are

```
(%i1) depends( [x, y, z], t );
```

(%o1) $[\,x(t), y(t), z(t)\,]$

```
(%i2) eqx : m*'diff(x, t, 2) = -b*'diff(x,t);
```

(%o2) $$m\,\frac{d^2x}{dt^2} = -b\,\frac{dx}{dt}$$

```
(%i3) eqy : m*'diff(y, t, 2) = -b*'diff(y,t);
```

(%o3) $$m\,\frac{d^2y}{dt^2} = -b\,\frac{dy}{dt}$$

```
(%i4) eqz :  m*'diff(z, t, 2) = -m*g -b*'diff(z, t);
```

(%o4) $$m\,\frac{d^2z}{dt^2} = -b\,\frac{dz}{dt} - mg$$

```
(%i5) solnx :  ode2( eqx, x, t );
```

Is b zero or nonzero? `n;`

(%o5) $x = \%k2\,e^{-bt/m} + \%k1$

```
(%i6) solnx1 : ic2( solnx, t=0, x=x0, 'diff(x,t)=vx0 );
```

(%o6) $$x = \frac{bx_0 + mv_{x0}}{b} - \frac{m\,e^{-bt/m}\,v_{x0}}{b}$$

```
(%i7) solny :  ode2( eqy, y, t );
```

Is b zero or nonzero? `n;`

(%o7) $y = \%k2\,e^{-bt/m} + \%k1$

```
(%i8) solny1 : ic2( solny, t=0, y=y0, 'diff(y,t)=vy0 );
```

(%o8) $$x = \frac{by_0 + mv_{y0}}{b} - \frac{m\,e^{-bt/m}\,v_{y0}}{b}$$

```
(%i9) solnz :  ode2( eqz, z, t );
```

Is b zero or nonzero? `n;`

```
(%o9)
```
$$z = \%k2\, e^{-bt/m} - \frac{bgmt - gm^2}{b^2} + \%k1$$

```
(%i10) solnz1 : ic2( solnz, t=0, z=z0, 'diff(z,t)=vz0 );
```

```
(%o10)
```
$$z = \frac{bz_0 + mv_{z0}}{b} - \frac{e^{-bt/m}(bmv_{z0} + gm^2)}{b^2} - \frac{bgmt - gm^2}{b^2}$$

To verify that each solution satisfies the original equation, we would invoke the MAXIMA statements

```
(%i11) is( ev( eqx, solnx1, diff ) );
(%o11)     true
(%i12) is( ev( eqy, solny1, diff ) );
(%o12)     true
(%i13) is( ev( eqz, solnz1, diff ) );
(%o13)     true
```

Further, we can verify the satisfaction of the initial conditions by invoking the statements

```
(%i14) expand( ev( solnx1, t = 0 ) );
```
```
(%o14)
```
$x = x_0$
```
(%i15) expand( ev( solny1, t = 0 ) );
```
```
(%o15)
```
$y = y_0$
```
(%i16) expand( ev( solnz1, t = 0 ) );
```
```
(%o16)
```
$z = z_0$
```
(%i17) diff(rhs(solnx1), t)$ ev( %, t=0 );
```
```
(%o17)
```
v_{x0}
```
(%i18) diff(rhs(solny1), t)$ ev( %, t=0 );
```
```
(%o18)
```
v_{y0}
```
(%i19) diff(rhs(solnz1), t)$ expand( ev( %, t=0 ) );
```
```
(%o19)
```
v_{z0}

Finally, to determine the behavior of the solutions $x(t)$, $y(t)$, and $z(t)$ for small b, we evaluate Taylor series, going far enough (after some trial) to include the terms of order b^2. We find that[5]

```
(%i20) collectterms( taylor(rhs(solnx1), [b, 0, 2] ), b );
```
```
(%o20)
```
$$(x_0 + v_{x0}t) - \frac{v_{x0}t^2}{2m}\, b + \frac{v_{x0}t^3}{6m^2}\, b^2$$
```
(%i21) collectterms( taylor(rhs(solny1), [b, 0, 2] ), b );
```
```
(%o21)
```
$$(y_0 + v_{y0}t) - \frac{v_{y0}t^2}{2m}\, b + \frac{v_{y0}t^3}{6m^2}\, b^2$$
```
(%i21) collectterms( taylor(rhs(solnz1), [b, 0, 2] ), b );
```
```
(%o21)
```
$$\left(-\frac{gt^2}{2} + tv_{z0} + z_0\right) - \frac{t^2(3v_{z0} - gt)}{6m} b + \frac{t^3(4v_{z0} - gt)}{24m^2} b^2$$

The zeroth-order terms, of course, agree with what we would expect in the absence of air resistance altogether.

In some contexts, one might try to simplify the appearance of the three solutions `solnx1`, `solny1`, and `solnz1` by introducing β for b/m (or, better, τ for m/b) and cancelling the common factors of m that then appear. To that end, we execute the statements

[5] We have rearranged the order of the terms output by MAXIMA to make their significance more apparent.

```
(%i22) solnx2 : expand( subst( m/\tau, b, solnx1 ) );
```

(%o22) $x = x_0 - \tau\, e^{-t/\tau}\, v_{x0} + \tau\, v_{x0}$

```
(%i23) solny2 : expand( subst( m/\tau, b, solny1 ) );
```

(%o23) $y = y_0 - \tau\, e^{-t/\tau}\, v_{y0} + \tau\, v_{y0}$

```
(%i24) solnz2 : expand( subst( m/\tau, b, solnz1 ) );
```

(%o24) $z = z_0 - \tau\, e^{-t/\tau}\, v_{z0} + \tau\, v_{z0} - g\tau^2\, e^{-t/\tau} + g\tau^2 - gt\tau$

These results would be even more transparent if written in the forms

$$\begin{aligned} x &= x_0 + v_{x0}\tau\,(1 - e^{-t/\tau}) \\ y &= y_0 + v_{y0}\tau\,(1 - e^{-t/\tau}) \\ z &= z_0 - gt\tau + (v_{z0} + g\tau)\tau\,(1 - e^{-t/\tau}) \end{aligned}$$

but MAXIMA statements to achieve this last recasting are elusive.

11.12 Vibrating String Fixed at Both Ends (MAXIMA)

Exercise: In an appropriate dimensionless presentation, standing waves in a string must satisfy the boundary value problem

$$\frac{d^2y}{dx^2} + k^2 y = 0 \quad ; \quad y(0) = y(1) = 0$$

Suppose that the interval $0 \le x \le 1$ is divided into n equal segments of length $\Delta x = 1/n$, let $x_i = i\,\Delta x$ (with $i = 0, 1, 2, \ldots, n$), and let $y_i = y(x_i)$. Evaluate the ODE at $x = x_i$, approximate the second derivative with the difference formula

$$\left.\frac{d^2y}{dx^2}\right|_{x=x_i} \approx \frac{y_{i+1} - 2y_i + y_{i-1}}{\Delta x^2} \quad ; \quad i = 1, 2, 3, \ldots n-1$$

and note that $y_0 = y_n = 0$. Show that the values y_i for $i = 0, 1, 2, \ldots, n$ satisfy a system of $n+1$ linear algebraic equations of the form $M\,Y = \alpha\,Y$, where Y is an $(n+1)$-component vector whose elements are the values of y_i and α is determined from k^2 and Δx. Then argue that the allowed values of k^2 can be determined from the eigenvalues of the matrix M. That is, show that this transformation turns a boundary value problem involving a differential equation into an approximately equivalent matrix eigenvalue problem.

Solution: Since the string is fixed at $x = 0$ and $x = 1$, we cannot use Euler's method or a more sophisticated method to address this exercise. Instead, we must divide the interval $0 \le x \le 1$ into n equal segments of length $\Delta x = 1/n$. Then using, $x_i = i\Delta x$ and $y_i = y(x_i)$ (where $i = 0, 1, 2, \ldots, n$), we use the formula in the second equation in the statement of the exercise to approximate the second derivative and seek to solve the family of equations

$$\frac{y_{i+1} - 2y_i + y_{i-1}}{\Delta x^2} = -k^2 y_i$$

To illustrate the procedure, we suppose $n = 4$, but the process is similar for other values of n. In `MAXIMA`, the statements[6]

```
(%i1) eq[0] : y[0] = 0;
```

(%o1) $y_0 = 0$

```
(%i2) eq[4] : y[4] = 0;
```

(%o2) $y_4 = 0$

```
(%i3) for i from 1 thru 3 do eq[i] : (y[i+1]-2*y[i]+y[i-1]) = (-k^2*d^2*y[i]);
```

(%o3) *done*

```
(%i4) for i from 1 thru 3 do print(eq[i]);
```

$$y_2 - 2y_1 + y_0 = -y_1 d^2 k^2$$
$$y_3 - 2y_2 + y_1 = -y_2 d^2 k^2$$
$$y_4 - 2y_3 + y_2 = -y_3 d^2 k^2$$

(%o4) *done*

create the five equations for the five unknowns y_0, y_1, y_2, y_3, y_4. We substitute the first and last of these equation into the second and fourth equation, respectively, with the statements

```
(%i5) eq[1] : ev( eq[1], eq[0]);
```

[6] For simplicity in notation, we use d instead of Δx in this coding.

```
(%o5)
```
$$y_2 - 2y_1 = -y_1 d^2 k^2$$
```
(%i6) eq[3] : ev( eq[3], eq[4]);
```
```
(%o6)
```
$$y_2 - 2y_3 = -y_3 d^2 k^2$$

yielding the three equations $[eq_1, eq_2, eq_3]$. These equations are now in the form $My = -k^2 d^2 y$ where y is the vector $[y_1, y_2, y_3]$ and M is the coefficient matrix for the system of three linear equations for the components of y. Specifically, $-k^2 d^2$ is an eigenvalue of the coefficient matrix M, which is readily extracted with the statement

```
(%i7) M : coefmatrix( [lhs(eq[1]),lhs(eq[2]),lhs(eq[3])], [y[1],y[2],y[3]]);
```

(M)
$$\begin{pmatrix} -2 & 1 & 0 \\ 1 & -2 & 1 \\ 0 & 1 & -2 \end{pmatrix}$$

Non-trivial solutions for the system of three equations $My = 0$ can be found only if $-d^2 k^2$ has one of the three eigenvalues of this 3×3 matrix. The statements

```
(%i8) val : eigenvalues(M);
```
```
(%o8)
```
$$\left[[-\sqrt{2} - 2, \sqrt{2} - 2, -2], [1, 1, 1]\right]$$
```
(%i9) float( val[1] );
```
```
(%o9)
```
$$[-3.414213562373095, -0.5857864376269049, -2.0]$$

find the three eigenvalues, each of which is reported to be non-degenerate, and convert the results to a floating form. Thus, solutions to this approximation for the motion of a string fixed at both ends can be found only when

$$k^2 d^2 = 3.414213562373095 \quad \text{or} \quad 0.5857864376269049 \quad \text{or} \quad 2.0$$

The same procedures can be implemented for much larger values of n to achieve a more accurate determination of the eigenvalues

Chapter 13

Evaluating Integrals

13.1 Non-relativistic Motion under Constant Force (MAXIMA)

Exercise: A particle of mass m moves non-relativistically in one dimension under the action of a constant force f. Starting with the equations

$$p(t) = p_0 + \int_0^t f(t')\,dt' \quad \text{and} \quad x(t) = x_0 + \int_0^t v(t')\,dt'$$

and using symbolic integration, find the position x, velocity v, and momentum p of this particle as functions of time if $x(0) = x_0$ and $v(0) = v_0$.

Solution: Since we are given the force, initial position and initial velocity, we have all we need to begin solving. First, we know that $p_0 = v_0 * m$, so we can make this substitution immediately and solve for $p(t)$. Knowing $p(t)$ as a function of time gives us $v(t)$, which in turn leads us to $x(t)$. The necessary coding in MAXIMA is

```
(%i1) p : p0 + integrate(f,tp,0,t);
```

(p) $f\,t + p_0$

```
(%i2) v : p/m;
```

(v) $$\frac{f\,t + p_0}{m}$$

```
(%i3) v : ev( v, p0=m*v0 );
```

(v) $$\frac{m\,v_0 + f\,t}{m}$$

```
(%i4) x : expand(integrate(v,t,0,t)+x0);
```

(x) $$x_0 + t\,v_0 + \frac{f\,t^2}{2\,m}$$

13.2 Non-relativistic Motion under Constant Force (MAXIMA)

Exercise: A particle of mass m moves non-relativistically in one dimension under the action of a constant force f. Starting with the equations

$$m\,\frac{d^2x}{dt^2} = f(x) \quad \rightarrow \quad mv\frac{dv}{dx} \quad \rightarrow \quad mv\,dv = f(x)\,dx$$

and using symbolic integration, find the position x, velocity v, and momentum p of this particle as functions of time if $x(0) = x_0$ and $v(0) = v_0$.

Solution: Since momentum is derived directly from velocity, it would be prudent at this point to solve for velocity so that we can check later that we actually have the correct solution from MAXIMA. By rearranging terms in the second equation above, we obtain

$$v = \sqrt{\frac{mv_0^2 + 2\int_{x_0}^{x} f(x')\,dx'}{m}}$$

Now that we know what the outcome should be, this problem proceeds as follows: since we know f, we can determine immediately what $\int_{x_0}^{x} f(x')\,dx'$ is, which in turn allows us to solve for v. Solving for v will then yield p as a trivial calculation. The necessary code in MAXIMA is

```
(%i1) integrate( f, xp,x0,x);
```

(%o1) $f(x - x_0)$

```
(%i2) ( (2*%)+m*(v0)^2 )/m;
```

(%o2) $\dfrac{2f(x - x_0) + mv_0^2}{m}$

```
(%i3) v : sqrt(%);
```

(v) $\sqrt{\dfrac{2f(x - x_0) + mv_0^2}{m}}$

```
(%o3) p : expand(m*v);
```

(p) $m\sqrt{-\dfrac{2fx_0}{m} + \dfrac{2f\,x}{m} + v_0^2}$

At this point, we know $v(x)$. Finding $x(t)$ then involves integrating the equation

$$\frac{dx}{dt} = v(x) \qquad \Longrightarrow \int_{x_0}^{x} \frac{dx'}{v(x')} = \int_0^t dt'$$

The second integral is easy in MAXIMA, specifically

```
(%i4) rs : integrate( 1, tp,0,t);
```

(rs) t

The first integral is more complicated, specifically

```
(%i5) ls : integrate( 1/ev(v,x=xp), xp,x0,x);
```

Is $x_0 - x$ positive, negative, or zero? `n;`
Is $f(2fx_0 - 2fX - mv_0^2)$ positive, negative, or zero? `n;`
Is fmv_0^2 positive, negative, or zero? `p;`

Is v_0 positive or negative? `p;`

(ls) $$\frac{m\sqrt{-\frac{2fx_0-2fx-mv_0^2}{m}}}{f}-\frac{mv_0}{f}$$

Finally, we solve for x and v as functions of t with the statements

```
(%i6) solve( ls = rs, x );
```

Is $m(mv_0+ft)$ positive, negative, or zero? `p;`

(%o6) $$\left[x=\frac{2mx_0+2mtv_0+ft^2}{2m}\right]$$

```
(%i7) expand(%);
```

(%o7) $$\left[x=x_0+t\,v_0+\frac{f\,t^2}{2m}\right]$$

```
(%i8) v : diff(rhs(%[1]),t);
```

(%o7) $$v_0+\frac{f\,t}{m}$$

Since $f/m=a$, the acceleration caused by a force f, we recognize these expressions as the familiar results $x(t)=x_0+v_0t+\frac{1}{2}at^2$ and $v(t)=v_0-+at$.

13.3 Non-relativistic Motion under Spring Force (MAXIMA)

Exercise: A particle of mass m moves non-relativistically in one dimension x under the action of a force given by $f(x) = -kx$, where k is a (spring) constant. Starting with

$$m\,\frac{d^2x}{dt^2} = f(x) \quad \rightarrow \quad mv\frac{dv}{dx} \quad \rightarrow \quad mv\,dv = f(x)\,dx$$

and using symbolic integration, find the position x, velocity v, and momentum p of this particle as functions of time if $x(0) = x_0 > 0$ and $v(0) = 0$.

Solution: The inclusion of the spring constant changes this exercise only slightly from the ones before it. Since we know the force on the particle, we can use the second equation to evaluate the integral on $f(x)$, thereby allowing us to rearrange and solve for v, and hence giving us p. The required coding in MAXIMA is

```
(%i1) rths : integrate( -k*xp, xp,x0,x );
```

(rs) $$-k\left(\frac{x^2}{2} - \frac{x_0^2}{2}\right)$$

```
(%i2) lths : integrate(m*vp, vp,v0,v);
```

(ls) $$m\left(\frac{v^2}{2} - \frac{v_0^2}{2}\right)$$

```
(%i3) eq : solve(rths = lths, v);
```

(eq) $$\left[v = -\sqrt{\frac{kx_0^2}{m} - \frac{kx^2}{m} + v_0^2}, v = \sqrt{\frac{kx_0^2}{m} - \frac{kx^2}{m} + v_0^2},\right]$$

Since $dx/dt = v(x)$, the next step in the solution is to evaluate the integrals

$$\frac{dx}{dt} = v(x) \quad \Longrightarrow \quad \frac{dx}{v(x)} = dt \quad \Longrightarrow \quad \int_{x_0}^{x}\frac{dx}{v(x)} = \int_0^t dt = t$$

The second integral is easy in MAXIMA, specifically

```
(%i4) rs : integrate( 1, tp,0,t);
```

(rs) t

The first integral is more complicated. Choosing the *positive* solution for v, we blunder ahead in the simplest approach with the statements

```
(%i5) assume( k>0, m>0 )$
(%i6) ls : integrate(1/rhs(eq[2]), x, x0, x );
     Is v0 positive or negative: p;
     Is x0-x positive, negative, or zero? n;
```

(ls) $$\frac{\sqrt{m}\,\mathtt{asin}\left(\frac{\sqrt{k}x}{\sqrt{kx_0^2 + mv_0^2}}\right)}{\sqrt{k}} - \frac{\sqrt{m}\,\mathtt{asin}\left(\frac{\sqrt{k}x_0}{\sqrt{kx_0^2 + mv_0^2}}\right)}{\sqrt{k}}$$

```
(%i7) soln : solve(rs=ls, x );
```

(soln)
$$\left[x = \frac{\sqrt{kx_0^2 + mv_0^2}\,\sin\left(\texttt{asin}\left(\frac{\sqrt{k}x_0}{\sqrt{kx_0^2+mv_0^2}}\right) + \frac{\sqrt{k}\,t}{\sqrt{m}}\right)}{\sqrt{k}} \right]$$

Along the way we have responded to several questions about various quantities. While k and m are certainly positive, v_0 and $x_0 - x$ could be either. If $v_0 > 0$, then the object *starts off* moving to larger x and $x_0 - x$ will be negative for the initial motion. Strictly, the solution we have obtained applies only to those particular choices.

Let's simplify this expression significantly by recognizing that the arcsine in the expression is a constant phase and substitute *phi* for that phase with the statement

```
(%i8) tmp : subst(phi, asin(sqrt(k)*x0/sqrt(k*x0^2+m*v0^2)), soln );
```

(tmp)
$$\left[x = -\frac{\sin(\sqrt{k}t/\sqrt{m} + \phi)\sqrt{kx_0^2 + mv_0^2}}{\sqrt{k}} \right]$$

We also substitute ω for the frequency $\sqrt{k/m}$ with the statement

```
(%i9) tmp1 : subst(sqrt(k)/omega, sqrt(m), tmp );
```

(tmp1)
$$\left[x = \frac{\sin(\omega t + \phi)\sqrt{kx_0^2 + mv_0^2}}{\sqrt{k}} \right]$$

We note next that $\sqrt{k}$ in the denominator could be divided into the constant multiplier in the numerator to effect a yet simpler expression. To that end, we recognize that $m = k/\omega^2$ and execute the statement

```
(%i10) tmp2 : subst( k/omega^2, m, tmp1 );
```

(tmp2)
$$\left[x = \frac{\sin(\omega t + \phi)\sqrt{kx_0^2 + kv_0^2/\omega^2}}{\sqrt{k}} \right]$$

Finally, we note that there is a factor of $\sqrt{k}$ in both numerator and denominator. The statement

```
(%i11) tmp3 : subst( 1, k, tmp2 );
```

(tmp3)
$$\left[x = \sin(\omega t + \phi)\sqrt{x_0^2 + v_0^2/\omega^2} \right]$$

This sinusoidal oscillation at frequency $\omega = \sqrt{k/m}$ and amplitude $\sqrt{x_0^2 + v_0^2/\omega^2}$ is exactly what we anticipated. Technically, the above analysis applies only when $v > 0$, i.e., during that portion of the oscillation before the object reaches its maximum positive excursion. Conveniently, however, it turns out to apply to the entire motion.

To find the velocity and momentum as functions of time, we recognize that $v = dx/dt$ and $p = mv$, so the statements

```
(%i12) v : diff(rhs(tmp3[1]),t);
```

(v) $\omega\,\cos(\omega t + \phi)\,\sqrt{x_0^2 + v_0^2/\omega^2}$

```
(%i13) p : m*v;
```

(p) $m\omega\,\cos(\omega t + \phi)\,\sqrt{x_0^2 + v_0^2/\omega^2}$

13.4 Non-Relativistic Motion with Decaying Force (MAXIMA)

Exercise: Suppose an object of mass m moves non-relativistically in one dimension under the action of the force $f(t) = f_0 e^{-bt}$, where both b and f_0 are positive. Let $x(0) = x_0$ and $v(0) = v_0$. Use symbolic integration to find $x(t)$ and $v(t)$ by evaluating the integrals in the equations

$$p(t) = p_0 + \int_0^t f(t')\,dt' \quad \text{and} \quad x(t) = x_0 + \int_0^t v(t')\,dt'$$

Then, find and interpret both the limits of these two results as $t \to \infty$ and the Taylor expansion of these two results for small t.

Solution: When the force experienced by a particle moving non-relativistically is given by

$$f(t) = f_0 e^{-bt}$$

the dynamic problem to be solved to find the motion is conveyed by the equations

$$m\frac{d^2x}{dt^2} = f_0 e^{-bt} \quad ; \quad x(0) = x_0 \quad ; \quad v(0) = v_0$$

where $v(t) = dx(t)/dt$. To find this solution by direct integration, we begin by recognizing from Eq. (13.3) that

$$p(t) = mv(t) = mv_0 + \int_0^t f(t')\,dt' = mv_0 + \int_0^t f_0\, e^{-bt'}\,dt'$$

and we evaluate this integral in MAXIMA with the statements

```
(%i1) f : f0*exp(-b*tp)$
(%i2) p : m*v0 + integrate( f, tp, 0, t );
```

(%o2) $$mv_0 + f_0\left(\frac{1}{b} - \frac{e^{-bt}}{b}\right)$$

Then, remembering that

$$x(t) = x_0 + \int_0^t v(t')\,dt'$$

and that, non-relativistically, $p(t) = mv(t)$ and mindful also of the need to keep integration variables separate from the limits of integration, we determine the velocity and integrate the result one more time with the statements

```
(%i3) v : expand( p/m );
```

(%o3) $$v_0 - \frac{f_0\, e^{-bt}}{bm} + \frac{f_0}{bm}$$

```
(%i4) vp : ev( v, t = tp)$
(%i5) x : collectterms( expand(x0 + integrate( vp, tp, 0, t )), t);
```

(%o5) $$x_0 + t\left(v_0 + \frac{f_0}{bm}\right) + \frac{f_0\, e^{-bt}}{b^2 m} - \frac{f_0}{b^2 m}$$

The limits of these expressions as $t \to \infty$ are desired. We can evaluate that limit of the velocity v with the single statement

```
(%i6) vlim : expand( limit( v, t, inf );
Is b positive, negative, or zero? p;
```

(%o6) $$v_0 + \frac{f_0}{bm}$$

For the position, the similar statement

```
(%i7) xlim : limit( x, t,inf );
Is b positive, negtive, or zero? p;
Is f0m positive, negative, or zero? p;
Is v0 poisitive, negative, or zero? p;
```

(%o7) ∞

We find that the velocity approaches a constant limiting value while the position increases in magnitude indefinitely. While it is not evident at (%o7), examination of (%o5) leads to the conclusion that, as $t \to \infty$, x becomes large and positive if $v_0 + f_0/(bm) > 0$ and large and negative if $v_0 + f_0/(bm) < 0$. In the special case that $v_0 = -f_0/bm$, the terminal velocity is zero and, looking again at the expression for x, we conclude that the object comes ultimately to rest at the position $x = x_0 - f_0/b^2m$.

To evaluate the limit as t becomes small, we choose to evaluate a Taylor series rather than look specifically to the limit. We invoke the statements

```
(%i8) expand( taylor( x, t, 0, 3 );
```

(%o8) $$x_0 + t\,v_0 - \frac{bf_0t^3}{6m} + \frac{f_0t^2}{2m}$$

```
(%i9) expand( taylor( v, t, 0, 2 );
```

(%o9) $$v_0 - \frac{bf_0t^2}{2m} + \frac{f_0t}{m}$$

We find that the first few terms in these expansions coincide with what we would expect for a constant force f_0 but then, as t becomes a bit larger (and the force becomes smaller), the position and the velocity depart from those appropriate to constant force. Not surprisingly, for a short while this motion looks like motion under a constant force.

13.5 Properties of Lorentz Distribution (MAXIMA)

Exercise: The normalized Lorentz distribution function is given by

$$p(x) = \frac{1}{\pi}\frac{a/2}{x^2 + (a/2)^2}$$

Using symbolic integration, (a) verify that $\int_{-\infty}^{+\infty} p(x)\,dx = 1$, (b) evaluate—as best you can—the average $\overline{x}$ and variance σ^2, defined by

$$\overline{x} = \lim_{b\to\infty}\int_{-b}^{+b} x\,p(x)\,dx \quad \text{and} \quad \sigma^2 = \lim_{b\to\infty}\int_{-b}^{+b} (x-\overline{x})^2\,p(x)\,dx$$

for this distribution, and (c) find the probability that a single, randomly selected value will lie in the range $-a \le x \le a$. Finally, (d) show *analytically* that you should have expected the result of part (c) to be independent of a. *Hint*: Introduce the dimensionless variable $\lambda = x/a$.

Solution: The normalized Lorentz distribution function is

$$p(x) = \frac{1}{\pi}\frac{a/2}{x^2 + (a/2)^2}$$

(a) Using symbolic integration, we can show that $\int_{-\infty}^{\infty} p(x)\,dx = 1$ by setting up the integrand in MAXIMA and evaluating the integral with the limits $-\infty$ to $+\infty$. We execute the statements

```
(%i1) assume(a>0)$
(%i2) int : (1/(%pi)) * (a/2)/(x^2 + (a/2)^2)$
(%i3) integrate(int, x, minf, inf);
(%o3)     1
```

(b) The average $\bar{x}$ and the variance σ^2 are defined by

$$\bar{x} = \int_{-\infty}^{\infty} x\,p(x)\,dx \qquad \text{and} \qquad \sigma^2 = \int_{-\infty}^{\infty} (x-\bar{x})^2 p(x)\,dx$$

respectively. To find $\bar{x}$, we invoke the statement

```
(%i4) integrate( x*int, x, minf, inf );
Principal Value
(%o4)      0
```

To evaluate σ^2, we substitute the value for $\bar{x}$ and evaluate the integral defining σ^2 with the statements

```
(%i5) integrate( x^2*int, x, minf, inf );
defint: integral is divergent.
 – an error. To debug this try: debugmode(true);
```

Alternatively, we can obtain a slightly more satisfying result by executing the statements

```
(%i6) integrate( int*x^2, x, -b, b );
Is b positive, negative, or zero? p;
```

$$\text{(\%o6)} \qquad \frac{a}{2\pi}\left(2b - a\arctan\left(\frac{2b}{a}\right)\right)$$

```
(%i7) limit(%, b, inf);
(%o7)      ∞
```

As we might have expected from the nature of the integrand in the integral giving σ^2, the variance diverges for the Lorentz distribution.

(c) To find the probability that a single, randomly selected value will be in the range $-a \leq x \leq a$, we need to evaluate the integral $\int_{-a}^{a} p(x)\,dx$. Since the integrand has already been defined earlier in this exercise, we can continue our session with the statements

```
(%i8) integrate( int, x, -a, a );
```

$$\text{(\%o8)} \qquad \frac{2\,\arctan(2)}{\pi}$$

```
(%i9) float(%);
```

```
(%o9)      0.7048327646991335
```

(d) To show analytically that part (c) is independent of a, we introduce the dimensionless variable $\lambda = x/a$. Since $d\lambda = dx/a$, the new form of our equation is

$$\int_{-a}^{a} p(x)\,dx = \frac{1}{2\pi}\int_{-1}^{1} \frac{1}{\lambda^2 + (1/4)}\,d\lambda$$

Since all trace of a can be absorbed in this form, we know that the value of the integral is independent of a.

13.6 Earth Falling into Sun (MAXIMA)

Exercise: Suppose some cataclysmic event stops the earth dead in its tracks and, responding to the sun's gravitational attraction, the earth falls into the sun. Using symbolic integration, find the time required for the earth to fall over the middle half of its journey to the sun. Expressed in years, what is the value of this time for the earth-sun system? *Hint*: Since the gravitational potential is $-GmM/x$, conservation of energy yields

$$\frac{1}{2}m\left(\frac{dx}{dt}\right)^2 - G\frac{mM}{x} = -G\frac{mM}{x_0} \qquad \Longrightarrow \qquad \frac{dx}{dt} = -\sqrt{2GM}\,\sqrt{\frac{1}{x} - \frac{1}{x_0}}$$

(The negative square root is taken because x, the distance to the sun, is known to be decreasing.) This expression then leads to the value

$$T_{\text{midhalf}} = \frac{1}{\sqrt{2GM}}\int_{x_0/4}^{3x_0/4}\left(\frac{1}{x} - \frac{1}{x_0}\right)^{-1/2} dx$$

Hint: The evaluation will be simpler if you begin by recasting the problem in dimensionless terms, expressing lengths in units of x_0 and times in units of $\sqrt{x_0^3/(2GM)}$. To interpret the significance of this unit of time, determine the period of a circular orbit of radius x_0, which will turn out to be $2\pi\sqrt{x_0^3/GM}$. For the earth around the sun, this latter time is, of course, 1 year. *Optional*: Evaluate the time required for the first half of the journey, which involves a convergent but improper integral.

Solution: Should the earth stop, the time it would take for it to fall towards the sun over the center half of its path is given by the integral

$$T_{\text{midhalf}} = \frac{1}{\sqrt{2GM}}\int_{x_0/4}^{3x_0/4}\left(\frac{1}{x} - \frac{1}{x_0}\right)^{-1/2} dx$$

To evaluate this integral symbolically in MAXIMA, we set up the integrand and integrate. Recall that the distances, although decreasing, are positive. We invoke the statements

```
(%i1) int: 1/sqrt(1/x - 1/x0)$
(%i2) integrate( int, x, x0/4, 3*x0/4 );
```

Is x_0 positive, negative or zero? `p;`

(%o2) $$\frac{(4\pi + 3^{3/2})x_0^{3/2}}{12} - \frac{(2\pi + 3^{3/2})x_0^{3/2}}{12}$$

```
(%i3) ratsimp(%);
```

(%o3) $$\frac{\pi x_0^{3/2}}{6}$$

```
(%i4) % / sqrt(2*G*M);
```

(%o4) $$\frac{\pi x_0^{3/2}}{3\ 2^{3/2}\sqrt{MG}}$$

Now, to find the period of a circular orbit at radius x_0, we first require that the gravitational force match the required centripetal force, i.e., that

$$\frac{mv^2}{x_0} = \frac{GmM}{x_0^2} \qquad \Longrightarrow \qquad v = \sqrt{\frac{GM}{x_0}}$$

and then compute the period from the relationship

$$T_0 = \frac{2\pi x_0}{v} = \frac{2\pi x_0}{\sqrt{GM/x_0}} = \frac{2\pi x_0^{3/2}}{\sqrt{GM}}$$

To cast the above result as a multiple of T_0, we execute the statements

```
(%i5) T0 : 2*%pi*x0^(3/2)/sqrt(G*M);
```

(%o5) $$\frac{2\pi x_0^{3/2}}{\sqrt{GM}}$$

```
(%i6) %o4/T0;
```

(%o6) $$\frac{1}{3\ 2^{5/2}}$$

```
(%i7) float(%);
(%o7)      0.05892556509887893
(%i8) %*365;
(%o8)      21.50783126109081
```

We find that the time required for the earth to fall towards the sun through the middle half of its journey is given symbolically by $\pi x_0^{3/2}/(3\ 2^{3/2}\sqrt{2GM})$ and numerically by 0.0589 years or 21.51 days.

The time required for the first half of the journey can be found by changing the upper limit of integration to x_0 and the lower limit to $x_0/2$. To evaluate, we continue our session with the statements

```
(%i9) integrate( int, x, x0/2, x0 );
```

Is x_0 positive, negative or zero? `p;`

(%o9) $$\frac{(\pi+2)\,x_0^{3/2}}{4}$$

```
(%i10) % / sqrt(2*G*M);
```

(%o10) $$\frac{(\pi+2)\,x_0^{3/2}}{2^{5/2}\sqrt{GM}}$$

Again, casting in terms of the period of the circular orbit of radius x_0, we find that this time is given by

```
(%i11) %/T0;
```

(%o11) $$\frac{\pi+2}{2^{7/2}\,\pi}$$

```
(%i12) float(%);
(%o12)      0.1446581174081375
(%i13) %*365;
(%o13)      52.80021285397019
```

We find that the time required for the earth to fall towards the sun through the first half of its journey is given symbolically by $(\pi+2)x_0^{3/2}/2^{5/2}\,\sqrt{GM}$ and numerically by 0.1447 years or 52.80 days.

Finally, let us find the time required for the earth to fall all the way from its starting point to the sun. We execute the statements

```
(%i14) integrate( int, x, 0, x0 );
```

Is x_0 positive, negative or zero? `p;`

```
(%o14)
```
$$\frac{\pi x_0^{3/2}}{2}$$

```
(i15) % / sqrt(2*G*M);
```

```
(%o15)
```
$$\frac{\pi x_0^{3/2}}{2^{3/2}\sqrt{GM}}$$

```
(%i16) %/ T0;
```

```
(%o16)
```
$$\frac{1}{2^{5/2}}$$

```
(%i17) float(%);
(%o17)      0.1767766952966368
(%i18) %*365;
(%o18)      64.52349378327244
```

From start to finish in this calamity requires a mere 64.5 days!

13.7 Electron inside Bohr Radius (MAXIMA)

Exercise: According to the quantum theory, the probability that the electron in the ground state of the hydrogen atom will be found between the center of the atom and some radius r is given by

$$P(r) = \frac{4}{a^3}\int_0^r e^{-2r'/a} r'^2 dr' = 4\int_0^{r/a} e^{-2\rho}\,\rho^2\,d\rho$$

where a is the Bohr radius and $\rho = r'/a$. Evaluate this integral symbolically. Then plot and comment on a graph of $P(r)$ versus r/a.

Solution: In this exercise, we examine the probability that an electron in the ground state of the hydrogen atom will be found between the center of the atom and some radius r. This probability is given by the equation

$$P(r) = 4\int_0^{r/a} e^{-2\rho}\rho^2\,d\rho$$

where a is the Bohr radius. To explore the probability in MAXIMA as a function of r/a, we must identify the integrand and evaluate the integral. Introducing the dimensionless variable $s = r/a$, we excecute the statements

```
(%i1) int : exp(-2*\rho)*\rho^2$
(%i2) p : 4 * integrate( int, \rho, 0, s );
```

Is s positive, negative or zero? `p;`

(%o2) $$4\left(\frac{1}{4} - \frac{(2s^2+2s+1)\,e^{-2s}}{4}\right)$$

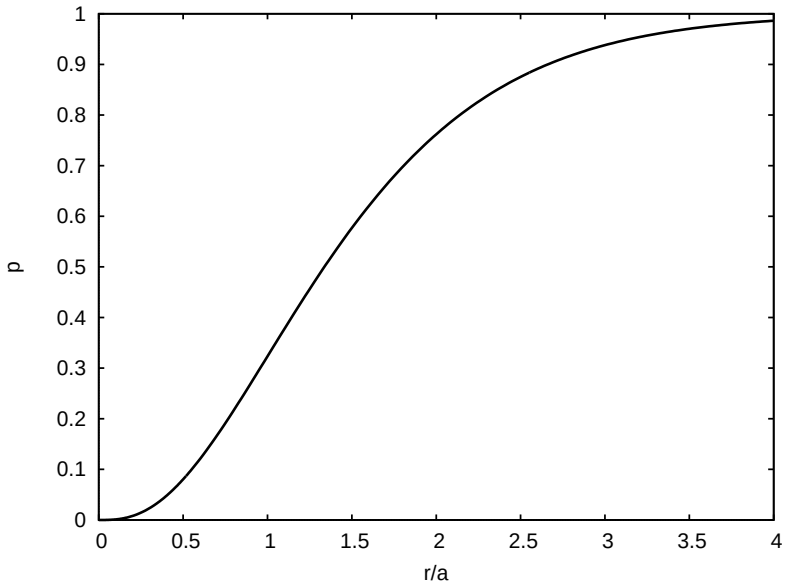
```
(%i3) p :  multthru(%);
```

(%o3) $$1 - (2s^2+2s+1)\,e^{-2s}$$

```
(%i4) plot2d( p, [s, 0.0, 4.0], [xlabel, "r/a"], [ylabel, "p"],
              [style, [lines, 3]], [color, black] );
```

The first statement `(c1)` sets up the integrand and binds it to the variable `int`. The second statement `(c2)` evaluates the integral from 0 to s, and the third `(c3)` rearranges the result into a more convenient form. Then, using the last statement `(c4)`, in which we allow the values of s to range from 0.0 to 4.0, we plot p as a function of s. The resulting graph is shown in Fig. E13.1. Note that the value of $P(s)$ goes to 1 as s approaches infinity.

Figure E13.1: Probability that an electron in the ground state of a hydrogen atom will be found inside the radius r.



13.8 On-Axis Potential of a Pair of Disks (MAXIMA)

Exercise: Consider a source consisting of two uniformly charged disks, each of radius a and each oriented with its center on the z axis and its plane perpendicular to the z axis. Let one disk have its center at (0,0,ca) and carry a positive charge density σ and the other have its center at (0,0,$-ca$) and carry a negative charge density $-\sigma$. Using a symbolic program, show that the on-axis electrostatic potential established by this source is given by

$$V(z) = \frac{\sigma}{2\epsilon_0}\left[\sqrt{a^2+(z-ca)^2} - |z-ca| - \sqrt{a^2+(z+ca)^2} + |z+ca|\right]$$

and then explore this potential as a function of z/a for various values of c. *Hint*: First find the on-axis potential of a single disk lying in the xy plane, and then construct the desired potential by superposition.

Solution: Consider first the on-axis electrostatic potential of a *single* charged disk of radius a located in the xy plane with its center at the origin. Then, the vectors $\mathbf{r}$ and $\mathbf{r}'$ locating the observation point on the z axis and an element of the source whose polar coordinates in the xy plane are (r', ϕ') are

$$\mathbf{r} = z\,\hat{\mathbf{k}} \qquad ; \qquad \mathbf{r}' = r'\cos\phi'\,\hat{\mathbf{i}} + r'\sin\phi'\,\hat{\mathbf{j}}$$

and an element of the source subtending the angle $d\phi'$ from the center of the circle has area $r'\,d\phi'\,dr'$ and, with uniform charge density σ, carries charge $dq' = \sigma r'\,d\phi'\,dr'$. Further, according to Eq. (13.28), the electrostatic potential at this observation point is given by

$$V(z) = \frac{\sigma}{4\pi\epsilon_0}\int_0^{2\pi}\int_0^a \frac{r'\,dr'\,d\phi'}{|\mathbf{r}-\mathbf{r}'|}$$

We need first to determine the integrand and then do the double integral. We invoke the MAXIMA statements[1]

```
(%i1) r0 : [ 0, 0, z ]$                          Set observation point.
(%i2) rp : [ r*cos(\phi), r*sin(\phi), 0 ]$      Set source point.
(%i3) sep : r0 + rp$                             Evaluate r - r'.
(%i4) trigsimp( sep.sep)$                        Evaluate |r - r'|^2.
(%i5) den : sqrt(%);                             Recognize cos^2 + sin^2 = 1
```

(den) $\sqrt{z^2+r^2}$ Calculate denominator in integral.

```
(%i6) cnst : 1/(4*%pi*\epsilon[0])$
(%i7) integ : \sigma*cnst*r/den;                 Set constant.
```

(integ) $\dfrac{r\sigma}{4\pi\epsilon_0\sqrt{z^2+r^2}}$ Evaluate integrand.

```
(%i8) integrate( integ, r, 0, a );               Evaluate integrals, doing the radial
Is a positive, negative, or zero? p;             integral first.
Is z positive or negative? p;
```

(%o8) $\dfrac{\sigma(\sqrt{z^2+a^2}-z)}{4\pi\epsilon_0}$

```
(%i9) single : integrate( %,
                    \phi, 0 ,2*%pi );
```

(single) $\dfrac{\sigma\,(\sqrt{z^2+a^2}-z)}{2\epsilon_0}$

Since z may alternatively be negative, we redo the above evaluation specifying instead that $z < 0$ to find that

[1] In this coding, we use `r` and `phi` for r' and ϕ'.

```
(%i10) integrate( integ, r, 0, a );
Is a positive, negative, or zero? p;
Is z positive or negative? n;
```

(%o10) $$\frac{\sigma(\sqrt{z^2+a^2}+z)}{4\pi\epsilon_0}$$

```
(%i11) single : integrate( %, \phi, 0 ,2*%pi );
```

(single) $$\frac{\sigma\left(\sqrt{z^2+a^2}+z\right)}{2\epsilon_0}$$

Comparing the two results for `single` above, we conclude that the expression

```
(%i12) single : (\sigma*(sqrt(z^2+a^2)-abs(z)))/(2*\epsilon[0]);
```

(single) $$\frac{\sigma\left(\sqrt{z^2+a^2}-|z|\right)}{2\epsilon_0}$$

covers both cases.

Now, we are ready to construct from this potential the potential produced by a pair of identical disks, one carrying charge density σ and located at $z = ca$ and the other carrying charge density $-\sigma$ and located at $z = -ca$. We need simply recognize that the symbol z in the above evaluation represents the coordinate of the observation point relative to the center of the disk. Thus, we can construct the potential established by the pair of disks located as described from that of the single disk we have calculated simply by combining two pieces, for the first of which we replace z in the above evaluation with $z - ca$ and for the second of which we replace z with $z + ca$ *and* σ with $-\sigma$. Equivalently, we subtract the above expression with $z + ca$ replacing z from the above expression with $z - ca$ replacing z. The appropriate MAXIMA statements are

```
(%i13) part1 : ev( single, z = z - c*a )$
(%i14) part2 : ev( single,  z = z + c*a )$
(%i15) double : collectterms( part1 - part2, \sigma );
```

(double) $$\sigma\left(\frac{\sqrt{(z-ac)^2+a^2}-|z-ac|}{2\epsilon_0}-\frac{\sqrt{(z+ac)^2+a^2}-|z+ac|}{2\epsilon_0}\right)$$

To explore this result for various values of c, it is wise to cast it in a dimensionless form by introducing the coordinate $z_d = z/a$ and measuring the potential in units of $\sigma a/2\epsilon_0$. Thus, we want ultimately to replace z in the above expression with az_d and divide the result by $\sigma a/2\epsilon_0$. Unfortunately, because of the way the factors appear in `(%o12)`, we must first divide `tmp` by σ, then substitute az_d for z, next multiply by $2\epsilon_0$, and finally divide by a. The statement

```
(%i16) tmp : multthru( 2*\epsilon[0], ev( double/\sigma, z = a*zd) )/a;
```

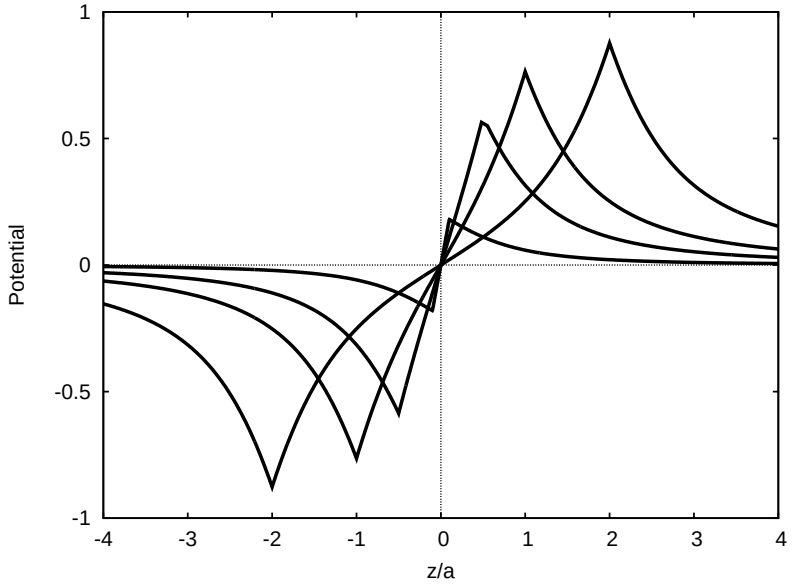
(tmp) $$\frac{|az_d+ac|-|az_d-ac|-\sqrt{(az_d+ac)^2+a^2}+\sqrt{(az_d-ac)^2+a^2}}{a}$$

achieves this rearrangement. Clearly, there is now a common factor of a in every term in both the numerator and the denominator. (Remember that $a > 0$.) To complete the translation of this expression to a dimensionless form, we need to remove the redundant factors a, thereby removing a from the expression altogether. Simple setting $a = 1$ achieves the same end result. Rather than wrestle with persuading MAXIMA to factor the a out of each term in the numerator *without expanding the binomials under the square roots*, we elect the easier route with the statement

```
(%i17) ev( tmp, a=1 );
```

(%o17) $$|z_d+c|-|z_d-c|-\sqrt{(z_d+c)^2+1}+\sqrt{(z_d-c)^2+1}$$

Figure E13.2: On-axis electrostatic potential of two charged disks. Counting inwards from the one with the cusp at $z/a = 2$, the graphs correspond respectively to $c = 2.0$, $c = 1.0$, $c = 0.5$, and $c = 0.1$.



At long last, we can plot this potential as a function of z_d for various values of c. We invoke the statements

```
(%i18) pot01 : ev( pot, c = 0.1 );
(%i19) pot05 : ev( pot, c = 0.5 );
(%i20) pot10 : ev( pot, c = 1.0 );
(%i21) pot20 : ev( pot, c = 2.0 );
(%i22) plot2d( [pot01, pot05, pot10, pot20], [zd, -4.0, 4.0], [y, -1.0, 1.0 ],
        [legend, false], [color, black], [style, [lines, 4] ], [xlabel, "z/a"],
        [ylabel, "Potential"])
```

The resulting graph is shown in Fig. E13.2. Note that, for the smaller values of c (at which the separation of the disks is small compared to their common radius), the potential varies linearly as we walk from one disk to the other in the region between them. As their separation grows relative to their radius, that linearity is gradually destroyed. In all cases, however, the potential remains zero at the on-axis point midway between the two disks.

13.10 Properties of Cardioid (MAXIMA)

Exercise: Consider a surface in the xy plane having uniform mass density σ and having the shape of a cardioid given in polar coordinates by the function $r(\phi) = a(1 - \cos\phi)$. Using symbolic integration, find (a) the center of mass of this object, (b) the moment of inertia tensor of this object about the x, y, and z axes, and (c) the radius of gyration about the z axis. *Hints*: The center of mass is defined in Section 13.1.2; the moment of inertia tensor is a 3×3 tensor whose ij element is given by

$$I_{ij} = \int [(x_1^2 + x_2^2 + x_3^2)\delta_{ij} - x_i x_j]\, dm$$

where x_1, x_2, and x_3 symbolize x, y, and z, respectively; δ_{ij} is the Kronecker delta, which has the value 1 when $i = j$ and the value 0 otherwise; and the radius of gyration is defined in Section 13.1.3.

Solution: (a) The location of the center of mass of a planar object is given in polar coordinates by the equation

$$\mathbf{r}_{\rm cm} = \frac{1}{M}\int \mathbf{r}\, dm = \frac{1}{M}\int \sigma(\mathbf{r},\phi)\; r\; \mathbf{r}\, dr\, d\phi$$

In this case, σ is a constant. We can write $\mathbf{r} = r\cos\phi\ \hat{\mathbf{i}} + r\sin\phi\ \hat{\mathbf{j}}$ and then evaluate the integral

$$\mathbf{r}_{\rm cm} = \frac{\sigma}{M}\int_0^{2\pi}\int_0^{a(1-\cos\phi)} (r\cos\phi\ \hat{\mathbf{i}} + r\sin\phi\ \hat{\mathbf{j}})\; r\, dr\, d\phi$$

To do this in MAXIMA, we define the x, y and z components of $\mathbf{r}$, construct the vector $\mathbf{r}$, and integrate by invoking the statements

```
(%i1) x : r*cos(\phi)$
(%i2) y : r*sin(\phi)$
(%i3) z : 0$
(%i4) vec : [x, y, z]$
(%i5) integrate( integrate( r*vec, r, 0, a*(1-cos(\phi)) ), \phi, 0, 2*%pi );
```

(%o5) $$\left[\frac{-5\pi a^3}{4}, 0, 0\right]$$

Note that the integral over r must be done first since the upper limit of r has a dependence on ϕ.

To find the total mass M, we need to integrate $dm = \sigma\, r\, dr\, d\phi$. To that end, we execute the statement

```
(%i6) m : integrate( integrate( \sigma*r, r, 0, a*(1-cos(\phi)) ), \phi, 0, 2*%pi );
```

m) $$\frac{3\pi a^2\sigma}{2}$$

Finally, we calculate the position of the center of mass with the statement

```
(%i7) (\sigma/m)*\%o5
```

(%o7) $$\left[\frac{-5a}{6},\ 0,\ 0\right]$$

The center of mass is therefore located at $\mathbf{r}_{\rm cm} = [-5a/6,\ 0,\ 0]$.

(b) The moment of inertia tensor is a 3×3 matrix whose components are given by

$$I_{ij} = \int [(x_1^2 + x_2^2 + x_3^2)\delta_{ij} - x_i x_j]\, dm = \sigma \int [(x_1^2 + x_2^2 + x_3^2)\delta_{ij} - x_i x_j]\, r\, dr\, d\phi$$

where x_1, x_2 and x_3 correspond to x, y and z, respectively, and δ_{ij} is the Kronecker delta. The integrand in square brackets is therefore an element of the matrix

$$\begin{bmatrix} y^2+z^2 & -xy & -xz \\ -yx & x^2+z^2 & -yz \\ -zx & -zy & x^2+y^2 \end{bmatrix}$$

Since x, y and z were defined in part (a), we need only enter this matrix and evaluate the integrals. Appropriate statements to MAXIMA are

```
(%i8) t : matrix( [y^2+z^2, -x*y, -x*z],
                  [-y*x, x^2+z^2, -y*z],
                  [-z*x, -z*y, x^2+y^2] );
```

(t)
$$\begin{bmatrix} \sin^2(\phi)r^2 & -\cos(\phi)\sin(\phi)r^2 & 0 \\ -\cos(\phi)\sin(\phi)r^2 & \cos^2(\phi)r^2 & 0 \\ 0 & 0 & \sin^2(\phi)r^2+\cos^2(\phi)r^2 \end{bmatrix}$$

```
(%i9) integrand : sigma*trigsimp(t);
```

(integrand)
$$\begin{bmatrix} \sin^2(\phi)r^2\sigma & -\cos(\phi)\sin(\phi)r^2\sigma & 0 \\ -\cos(\phi)\sin(\phi)r^2\sigma & \cos^2(\phi)r^2\sigma & 0 \\ 0 & 0 & r^2\sigma \end{bmatrix}$$

```
(%i10) i : integrate( integrate( integrand*r, r, 0, a*(1-cos(\phi)) ), \phi, 0, 2*%pi );
```

(i)
$$\begin{bmatrix} 21\pi a^4\sigma/32 & 0 & 0 \\ 0 & 49\pi a^4\sigma/32 & 0 \\ 0 & 0 & 35\pi a^4\sigma/16 \end{bmatrix}$$

Finally, let us express this result in terms of the total mass of the cardioid by replacing σ with $2M/(3\pi a^2)$. Unfortunately, we have already bound a value to the MAXIMA variable `m`, so we can't (easily) use that variable again. Let us represent the total mass by the symbol `mm`. Then the statement

```
(%i11) i : subst( 2*mm/(3*%pi*a^2), sigma, i );
```

(i)
$$\begin{bmatrix} 7a^2M/16 & 0 & 0 \\ 0 & 49a^2M/48 & 0 \\ 0 & 0 & 35a^2M/24 \end{bmatrix}$$

recasts the above moment of inertia tensor in terms of the total mass (which we symbolize by M in the displayed output).

(c) The radius of gyration about a particular axis is the distance from the axis at which the total mass of the object would have to be placed (as a point) for it to have the same moment of inertia about that axis as the object itself. Since—see `(d11)`—we found I in terms of the total mass M in part (b), we need only solve the equation $I = Mk^2$ for k. To do this, we continue our MAXIMA session with the statement

```
(%i12) assume(a > 0); sqrt( i/mm );
```

(%o12)
$$\begin{bmatrix} \sqrt{7}\,a/4 & 0 & 0 \\ 0 & 7\,a/4\sqrt{3} & 0 \\ 0 & 0 & \sqrt{35}\,a/2\sqrt{6} \end{bmatrix}$$

The diagonal elements of this matrix are the radii of gyration associated with the three coordinate axes x, y, and z, respectively. Thus , $\sqrt{7}|a|/4$ is the distance from the x-axis, $7|a|/(4\sqrt{3})$ is the distance from the y-axis, and $\sqrt{35}|a|/(2\sqrt{6})$ is the distance from the z-axis at which the total mass would have to be placed for it to have the same moment of inertia about the axis as the object itself has.

13.11 Oscillator Matrix Elements (MAXIMA)

Exercise: In quantum mechanics, the two integrals

$$x_{mn} = \int_{-\infty}^{\infty} \psi_m^*(x)\, x\, \psi_n(x)\, dx \quad \text{and} \quad p_{mn} = \int_{-\infty}^{\infty} \psi_m^*(x)\, \frac{\hbar}{i}\frac{\partial}{\partial x}\psi_n(x)\, dx$$

are important in a variety of contexts. For a particle in an infinitely deep potential well that extends over the region $-a \le x \le a$,

$$\psi_n(x) = \begin{cases} \dfrac{1}{\sqrt{a}} \cos\dfrac{n\pi x}{2a} & n = 1, 3, 5, \ldots \\ \dfrac{1}{\sqrt{a}} \sin\dfrac{n\pi x}{2a} & n = 2, 4, 6, \ldots \end{cases}$$

Using symbolic integration, show that, for these wave functions, $x_{mn} = 0$ and $p_{mn} = 0$ when m and n are *both* even or *both* odd and that

$$\begin{aligned} x_{mn} &= \frac{16a}{\pi^2}(-1)^{(m+n+1)/2}\frac{mn}{(m^2-n^2)^2} \\ p_{mn} &= \frac{2i\hbar}{a}(-1)^{(m+n+1)/2}\frac{mn}{(m^2-n^2)} \end{aligned}$$

otherwise. Note that, for purposes of translating the general integrals above to the circumstances of this exercise, the wave functions should both be regarded as zero outside of the interval $-a \le x \le a$.

Solution: We are given that

$$\begin{aligned} \psi_n(x) &= \frac{1}{\sqrt{a}} \cos\frac{n\pi x}{2a} \qquad n = 1, 3, 5, ... \\ &= \frac{1}{\sqrt{a}} \sin\frac{n\pi x}{2a} \qquad n = 2, 4, 6, ... \end{aligned}$$

and that

$$x_{mn} = \int_{-\infty}^{\infty} \psi_m(x)\, x\, \psi_n(x)\, dx$$

where the asterisk on ψ_m can be omitted because the function is real. We now examine four possible cases:

1.) m even, n even. For this case,

$$x_{mn} = \int_{-a}^{a} \left(\frac{1}{\sqrt{a}} \sin\frac{m\pi x}{2a}\right) x \left(\frac{1}{\sqrt{a}} \sin\frac{n\pi x}{2a}\right) dx$$

We quickly conclude that the integral is zero because we notice that the integrand is an odd function of x and the limits are symmetric. This yields the result

$$x_{mn} = 0 \qquad \text{when} \qquad m \text{ and } n \text{ both even}$$

2.) m odd, n odd. For this case,

$$x_{mn} = \int_{-a}^{a} \left(\frac{1}{\sqrt{a}} \cos\frac{m\pi x}{2a}\right) x \left(\frac{1}{\sqrt{a}} \cos\frac{n\pi x}{a}\right) dx$$

We again quickly conclude that the integral is zero because the integrand is an odd function of x and the limits are symmetric. This yields the same result

$$x_{mn} = 0 \qquad \text{when} \qquad m \text{ and } n \text{ both odd}$$

3.) m even, n odd. For this case, the integral reduces to

$$\begin{aligned} x_{mn} &= \int_{-a}^{a} \left(\frac{1}{\sqrt{a}} \sin\frac{m\pi x}{2a}\right) x \left(\frac{1}{\sqrt{a}} \cos\frac{n\pi x}{a}\right) dx \\ &= \frac{1}{a}\frac{4a^2}{\pi^2} \int_{-\pi/2}^{\pi/2} y \, \sin my \cos ny \, dy \qquad (y = \pi x/2a) \\ &= \frac{8a}{\pi^2} \int_{0}^{\pi/2} y \sin my \cos ny \, dy \end{aligned}$$

where we can make the last reduction because we have an even integrand with symmetric limits. Now, we go to the integral tables or use MAXIMA to find the integral (without the premultiplier $8a/\pi^2$) with the statements

```
(%i1) declare( m, even, n, odd )$
(%i2) xmn : integrate( y*sin(m*y)*cos(n*y), y, 0, \%pi/2 );
```

$$\text{(\%o2)} \qquad \frac{A\sin(\frac{\pi n+\pi m}{2}) + B\cos(\frac{\pi n+\pi m}{2}) + C\sin(\frac{\pi n-\pi m}{2}) + D\cos(\frac{\pi n-\pi m}{2})}{4n^4 - 8m^2n^2 + 4m^4}$$

where A, B, C, and D are involved polynomials in m and n. Note, however, that in the present case both $n+m$ and $n-m$ are odd, and $\cos(\pi(n \pm m)/2) = 0$, a fact we can impose with the statement[2]

```
(%i3) xmn : ev( xmn, cos((%pi*n+%pi*m)/2)=0, cos((%pi*n-%pi*m)/2)=0 );
```

$$\text{(\%o3)} \qquad \frac{(2n^2 - 4mn + 2m^2)\sin(\frac{\pi n+\pi m}{2}) + (-2n^2 - 4mn - 2m^2)\sin(\frac{\pi n-\pi m}{2})}{4n^4 - 8m^2n^2 + 4m^4}$$

where we have now filled in the values of A and C. The remaining sine functions are a bit more complicated but can be simplified a bit if we can invoke the trigonometric addition formulae with `trigexpand`. Unfortunately, the form of the arguments in the sine function results in `trigexpand` not recognizing the sums in those arguments. The replacements effected by the statement

```
(%i4) xmn : ev( xmn, (%pi*n+%pi*m)/2 = %pi*n/2+%pi*m/2,
                     (%pi*n-%pi*m)/2 = %pi*n/2-%pi*m/2 )$
```

however, overcome that inability of `trigexpand`, which then yields the result

```
(%i5) xmn : trigexpand( xmn );
```

$$\text{(\%o5)} \qquad \frac{(-1)^{m/2}(2n^2 - 4mn + 2m^2)\sin(\pi n/2) + (-1)^{m/2}(-2n^2 - 4mn - 2m^2)\sin(\pi n/2)}{4n^4 - 8m^2n^2 + 4m^4}$$

Further simplification emerges in response to the statement

```
(%i6) xmn : factor(xmn);
```

$$\text{(\%o6)} \qquad -\frac{2m(-1)^{m/2}\, n\sin(\pi n/2)}{(n-m)^2(n+m)^2}$$

Since n is by definition odd, $\sin(\pi n/2)$ alternates between $+1$ and -1 as in the table created with the MAXIMA statement

[2] One might think that the declaration at `(%i1)` would provide MAXIMA with the wherewithall to recognize this fact, but that doesn't seem to be the case..

```
(%i7) sin([1,3,5,7,9]*%pi/2);
(%o7)      [1, −1, 1, −1, 1[
```

In other words, $\sin(\pi n/2) = (-1)^{(n-1)/2}$, a fact MAXIMA doesn't seem to recognize. We can, however, impose that fact with the statement

```
(%i8) xmn : (8*a/%pi^2)*ev( xmn, sin(%pi*n/2)= (-1)^((n-1)/2) );
```

$$\text{(\%o8)} \qquad -\frac{16amn\,(-1)^{\dfrac{n-1}{2}+\dfrac{m}{2}}}{\pi^2(n-m)^2(n+m)^2}$$

where we have also inserted the premultiplying factor previously suppressed.

This is perhaps as far as we should try to go with the assistance of MAXIMA commands. Note, however, that a still simpler expression emerges if we (a) combine the premultiplying minus sign with the other factors of -1 and (b) recognize in the denominator that $(n-m)^2(n+m)^2 = (n^2-m^2)^2$, we find finally in the case of m even and n odd that

$$x_{mn} = \frac{16a}{\pi^2}(-1)^{(n+m+1)/2}\frac{mn}{(n^2-m^2)^2} \tag{E13.11.1}$$

Note that, in this case, $n+m+1$ is *even*, so the factor $(-1)^{(n+m+1)/2}$ is real—simply ± 1.

4.) m odd, n even. The matrix element reduces to

$$\begin{aligned} x_{mn} &= \int_{-a}^{a}\left(\frac{1}{\sqrt{a}}\cos\frac{m\pi x}{2a}\right)x\left(\frac{1}{\sqrt{a}}\sin\frac{n\pi x}{a}\right)dx \\ &= \frac{1}{a}\frac{4a^2}{\pi^2}\int_{-\pi/2}^{\pi/2} y\,\cos my\sin ny\,dy \qquad (y=\pi x/2a) \\ &= \frac{8a}{\pi^2}\int_0^{\pi/2} y\cos my\sin ny\,dy \end{aligned}$$

where we can make the last reduction because we have an even integrand with symmetric limits. We can evaluate this integral in the same way that we evaluated the reverse case. Note, however, that the integral of present interest is simply obtained from the previous case by exchanging the roles of m and n. Since the value obtained at Eq. (E13.11.1) is invariant to that exchange, we conclude that that result applies also to the present case.

Now we shift to the matrix elements of momentum, defined by

$$p_{mn} = \int_{-a}^{a}\psi_m(x)\,\frac{\hbar}{i}\frac{d}{dx}\psi_n(x)\,dx$$

where again the asterisk on the first ψ can be omitted because the function is real. We now examine four possible cases:

1.) m even, n even. The integral defining p_{mn} reduces to

$$p_{mn} = \int_{-a}^{a}\left(\frac{1}{\sqrt{a}}\sin\frac{m\pi x}{a}\right)\frac{\hbar}{i}\frac{d}{dx}\left(\frac{1}{\sqrt{a}}\sin\frac{n\pi x}{a}\right)dx = \frac{\hbar n\pi}{ia^2}\int_{-a}^{a}\left(\sin\frac{m\pi x}{a}\right)\left(\cos\frac{n\pi x}{a}\right)dx$$

We quickly conclude that the integral is zero because the integral is a product of an even and an odd function of x and so the integrand of this integral is an odd function of x. Since the limits are symmetric, the integral is zero and we conclude that

$$p_{mn} = 0 \qquad \text{when} \qquad m \text{ and } n \text{ both even}$$

2.) m odd, n odd. The situation here is similar to that in the previous case, and the integral reduces to

$$p_{mn} = \int_{-a}^{a} \left(\frac{1}{\sqrt{a}}\cos\frac{m\pi x}{a}\right) \frac{\hbar}{i}\frac{d}{dx}\left(\frac{1}{\sqrt{a}}\cos\frac{n\pi x}{a}\right) dx = \frac{-\hbar n\pi}{ia^2}\int_{-a}^{a}\left(\cos\frac{m\pi x}{a}\right)\left(\sin\frac{n\pi x}{a}\right) dx$$

The ultimate integrand of the integral we need to evaluate is again an odd function of x. Since the limits are symmetric, the integral is zero, and we conclude that

$$p_{mn} = 0 \qquad \text{when} \qquad m \text{ and } n \text{ both odd}$$

3.) m even, n odd. In this case,

$$\begin{aligned} p_{mn} &= \frac{\hbar}{ia}\int_{-a}^{a}\left(\sin\frac{m\pi x}{a}\right)\frac{d}{dx}\left(\cos\frac{n\pi x}{a}\right) dx \\ &= -\frac{\hbar}{ia}\frac{n\pi}{2a}\int_{-a}^{a}\sin\frac{m\pi x}{2a}\sin\frac{n\pi x}{2a}\,dx \\ &= -\frac{\hbar n}{ia}\int_{-\pi/2}^{\pi/2}\sin my\sin ny\,dy \qquad (y=\pi x/2a) \\ &= -\frac{2\hbar n}{ia}\int_{0}^{\pi/2}\sin my\sin ny\,dy \end{aligned}$$

where we can make the last reduction because we have an even integrand with symmetric limits. At this point, we go to the integral tables or to MAXIMA. To avoid inadvertent bleeding of earlier definitions into the present calculation, we restart MAXIMA. We begin with the statements

```
(%i1) declare( m, even, n, odd )$
(%i2) pmn : integrate( sin(m*y)*sin(n*y), y, 0, %pi/2 );
```

$$\text{(\%o2)} \qquad -\frac{(n-m)\sin(\frac{\pi n+\pi m}{2})+(-n-m)\sin(\frac{\pi n-\pi m}{2})}{2n^2-2m^2}$$

Using the same adjustments as before (and reinstating the previously ignored premultiplying factor), we execute the statements

```
(%i4) pmn : ev( pmn, (%pi*n+%pi*m)/2 = %pi*n/2+%pi*m/2,
                     (%pi*n-%pi*m)/2 = %pi*n/2-%pi*m/2 )$ \\
(%i5) pmn : factor( trigexpand( pmn ) )$ \\
(%i6) pmn : -(2*hbar*n/(%i*a))*ev( pmn, sin(%pi*n/2)= (-1)^((n-1)/2) );
```

$$\text{(\%o6)} \qquad \frac{2i\hbar mn\,(-1)^{\frac{n-1}{2}+\frac{m}{2}}}{a(n-m)(n+m)}$$

Again forsaking MAXIMA, we would combine the two factors in the denominator and rearrange the terms to obtain the final result that

$$p_{mn} = \frac{2i\hbar}{a}(-1)^{(n+m-1)/2}\frac{mn}{n^2-m^2} = \frac{2i\hbar}{a}(-1)^{(n+m+1)/2}\frac{mn}{m^2-n^2} \qquad \text{(E13.11.2)}$$

Note, as before, that, $n+m+1$ is *even*, so the factor $(-1)^{(n+m+1)/2}$ is real—simply ± 1.

4.) m odd, n even. In this case,

$$p_{mn} = \frac{\hbar}{ia}\int_{-a}^{a}\left(\cos\frac{m\pi x}{a}\right)\frac{d}{dx}\left(\sin\frac{n\pi x}{a}\right) dx$$

$$
\begin{aligned}
&= \frac{\hbar}{ia}\frac{n\pi}{2a}\int_{-a}^{a}\cos\frac{m\pi x}{2a}\cos\frac{n\pi x}{2a}\,dx \\
&= \frac{\hbar n}{ia}\int_{-\pi/2}^{\pi/2}\cos my\cos ny\,dy \qquad (y=\pi x/2a) \\
&= \frac{2\hbar n}{ia}\int_{0}^{\pi/2}\cos my\cos ny\,dy
\end{aligned}
$$

where we can make the last reduction because we have an even integrand with symmetric limits. To reverse the declarations on m and n, we execute the statements

```
(%i7) remove( all, odd, all, even )$
(%i8) declare( m, odd, n, even )$
```

and then evaluate the integral with the statements

```
(%i9) pmn : integrate( cos(m*y)*cos(n*y), y, 0, %pi/2 );
```

$$\text{(\%o9)} \qquad \frac{(n-m)\sin(\frac{\pi n+\pi m}{2})+(n+m)\sin(\frac{\pi n-\pi m}{2})}{2n^2-2m^2}$$

```
(%i10) pmn : ev( pmn, (%pi*n+%pi*m)/2 = %pi*n/2+%pi*m/2,
                      (%pi*n-%pi*m)/2 = %pi*n/2-%pi*m/2 )$
(%i11) pmn : factor( trigexpand( pmn ) )$ \\
(%i12) pmn : (2*hbar*n/(%i*a))*ev( pmn, sin(%pi*m/2)= (-1)^((m-1)/2) );
```

$$\text{(\%o12)} \qquad \frac{2i\hbar mn\,(-1)^{\frac{n}{2}+\frac{m-1}{2}}}{a(n-m)(n+m)}$$

By following these steps we find that in the case of m odd and n even, just as in the case of m even and n odd,

$$p_{mn}=\frac{2i\hbar}{a}(-1)^{(n+m-1)/2}\frac{mn}{n^2-m^2}=\frac{2i\hbar}{a}(-1)^{(n+m+1)/2}\frac{mn}{m^2-n^2}$$

Note, as before, that, $n+m+1$ is *even*, so the factor $(-1)^{(n+m+1)/2}$ is real—simply ± 1.

13.12 Fourier Coefficients for Sawtooth (MAXIMA)

Exercise: The sawtooth wave is defined by

$$f(x) = \frac{x}{l} \quad ; \quad -l \le x \le l$$

Use symbolic integration to find the Fourier coefficients a_n and b_n for this function and then generate graphs showing the function given by the truncated series

$$f_{\text{trunc}}(x) = \frac{a_0}{2} + \sum_{n=1}^{N} \left(a_n \cos\frac{n\pi x}{l} + b_n \sin\frac{n\pi x}{l} \right)$$

for various values of N, including 0, 1, 2, 3, and 10.

Solution: For the sawtooth wave defined by

$$f(x) = \frac{x}{l} \qquad ; \qquad -l \le x \le l$$

the Fourier coefficients, a_n and b_n, are given by

$$a_n = \frac{1}{l}\int_{-l}^{l} f(x)\cos\frac{n\pi x}{l}dx \quad ; \quad b_n = \frac{1}{l}\int_{-l}^{l} f(x)\sin\frac{n\pi x}{l}dx$$

In terms of the variable $\overline{x} = x/l$, these integrals become

$$a_n = \int_{-1}^{1} \overline{x}\cos(n\pi\overline{x})\,d\overline{x} \quad ; \quad b_n = \int_{-1}^{1} f(x)\sin(n\pi\overline{x}\,d\overline{x}$$

Dropping the overbars, we find these coefficients in MAXIMA using the statements

```
(%i1) $
(%i2) a[n] := integrate( x*cos(n*%pi*x), x, -1, 1 )$
(%i3) b[n] := integrate( x*sin(n*%pi*x), x, -1, 1 )$
```

(Remember to use `%pi` not `pi` so MAXIMA will know you mean the value as well as the symbol.)

Our remaining task is to evaluate these coefficients for specific terms and generate plots for the truncated function

$$f_{\text{trunc}}(x) = \frac{a_0}{2} + \sum_{n=1}^{N} \left(a_n \cos\frac{n\pi x}{l} + b_n \sin\frac{n\pi x}{l} \right) = \frac{a_0}{2} + \sum_{n=1}^{N} \left(a_n \cos(n\pi\overline{x}) + b_n \sin(n\pi\overline{x}) \right)$$

for $N = 1, 2, 3$ and 10. We evaluate the sum for these four cases and create the plots with the statements

```
(%i4) ftrunc1 : a[0]/2 + sum( a[n]*cos(n*%pi*x)+b[n]*sin(n*%pi*x), n, 0, 1 )$
(%i5) plot2d(ftrunc1, [x, -2.5, 2.5], [y, -1.2,1.2], [color, black],
       [style, [lines, 4] ], [legend, false], [xlabel, "x"], [ylabel, "ftrunc1"] )$
(%i6) ftrunc2 : a[0]/2 + sum( a[n]*cos(n*%pi*x)+b[n]*sin(n*%pi*x), n, 0, 2 )$
(%i7) plot2d(ftrunc2, [x, -2.5, 2.5], [y, -1.2,1.2], [color, black],
       [style, [lines, 4] ], [legend, false], [xlabel, "x"], [ylabel, "ftrunc2"] )$
(%i8) ftrunc3 : a[0]/2 + sum( a[n]*cos(n*%pi*x)+b[n]*sin(n*%pi*x), n, 0, 3 )$
(%i9) plot2d(ftrunc3, [x, -2.5, 2.5], [y, -1.2,1.2], [color, black],
       [style, [lines, 4] ], [legend, false], [xlabel, "x"], [ylabel, "ftrunc3"] )$
(%i10) ftrunc10 : a[0]/2 + sum( a[n]*cos(n*%pi*x)+b[n]*sin(n*%pi*x), n, 0, 10 )$
(%i11) plot2d(ftrunc10, [x, -2.5, 2.5], [y, -1.2,1.2], [color, black],
       [style, [lines, 4] ], [legend, false], [xlabel, "x"], [ylabel, "ftrunc10"] )$
```

The plots—shown in Figs. E13.3–E13.6—get progressively sharper and more like a sawtooth as the number of terms in the Fourier series increases.

Figure E13.3: Approximation of the sawtooth wave $f_{\text{trunc}}(1)$.

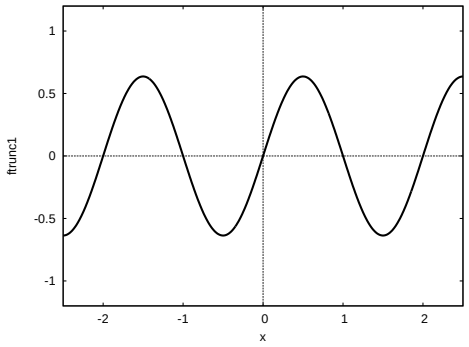


Figure E13.4: Approximation of the sawtooth wave $f_{\text{trunc}}(2)$.

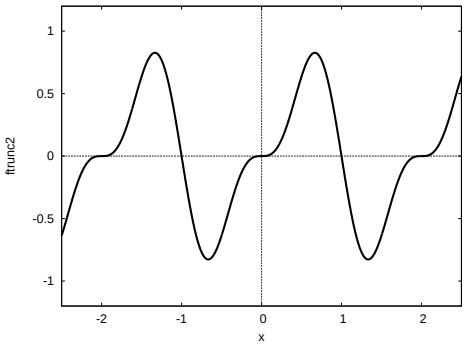


Figure E13.5: Approximation of the sawtooth wave $f_{\text{trunc}}(3)$.

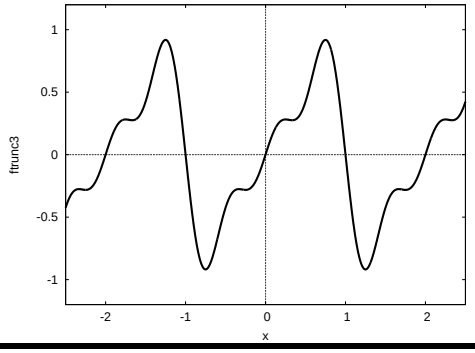
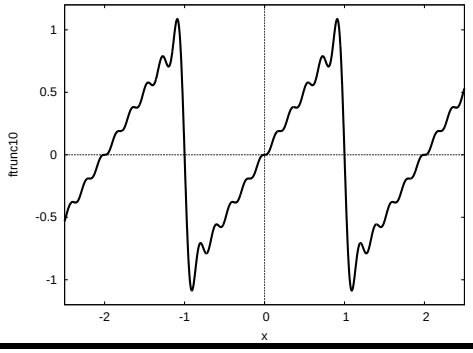


Figure E13.6: Approximation of the sawtooth wave $f_{\text{trunc}}(10)$.



13.13 Expansion in Legendre Polynomials (MAXIMA)

Exercise: The Legendre polynomials $L_n(x)$, $n = 0, 1, 2, \ldots$, are orthogonal on the interval $-1 \le x \le 1$ with weight 1. In particular,

$$\int_{-1}^{1} L_m(x)\, L_n(x)\, dx = \frac{2}{2n+1}\, \delta_{mn}$$

Any function defined over the interval $-1 \le x \le 1$ can then be expanded in the Legendre series

$$f(x) = \sum_{n=0}^{\infty} c_n\, L_n(x)$$

(a) Show by hand that the coefficient c_n in this expansion is given by

$$c_n = \frac{2n+1}{2} \int_{-1}^{1} f(x)\, L_n(x)\, dx$$

(b) Use symbolic integration to find c_n for $n = 0, 1, 2, 3, 4, 5$ and 6 in the Legendre expansion for the function

$$f(x) = \begin{cases} -1 & -1 < x < 0 \\ 1 & 0 < x < 1 \end{cases}$$

(c) Graph the functions defined by the partial sums $\sum_{n=0}^{N} c_n\, L_n(x)$ for N = 0, 1, 2, 3, 4, 5, and 6.

Hint: Quite possibly the symbolic program you are using has the Legendre polynomials built in somehow, and you should study its manuals to find out how to invoke them. Just in case that isn't true, the first nine Legendre polynomials are

$$L_0(x) = 1 \qquad L_5(x) = \tfrac{1}{8}(63x^5 - 70x^3 + 15x)$$
$$L_1(x) = x \qquad L_6(x) = \tfrac{1}{16}(231x^6 - 315x^4 + 105x^2 - 5)$$
$$L_2(x) = \tfrac{1}{2}(3x^2 - 1) \qquad L_7(x) = \tfrac{1}{16}(429x^7 - 693x^5 + 315x^3 - 35x)$$
$$L_3(x) = \tfrac{1}{2}(5x^3 - 3x) \qquad L_8(x) = \tfrac{1}{128}(6435x^8 - 12012x^6 + 6930x^4 - 1260x^2 + 35)$$
$$L_4(x) = \tfrac{1}{8}(35x^4 - 30x^2 + 3) \qquad L_9(x) = \tfrac{1}{128}(12155x^9 - 25740x^7 + 18018x^5 - 4620x^3 + 315x)$$

Solution: (a) To solve this part of the exercise, we multiply the second equation in the problem statement by $L_m(x)$ and integrate over the interval $-1 \le x \le 1$ to find that

$$\int_{-1}^{1} L_m(x)\, f(x)\, ds = \sum_n c_n \int_{-1}^{1} L_n(x)\, L_m(x)\, dx = \sum_n c_n \left(\frac{2}{2n+1}\, \delta_{mn} \right) = c_m \left(\frac{2}{2m+1} \right)$$

Here we have recognized the orthogonality of the polynomials as given in the first equation in the statement of the exercise. We conclude that

$$c_n = \frac{2n+1}{2} \int_{-1}^{1} L_n(x)\, f(x)\, dx$$

(b) The first thing we can notice about this function is that it is odd, i.e. $f(-x) = -f(x)$. Recalling the properties of even and odd functions, we know also that two odd functions multiplied together will produce an even function, while one of each will produce an odd function. This proves useful because $L_n(x)$ is even when n is even, and odd when n is odd. Therefore, $g_n(x) = f(x)\, L_n(x)$ will be odd when n is even and even when n is odd. Since odd functions integrated over symmetric limits will be zero, we only care therefore about the even functions $g_n(x) = f(x)\, L_n(x)$, when n is odd.

Likewise, even functions integrated over symmetric limits will have values double that of half of the integral, e.g.

$$\int_{-1}^{1} g_n(x)\,dx = 2\int_{0}^{1} g_n(x)\,dx$$

when $g_n(x)$ is an even function. We can use this property to solve for c_n much more quickly now with the statements

```
(%i1) load(orthopoly);
(%i2) for a from 1 thru 9 step 2 do c[a] :
      (2*a+1)*integrate(legendre_p(a,x),x,0,1); print(c[a]);
```

(%o3) *done*

```
(%i3) [c[1],c[3],c[5],c[7],c[9]];
```

(%o3) $\left[\frac{3}{2},-\frac{7}{8},\frac{11}{16},-\frac{75}{128},\frac{133}{256}\right]$

Then we construct and display the several partial sums with the statements

```
(%i4) f[0] : c[1]*legendre_p(1,x);
```

(%o4) $\frac{3\,x}{2}$

```
(%i5) for i from 1 thru 4 do f[i] : expand( f[i-1] +
        c[(i*2)+1]*legendre_p((i*2)+1,x) );
```

(%o5) *done*

```
(%i6) f[1];
```

(%o6) $\frac{45\,x}{16}-\frac{35\,x^3}{16}$

```
(%i7) f[2];
```

(%o7) $\frac{693\,x^5}{128}-\frac{525\,x^3}{64}+\frac{525\,x}{128}$

```
(%i8) f[3];
```

(%o8) $-\frac{32175\,x^7}{2048}+\frac{63063\,x^5}{2048}-\frac{40425\,x^3}{2048}+\frac{11025\,x}{2048}$

```
(%i9) f[4];
```

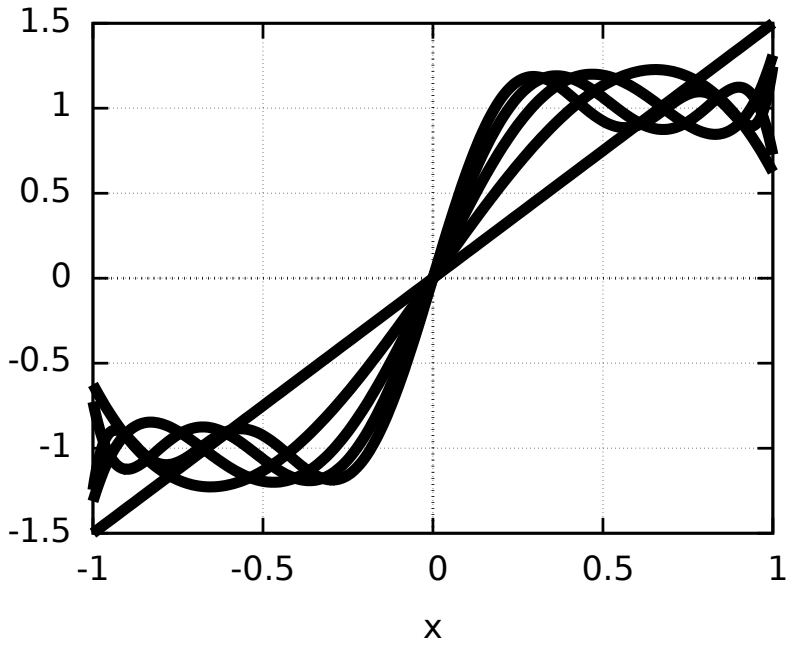
(%o9) $\frac{1616615\,x^9}{32768}-\frac{984555\,x^7}{8192}+\frac{1702701\,x^5}{16384}-\frac{315315\,x^3}{8192}+\frac{218295\,x}{32768}$

Finally, we plot the requested graph with the statement

```
(%i10) plot2d( [f[0],f[1],f[2],f[3],f[4]], [x, -1, 1], [color,black], [legend,false],
       [grid2d, true], [style,[lines,4]]);
```

The graph is shown in Fig. E13.7.

Figure E13.7: An overlay plot showing various degrees of Legendre polynomials graphed to approximate the function in part (b) and revealing the beginnings of convergence to that function.
.



13.14 Deduction of Simpson's Rule (MAXIMA)

Exercise: To deduce Simpson's rule, we start by supposing three consecutive values f_1, f_2, and f_3 of the integrand, where for simplicity in notation we take the points of evaluation to be $x_1 = x_2 - \Delta x$, x_2, and $x_3 = x_2 + \Delta x$. Using a symbol manipulating program to do the algebra and calculus, (a) find the coefficients A, B, and C needed to make the parabola $Ax^2 + Bx + C$ pass through the three points (x_i, f_i), $i = 1, 2, 3$, (b) integrate that parabola over the interval $x_1 < x < x_3$ to find that

$$\int_{x_1}^{x_3} f(x)\,dx \approx \int_{x_1}^{x_3} (Ax^2 + Bx + C)\,dx = \frac{\Delta x}{3}\Big(f_1 + 4f_2 + f_3\Big)$$

(c) show that this result actually gives the correct value for $f(x) = x^3$ and, finally, (d) deduce the (extended) Simpson's rule

$$\int_a^b f(x)\,dx \approx S_N = \tfrac{1}{3}\left(f_0 + 4f_1 + 2f_2 + 4f_3 + \cdots + 2f_{N-2} + 4f_{N-1} + f_N\right)\,\Delta x$$

Note: Because this exercise relates to numerical algorithms, it has been placed in with other exercises that are numerical. This exercise is symbolic, and you should use a symbol manipulating program for parts (a), (b), and (c); however, you should address part (d) by hand.

Solution: (a), (b) We want to begin by fitting the parabola

$$f(x) = Ax^2 + Bx + C$$

to the three points

$$f(x_2 - \Delta x) = f_1 \quad ; \quad f(x_2) = f_2 \quad ; \quad f(x_2 + \Delta x) = f_3$$

Using MAXIMA, we first define the general function $f(x)$ with the statement

```
(%i1) f(x) := A*x^2+B*x+C$
```

Then, we stipulate that the three points shall lie on the parabola by defining the three equations

```
(%i2) eq1 : f1 = f(x2-dx)$
(%i3) eq2 : f2 = f(x2)$
(%i4) eq3 : f3 = f(x2+dx)$
```

Next, we solve the three equations for the constants A, B, and C with the statement

```
(%i5) soln : solve( [eq1, eq2, eq3], [A, B, C] );
```

(%o5) qquad $\Big[\ [A = \ldots, B = \ldots, C = \ldots\ ]\ \Big]$

though we suppress the explicit solutions because of their complexity. Finally, we substitute this solution into the original function and integrate it with the statements

```
(%i6) integ : ev( f(x), soln )$
(%i7) S : integrate( integ, x, x2-dx, x2+dx )$
(%i8) S : factor( expand( S ) );
```

(%o8) $$\frac{dx\,(f_3 + 4f_2 + f_1}{3}$$

which demonstrates the correctness of the desired relationship.

(c) For the cubic polynomial $g(x) = x^3$, we use the statements

```
(%i9) g(x) := x^3$
(%i10) ev( S,  f1 = g(x2-dx), f2 = g(x2), f3 = g(x2+dx) )$
(%i11)  factor( expand(%) );
```

(%o11) $2\,dx\,x_2(x_2^2 + dx^2)$

to find $\int_{x2-dx}^{x_2+dx} g(x)\,dx$ via Simpson's rule and the statements

```
(%i12) integrate( g(x), x,  x2-dx, x2+dx );
(%i13)  factor( expand(%) );
```

(%o13) $2\,dx\,x_2(x_2^2 + dx^2)$

to evaluate the same integral formally and *exactly*. We do indeed discover that Simpson's rule, which was designed to be exactly correct for quadratic polynomials is apparently also exactly correct for the cubic function!

(d) To deduce the extended Simpson's rule, let us introduce the points $x_i = i\,\Delta x$, $i = 0, 1, 2, \ldots, N$, with $a = x_0$ and $b = x_N$. Make sure N is even, so we have an integer number of *pairs* or strips. Then, we write

$$\begin{aligned}
\int_a^b f(x)\,dx &= \sum_{i=0}^{N/2} \int_{x_{2i}}^{x_{2i+2}} f(x)\,dx \\
&= \int_{x_0}^{x_2} f(x)\,dx + \int_{x_2}^{x_4} f(x)\,dx + \cdots + \int_{x_{N-4}}^{x_{N-2}} f(x)\,dx + \int_{x_{N-2}}^{x_N} f(x)\,dx \\
&= \frac{dx}{3}\Big((f_0 + 4f_1 + f_2) + (f_2 + 4f_3 + f_4) + \cdots \\
&\qquad\qquad + (f_{N-4} + 4f_{N-3} + f_{N-2}) + (f_{N-2} + 4f_{N-1} + f_N) \Big) \\
&= \frac{dx}{3}\Big(f_0 + 4f_1 + 2f_2 + 4f_3 + 2f_4 + \cdots + 2f_{N-4} + 4f_{N-3} + 2f_{N-2} + 4f_{N-1} + f_N \Big)
\end{aligned}$$

where, in this last expression (which was obtained by substituting appropriate values into the result of part (b)), $f_i = f(x_i)$. Q.E.D.

13.20 Black Body Radiation (MAXIMA)

Exercise: Planck's black body radiation law gives the expression

$$I(\nu_2, \nu_1) = \frac{8\pi h}{c^3} \int_{\nu_1}^{\nu_2} \frac{\nu^3}{e^{h\nu/kT} - 1}\, d\nu$$

for the power radiated per unit area in the frequency range $\nu_1 \leq \nu \leq \nu_2$. Using numerical means, explore the power radiated in the visible spectrum 4×10^{14} Hz $\leq \nu \leq 7 \times 10^{14}$ Hz as a function of temperature. *Hint*: One way to approach this exercise would be to choose a reference frequency ν_0 arbitrarily (say 10^{14} Hz) and recast the integral on the dimensionless variable $s = \nu/\nu_0$. Examination of I in units of $8\pi h\nu_0^4/c^3$ as a function of T in units of $h\nu_0/k$ would then be indicated.

Solution: Planck's black body radiation law gives us the expression

$$I = \frac{8\pi h}{c^3} \int_{\nu_1}^{\nu_2} \frac{\nu^3}{e^{hv/kT} - 1}\, d\nu$$

for the power radiated per unit area over the frequency range $\nu_1 \leq \nu \leq \nu_2$. Introducing the dimensionless variable $s = \nu/\nu_0$ (where ν_0 is an arbitrarily selected reference frequency) and noting that $ds = d\nu/\nu_0$, we can recast this equation as

$$I = \frac{8\pi h\nu_0^4}{c^3} \int_{\nu_1/\nu_0}^{\nu_2/\nu_0} \frac{s^3}{e^{hs\nu_0/kT} - 1}\, ds$$

If we express I in units of $I_0 = 8\pi h\nu_0^4/c^3$ as a function of T in units of $T_0 = h\nu_0/k$, then we have

$$I' = \int_{\nu_1/\nu_0}^{\nu_2/\nu_0} \frac{s^3}{e^{s/T'} - 1}\, ds$$

where $I' = I/(8\pi h\nu_0^4/c^3)$ and $T' = T/(h\nu_0/k)$. To explore this integral over the frequency range 4×10^{14} Hz $\leq \nu \leq 7\times10^{14}$ Hz, let us choose the reference frequency to be 10^{14} Hz so that the integral of interest ranges from 4 to 7. Note also that this choice of ν_0 implies the reference temperature

$$T_0 = \frac{h\nu_0}{k} = \frac{6.62 \times 10^{-34}\ \text{Js} \times 10^{14}\ \text{s}}{1.38 \times 10^{-23}\ \text{J/K}} = 4797\ \text{K}$$

Since the temperature of a white-hot tungsten filament is about 2800 K and the average temperature of the sun is 5000-6000 K, the region of interest for this exercise probably is in the region of $T/T_0 \leq 1$.

Adopting an optimistic perspective, we first attempt to evaluate the integral analytically by invoking the statement

```
(%i1) iii : integrate( s^3/(exp(s/t) - 1), s, 4, 7 )$
```

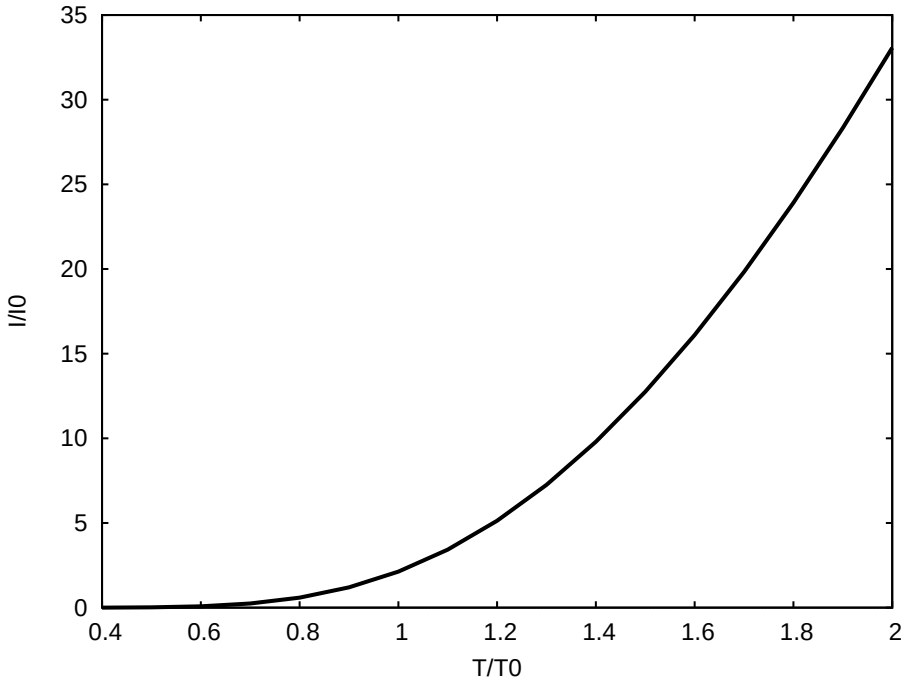
Somewhat surprisingly, MAXIMA indeed returns what it claims to be an analytic evaluation, though the expression is a bit of a mess and, in addition to logarithms, exponentials, and powers, involves the polylogarithm functions Li_4 and Li_3 defined by

$$\text{Li}_s(z) = \sum_{k=1}^{\infty} \frac{z^k}{k^s}$$

While MAXIMA can evaluate Li_s for $1 \leq s \leq 3$, it evidently cannot evaluate Li_4, so we must resort to numerical integration to solve this exercise. We start by defining the integrand with the statement

```
(%i2) int : s^3/(exp(s/t) - 1)$
```

Figure E13.8: Planck's black body radiation intensity $I' = I/I_0$ as a function of temperature $T' = T/T_0$.



Then, having explored a variety of intervals, we create a list with values of t between 0.4 and 2.0 with the statements[3]

```
(%i3) for i from 4 thru 20 do t[i] : i/10$
(%i4) t1 : makelist( t[i], i, 4, 20 );
```

(%o4) $\left[\frac{2}{5}, \frac{1}{2}.\frac{3}{5}, \ldots, \frac{9}{5}, \frac{19}{10}, 2\right]$

Finally, we evaluate the integral at these several values of t with the statements

```
(%i5) for i from 1 thru 17 do I[i] : quad_qag( ev(int, t = t1[i]), s, 4.0, 7.0, 2)$
(%i6) I1 : makelist( I[i], i, 1, 17 );
```

(%o6) $[0.001583541109973682, 0.01571688219001995 \ldots,$
$28.33524173120366, 33.0655671937392]$

The statement

```
(%i7) plot2d( [discrete, t1, I1], [color, black], [style, [lines, 4] ],
            [legend, false], [xlabel, "T/T0"], [ylabel, "I/I0"] );
```

The result is shown in Fig. E13.8.

[3]Here, after a bit of experimentation, we learn that the evaluation of the function, which contains terms like $e^{4/t}$ and $e^{7/t}$ causes MAXIMA major indigestion for values of t too close to zero. Thus, we ignore values of t less than 0.4 in the plotting.

13.21 Confidence Intervals for Gaussian (MAXIMA)

Exercise: As used in statistical data analysis, the Gaussian distribution for a variable t is usually expressed in terms of the standard deviation σ, the distribution function being

$$\frac{1}{\sqrt{2\pi}\,\sigma} e^{-t^2/2\sigma^2}$$

Thus, the probability of finding a value between a and b is given by

$$P(a,b) = \frac{1}{\sqrt{2\pi}\sigma} \int_a^b e^{-t^2/(2\sigma^2)}\, dt$$

Show analytically that $P(-x,x) = \text{erf}(x/(\sqrt{2}\sigma))$, and then evaluate $P(-\sigma,\sigma)$, $P(-2\sigma,2\sigma)$, and $P(-3\sigma,3\sigma)$ numerically. The values of these three quantities are 0.6827, 0.9545, and 0.9973, respectively—values that give rise to the designations of 68%, 95%, and 99% confidence intervals in statistical data analysis.

Solution: The probability that a single measurement drawn from a normally distributed universe of possible values will lie between a and b is given by the integral

$$P(a,b) = \frac{1}{\sqrt{2\pi}\sigma} \int_a^b e^{-t^2/(2\sigma^2)}\, dt$$

To show analytically that $P(-x,x) = \text{erf}(x/(\sqrt{2}\sigma))$, we simply a change variables. If we let $t^2/2\sigma^2 = s^2$, then $dt = \sqrt{2}\,\sigma\, ds$ and, when $t = x$, $s = x/\sqrt{2}\,\sigma$. Thus, the integral becomes

$$P(-x,x) = \frac{1}{\sqrt{\pi}} \int_{-x/\sqrt{2}\sigma}^{x/\sqrt{2}\sigma} e^{-s^2}\, ds \;=\; \frac{2}{\sqrt{\pi}} \int_0^{x/\sqrt{2}\sigma} e^{-s^2}\, ds \;=\; \text{erf}\left(\frac{x}{\sqrt{2}\sigma}\right)$$

The first equality is a result of the change of variables. The second stems from the observations that (1) the integrand is an even function of the integration variable and (2) the limits are symmetric.

Another way to determine the relationship between $P(-x,x)$ and $\text{erf}(x)$ is to integrate the probability function symbolically. We can define the integrand and perform the integral in MAXIMA by entering the statements

```
(%i1) assume(\sigma>0)$
(%i2) int : exp(-t^2/(2*\sigma^2))$
(%i3) coef : 1/(sqrt(2*%pi)*\sigma)$
(%i4) val : coef * integrate( int, t, -x, x );
```

(%o4) $\text{erf}\left(\dfrac{x}{\sqrt{2}\,\sigma}\right)$

This sequence of statements to MAXIMA confirms the demonstration of the previous paragraph that $P(-x,x) = \text{erf}(x/(\sqrt{2}\sigma))$.

To evaluate the probability function for particular values of x, it is only necessary to evaluate the error function for $x/(\sqrt{2}\sigma)$. For $x = \sigma$, 2σ, and 3σ, respectively, we seek the value of $\text{erf}(1/\sqrt{2})$, $\text{erf}(2/\sqrt{2})$, and $\text{erf}(3/\sqrt{2})$. Appropriate statements are

```
(%i5) float( ev(val, x=\sigma) );
(%o5)     0.6826894921370859
(%i6) float( ev(val, x=2*\sigma) );
(%o6)     0.9544997361036416
(%i7) float( ev(val, x=3*\sigma) );
(%o7)     0.9973002039367398
```

We conclude that a single value randomly selected from a Gaussian distribution stands a 68.3% chance of falling within one standard deviation of the mean, a 95.4% chance of falling within two standard deviations, and a 99.7% chance of falling within three standard deviations.

13.23 Properties of Lorentz Distribution (MAXIMA)

Exercise: The normalized Lorentz distribution function is given by

$$p(x) = \frac{1}{\pi} \frac{a/2}{x^2 + (a/2)^2}$$

Find the probability that a single, randomly selected value will be in the range $-a \leq x \leq a$. Make sure to assess the precision of your result by methods that do *not* exploit *a priori* knowledge of the exact value. *Hint*: Before evaluating the integral, introduce the dimensionless variable $s = x/a$ and note that the result actually doesn't depend on a, so there is but *one* number to determine.

Solution: The normalized Lorentz distribution function is

$$p(x) = \frac{1}{\pi} \frac{a/2}{x^2 + (a/2)^2}$$

To find the probability that a random value will be in the range $a \leq x \leq a$, we simply integrate $p(x)dx$ from a to $-a$. First, however, let us subsitute the dimensionless variable $s = x/a$. Then the integral we wish to evaluate becomes

$$\int_{-a}^{a} p(x)\, dx = \frac{1}{2\pi} \int_{-1}^{1} \frac{1}{s^2 + (1/4)} ds$$

We explored this integral symbolically in Exercise 13.5. Here, we wish to exhibit MAXIMA's capabilities to evaluate integrals numerically. To do so, we invoke the statement

```
(%i1) p : (1/(2*%pi))/(s^2+1/4);
```

(%o1) $$\frac{1}{2\pi(s^2 + 1/4)}$$

to define the integrand. Then, we evaluate the integral directly by a numerical method with the statement

```
(%i2) quad_qag( p, s, -1.0, 1.0, 2 );
```

(%o2) $[0.7048327646991336, 6.387177245437137 \times 10^{-12}, 63, 0]$

The probability that a value will fall in that range $-a \leq x \leq a$ is evidently 70.5%. We learn also that the result has evidently been determined in 63 evaluations of the integrand to an absolute precision of about 6×10^{-12}, so we can believe the first 12 digits or so of the displayed value.

Chapter 14

Finding Roots

14.3 Finding Extrema in a Cubic Polynomial (MAXIMA)

Exercise: Find the points at which the function $f(x) = ax^3 + bx^2 + cx + d$ has (local) extrema and find a criterion involving the coefficients (or some of them) that will assure that the function has three real roots. Assume that a, b, c, and d are real.

Solution: We seek the extreme points of $f(x) = ax^3 + bx^2 + cx + d$. We enter this equation into MAXIMA with the statement

```
(%i1) eq : a*x^3 + b*x^2 + c*x + d;
```

Then we find the roots of the first derivative with the statements

```
(%i2) deq : diff( eq, x )$
(%i3) soln : solve( deq = 0, x );
```

(%o3) $$\left[x = -\frac{\sqrt{b^2 - 3ac} + b}{3a}, x = \frac{\sqrt{b^2 - 3ac} - b}{3a}\right]$$

```
(%i4) soln1 : part( soln, 1, 2 );
```

(%o4) $$-\frac{\sqrt{b^2 - 3ac} + b}{3a}$$

```
(%i5) soln2 : part( soln, 2, 2 );
```

(%o5) $$\frac{\sqrt{b^2 - 3ac} - b}{3a}$$

These are the values of x at which the polynomial `eq` has a (local) maximum or a (local) minimum. The values of `eq` at these points are given by

```
(%i6) ext1 : ev( eq, x=soln1 );
```

(%o6) $$d - \frac{(\sqrt{b^2 - 3ac} + b)^3}{27a^2} + \frac{b(\sqrt{b^2 - 3ac} + b)^2}{9a^2} - \frac{c(\sqrt{b^2 - 3ac} + b)}{3a}$$

```
(%i7) ext2 : ev( eq, x=soln2 );
```

(%o7) $$d + \frac{(\sqrt{b^2 - 3ac} - b)^3}{27a^2} + \frac{b(\sqrt{b^2 - 3ac} - b)^2}{9a^2} + \frac{c(\sqrt{b^2 - 3ac} - b)}{3a}$$

We know that if all three roots of `eq` are real, then the two extreme points must also be real. This means that the quantity under the square root, $b^2 - 3ca$, in both solutions must be positive. We also know that if all three roots are real, then the product of the two extreme points ought to be negative, i.e., the value of `eq` at the maximum extreme point and its value at the minimum extreme point must have opposite signs. To see what restrictions this additional criterion puts on a, b, c and d, we begin by evaluating the critical product with the statements

```
(%i8) prod : expand( ext1 * ext2 );
```

$$\text{(\%o8)} \qquad d^2 - \frac{2bcd}{3a} + \frac{4b^2d}{27a^2} + \frac{4c^3}{27a} - \frac{b^2c^2}{27a^2}$$

Since both 27 and a^2 are positive, we can multiply this result by $27a^2$ without affecting its sign. Thus, we must require that the quantity given by

```
(%i9) expand( 27*a^2*prod );
```

$$\text{(\%o9)} \qquad 27a^2d^2 - 18abcd + 4b^3d + 4ac^3 - b^2c^2$$

be less than zero. In summary, we conclude that

- If $b^2 - 3ca < 0$, the cubic polynomial has only one real root.
- If $b^2 - 3ca > 0$, the cubic polynomial *may* have three real roots, but only if, in addition

$$27a^2d^2 - 18abcd + 4b^3d + 4ac^3 - b^2c^2 < 0$$

14.4 Roots of Quadratic Equation (MAXIMA)

Exercise: In some quantum calculations, the need to solve the equation $x(x+1) = l(l+1)$ for x arises. Find those roots, noting particularly that, since the equation is *quadratic*, there are *two* roots. The obvious root $x = l$ is *not* the only one.

Solution: We wish to find the roots to the equation $x(x+1) = l(l+1)$ or, equivalently, to solve $x(x+1) - l(l+1) = 0$ for x. To do this symbolically in *MAXIMA* is a straight forward calculation. We simply define the expression and request its roots with the statements

```
(%i1) qm :  x*(x+1) - l*(l+1)$
(%i2) soln :  solve( qm=0 , x);
(%o2)     [x = -l - 1, x = l]
```

We see that there are two roots, which we expect from a quadratic equation. To verify these solutions we substitute them into the original equation with the statements

```
(%i3) expand( ev( qm, soln[1] ) );
(%o3)     0
(%i4) expand( ev( qm, soln[2] ) );
(%o4)     0
```

14.5 Lawnmower Blade Overlap (MAXIMA)

Exercise: Each of the three blades of a lawn mower has radius a. As shown in Fig. E14.1, the center blade is invariably mounted somewhat in front of the two outside ones so that the areas cut by each blade can overlap without risking collision of the blades with one another. What must be the minimum offset x of the center of the middle blade from the line joining the centers of the two outer blades so that their cutting paths will overlap by an amount y without collision of the blades?

Solution: We need to find the distance x by which the center blade of a lawnmower must be offset so that the cutting areas overlap by an amount y without collision of the blades. By looking at the geometry of the configuration, we see that we have a right triangle with a height x, a base $a+(a-y)$, and a hypotenuse of at least $2a$ if the blades are not to touch. The Pythagorean Theorem gives us the equation $x^2 + (2a - y)^2 = (2a)^2$, which we can solve for x in MAXIMA by using the statements

```
(%i1) pyth : x^2 + (2*a - y)^2 - 4*a^2 $
(%i2) solve( pyth = 0, x);
```

(%o2) $\left[x = -\sqrt{4ay - y^2}, x = \sqrt{4ay - y^2}\right]$

Since x is a distance from an axis, we take the positive root. The center blade should be placed a distance $\sqrt{4ay - y^2}$ from the axis for given a and y. To look at a graph of x as a function of y, we can set a to the unit value $a = 1$, which—in effect—declares that we will measure both y and the solution in units of a. In this case, y can be in the range $0 \le y \le 1$ since the value $y = a$ is the point at which the two side blades meet behind the center blade. To produce the plot, we extract the positive root, set a equal to one, and plot with the statements

```
(%i3) part(%, 2, 2)$
```

(%o3) $\sqrt{4ay - y^2}$

```
(%i4) p : subst(1, a, %)$
```

(%o4) $\sqrt{4y - y^2}$

```
(%i5) plot2d(p, [y, 0.0, 1.0], [xlabel, "y"], [ylabel, "x"],
        [style, [lines, 3]], [color, black] );
```

The resulting figure is shown in Fig. E14.2.

Figure E14.1: The three-bladed lawnmower described in Exercise 14.0.

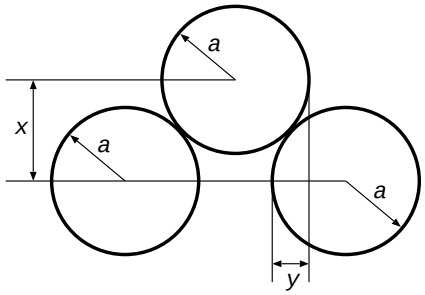
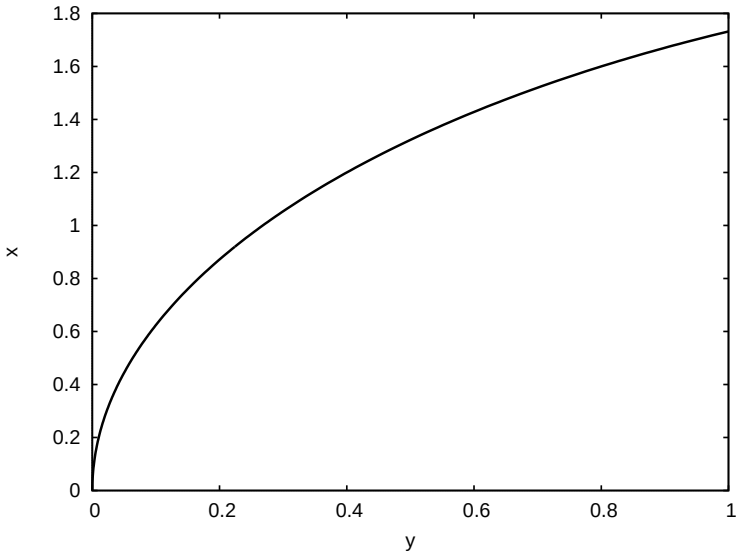


Figure E14.2: X as a function of Y.



14.8 Three Coupled Oscillators (MAXIMA)

Exercise: Find the natural frequencies for the *three* modes of oscillation characterizing the system that results when the system shown in Fig. E14.3 is extended to contain three objects coupled in a line. Take the four springs all to have the same spring constant but allow for the possibility that the middle object may have a mass m' different from the mass m of the two outside objects. In particular, measure frequencies in units of $\sqrt{k/m}$ and seek a graph showing the frequency of each of the modes as a function of $\beta = m'/m$. *Hint*: To help you get started and to facilitate focusing on the solution of the ODEs rather than on deriving them, note that, for three masses, the equations of motion will be

$$\begin{aligned} m\frac{d^2x_1}{dt^2} &= -kx_1 + k(x_2 - x_1) \\ m'\frac{d^2x_2}{dt^2} &= -k(x_2 - x_1) + k(x_3 - x_2) \\ m\frac{d^2x_3}{dt^2} &= -k(x_3 - x_2) - kx_3 \end{aligned}$$

Solution: Given the equations

$$m\frac{d^2x_1}{dt^2} = -kx_1 + k(x_2 - x_1); \quad m'\frac{d^2x_2}{dt^2} = -k(x_2 - x_1) + k(x_3 - x_2); \quad m\frac{d^2x_3}{dt^2} = -k(x_3 - x_2) - kx_3$$

for the motion of three objects connected by four springs, the substitutions $\omega_0 = \sqrt{k/m}$, $\beta = m'/m$, and $\bar{t} = \omega_0 t$ causes the equations to become

$$\frac{d^2x_1}{d\bar{t}^2} = -2x_1 + x_2; \qquad \beta\frac{d^2x_1}{dt^2} = x_1 - 2x_2 + x_3; \qquad \frac{d^2x_3}{d\bar{t}^2} = x_2 - 2x_3$$

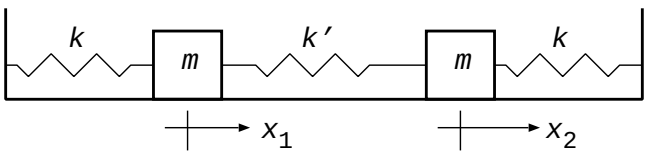
We enter these into MAXIMA using the statements

```
(%i1) equ1 : 'diff(x1(t),t,2) = -2*x1(t) + x2(t)$
(%i2) equ2 : \beta*'diff(x2(t),t,2) = x1(t) - 2*x2(t) + x3(t)$
(%i3) equ3 : 'diff(x3(t),t,2) = x2(t) - 2*x3(t)$
```

We then assume solutions of the form $x_1 = x_{10}\cos\omega t$ et al., and substitute these into the equations to find the natural frequencies. To insure non-trivial solutions, we must require the determinant of the coefficient matrix to be zero. To extract that determinant and find its roots, we invoke the statements

```
(%i4) x1(t) := x10*cos(\omega*t)$
(%i5) x2(t) := x20*cos(\omega*t)$
```

Figure E14.3: A system of two masses and three springs. The surface on which the two objects rest is assumed to be frictionless.



```
(%i6) x3(t) := x30*cos(\omega*t)$
```

```
(%i7) eq1 : ev(equ1, diff );
```
$$(\%o7) \qquad -\omega^2 \cos(\omega t)\, x_{10} = \cos(\omega t)\, x_{20} - 2\cos(\omega t)\, x_{10}$$
```
(%i8) eq1 : expand( eq1/cos(\omega*t) );
```
$$(\%o8) \qquad -\omega^2\, x_{10} = x_{20} - 2\, x_{10}$$
```
(%i9) eq2 : expand( ev(equ2, diff )/cos(\omega*t) );
```
$$(\%o9) \qquad -\beta\,\omega^2\, x_{20} = x_{20} - 2\, x_{20} + x_{10}$$
```
(%i10) eq3 : expand( ev(equ3, diff )/cos(\omega*t) );
```
$$(\%o10) \qquad -\omega^2\, x_{30} = x_{20} - 2\, x_{30}$$

Now we create the matrix by using a function from an external package. The appropriate statements to *Mathematica* are

```
(%i11)  mat : coefmatrix [eq1, eq2, eq3], [x10, x20, x30] );
```
$$(\%o11) \qquad \begin{pmatrix} 2-\omega^2 & -1 & 0 \\ -1 & 2-\beta\omega^2 & -1 \\ 0 & -1 & 2-\omega^2 \end{pmatrix}$$
```
(%i12)  detm : collectterms( expand(determinant(mat)), \omega );
```
$$(\%o12) \qquad -\beta\omega^6 + (4\beta+2)\,\omega^4 + (-4\beta-6)\,\omega^2 + 4$$
```
(%i13) soln : solve[detm = 0, \omega )$
```

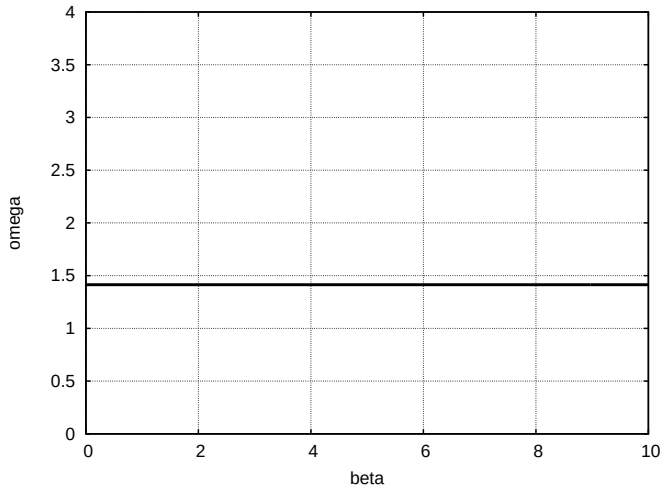
Ignoring the negative solutions, we extract the three solutions of interest with the statements

```
(%i14) w1 : part( soln, 2, 2);
```
$$(\%o14) \qquad \sqrt{2}$$
```
(%i15) w2 : part( soln, 4, 2);
```
$$(\%o15) \qquad \sqrt{\frac{\sqrt{\beta^2+1}}{\beta} + \frac{1}{\beta} + 1}$$
```
(%i16) w3 : part( soln, 6, 2);
```
$$(\%o16) \qquad \sqrt{-\frac{\sqrt{\beta^2+1}}{\beta} + \frac{1}{\beta} + 1}$$

We can find the initial conditions for these frequencies by plugging ω back into `eqn1`, `eqn2`, and `eqn3` with the statements

```
(%i17) amp1 : ev( [ eq1, eq2, eq3 ], \omega = w1 );
```
$$(\%i17) \qquad [-2x_{10} = x_{20} - 2x_{10}, -2\beta x_{20} = x_{30} - 2x_{20} + x_{10}, -2x_{30} = x_{20} - 2x_{30}]$$
```
(%i18) soln1 : solve( amp1, [x10, x20, x30] )
```
solve: dependent equations eliminated: (3)
$$(\%o18) \qquad \left[\, [\, x_{10} = -\%r1, x_{20} = 0, x_{30} = \%r1\,]\, \right]$$
```
(%i19) amp2 : ev( [ eq1, eq2, eq3 ], \omega = w2 )$
```
```
(%i20) soln2 : solve( amp2, [x10, x20, x30] )
```
solve: dependent equations eliminated: (3)

Figure E14.4: Frequency vs. m'/m for the first solution



(%o20) $$\left[\left[x_{10} = \%r2, x_{20} = -\frac{\%r2\sqrt{\beta^2+1} - \%r2\,\beta + \%r2}{\beta}, x_{30} = -\%r2 \right]\right]$$

```
(%i21) amp3 : ev( [ eq1, eq2, eq3 ], \omega = w3 )$
```

```
(%i22) soln3 : solve( amp3, [x10, x20, x30] );
```
solve: dependent equations eliminated: (3)

(%o22) $$\left[\left[x_{10} = \%r3, x_{20} = \frac{\%r3\sqrt{\beta^2+1} + \%r3\,\beta - \%r3}{\beta}, x_{30} = \%r3 \right]\right]$$

To plot the frequency against the ratio of the masses (m'/m), we execute the statements

```
(%i23) plot2d( w1, [\beta, 0, 10], [y, 0, 4], [color, black], [grid2d, true],
               [style, [lines, 4]], [xlabel, "\beta"], [ylabel, "\omega"] );
(%i24) plot2d( w2, [\beta, 0, 10], [y, 0, 4], [color, black], [grid2d, true],
               [style, [lines, 4]], [xlabel, "\beta"], [ylabel, "\omega"] );
(%i25) plot2d( w3, [\beta, 0, 10], [y, 0, 4], [color, black], [grid2d, true],
               [style, [lines, 4]], [xlabel, "\beta"], [ylabel, "\omega"] );
```

```
In[29]:= Plot[ w3[[2]], {\[Beta], 0, 10}, PlotRange -> {0, 4},
               AxesLabel -> {"\[Omega]", "\[Beta]"} ]
```

The results can be seen in Figs. E14.4, E14.5, and E14.6.

In Fig. E14.4 the frequency has no dependence on β. The outside objects are initially displaced the same amount in opposite directions which means the middle object does not move, so its relative mass has no affect on the motion of the other objects. For the two other natural frequencies, the outside objects are displaced the same amount in the *same* direction so the action and relative of the middle object matters a great deal, as one can see from the graphs.

Figure E14.5: Frequency vs. m'/m for the second solution

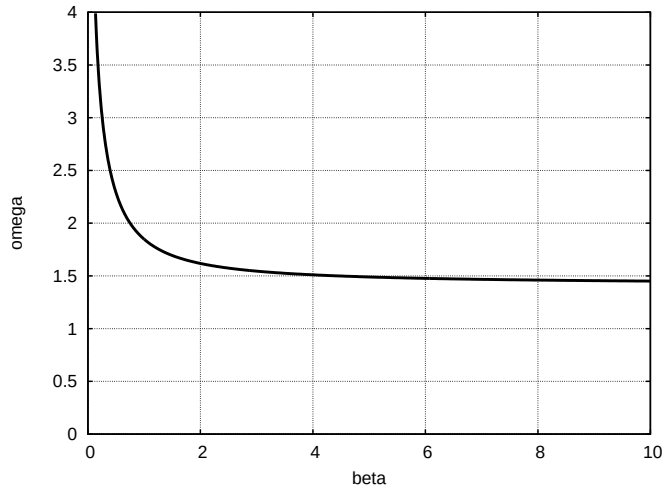
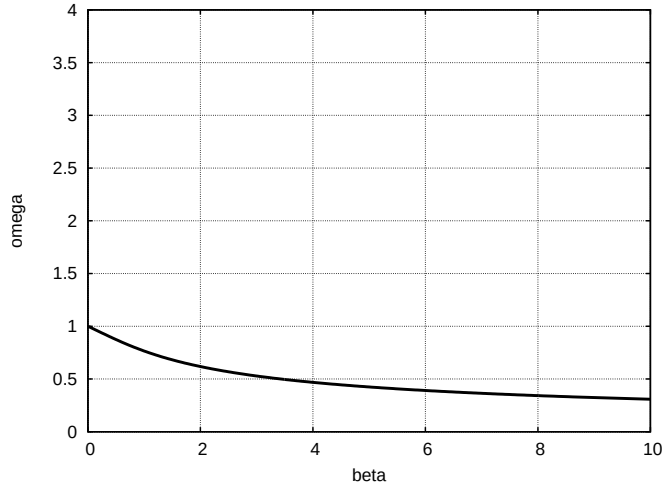


Figure E14.6: Frequency vs. m'/m for the third solution



14.18 Single-Slit Diffraction (MAXIMA)

Exercise: The intensity $I(x)$ in the diffraction pattern produced by a single slit is given by

$$\frac{I(x)}{I_0} = \frac{\sin^2 x}{x^2}$$

where I_0 is the intensity in the center and x is related to the position of the observation point away from the central maximum. The *zeroes* in this pattern are easy to locate (they occur at $x = n\pi$, $n = 0, \pm 1, \pm 2, \ldots$). Careful location of the *maxima*, however, is more complicated. They don't occur where $\sin^2 x = 1$ because of the influence of the denominator that steadily increases as x increases. Locate the positions of the first half dozen maxima in this pattern, which—basically—is a request to find the roots of the derivative of the function (though note that not all roots correspond to *maxima*). Use at least three different methods and at least two different computational tools, and compare the results. Do your results confirm that the roots approach odd multiples of $\frac{1}{2}\pi$ as they become large? *Optional*: You might also find it interesting to approximate the function with a power series expansion for $\sin x$, keeping quite a few terms but converting the root finding problem into that of finding the roots of a polynomial. Then, use methods for finding roots of polynomials and see if you can come to understand how accuracy depends on how many terms you keep and which root you seek.

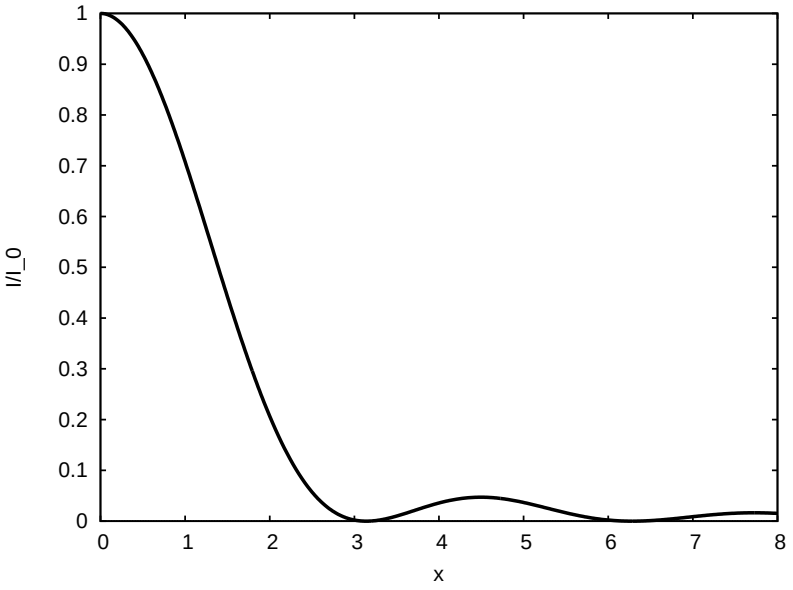
Solution: The intensity $I(x)$ in the diffraction pattern produced by a single slit is given by

$$\frac{I(x)}{I_o} = \frac{\sin^2 x}{x^2}$$

A graph of this function—shown in Fig. E14.7—can be made easily in MAXIMA by entering the statements

```
(%i1) i : (sin(x))^2/x^2$
(%i2) plot2d(i, [x, 0.0001, 8.0], [xlabel, "x"], [ylabel, "I/I_0"],
        [color, black], [style, [lines, 4]] );
```

Figure E14.7: Graph of I/I_0 versus x.



Note that we have set x values to be always *greater* than zero to avoid a **division by 0.0** error. We want to find the extrema of the function. As a start, we must find the roots of $d(I/I_0)/dx$. To use `find_root` (which uses—at least initially—the method of bisection), we can enter the following commands:

```
(%i3) d :  diff( i, x )$
(%i4) find_root(d, x, -0.5, 1.7);
(%o4)     6.816719897495375 × 10^-9
(%i5) find_root(d, x, 1.7, 3.5);
(%o5)     3.141592653589793
(%i6) find_root(d, x, 3.5, 4.6);
(%o6)     4.493409457909064 (%i7) find_root(d, x, 6.0, 6.5);
(%o7)     6.283185307179586 (%i8) find_root(d, x, 7.4, 8.0);
(%o8)     7.725251836937707
```

In this sequence of commands, we are testing for a root in ranges determined by estimating from the graph, specifically, $-0.5 \leq x \leq 1.7$, $1.7 \leq x \leq 3.5$, $3.5 \leq x \leq 4.6$, $6.0 \leq x \leq 6.5$, and $7.4 \leq x \leq 8.0$. Roots are found at $x \approx 0, \pi, \frac{3}{2}\pi, 2\pi, \frac{5}{2}\pi, etc.$ In general, the *maxima* in the graph appear at x equal to odd multiples of $\frac{1}{2}\pi$ and the *minima* in the graph appear at x equal to even multiples of $frac12\pi$, though the maximum at $x = 0$ and the absence of a maximum at $x = \frac{1}{2}\pi$ are exceptions to these conclusions. The second derivative tests,

```
(%i9) dd :  diff(d,x)$
(%i10) ev( dd, x = %o5)
(%o10)     0.2026423672846756
(%i11) ev( dd, x = %o6)
(%o11)     −0.09438089845162256
```

for example, reveal that a minimum occurs at π (second derivative positive) and a maximum occurs at $\frac{3}{2}\pi$ (second derivative negative).

To use `newton`, we only have to supply the equation and an initial guess, though `newton` resides in an auxiliary package which must first be loaded. Thus, the statements

```
(%i12) load(newton1)$
(%i13) newton(d, x, 0.1, 1.0e-10);
(%o13)     7.145228982481614 × 10^-11
(%i14) newton(d, x, 2.0, 1.0e-10);
(%o14)     3.141592653578305
(%i15) newton(d, x, 3.0, 1.0e-10);
(%o15)     3.141592653589792
(%i16) newton(d, x, 4.0, 1.0e-10);
(%o16)     4.49340945766266
(%i17) newton(d, x, 5.0, 1.0e-10);
(%o17)     4.493409457909058
(%i18) newton(d, x, 6.0, 1.0e-10);
(%o18)     6.283185306207734
```

find some of the roots (essentially 0, π, $\frac{3}{2}\pi$, and 2π), though (1) we have specified a somewhat less restrictive tolerance than the default used by `find_root` and (2) convergence to the same root from different starting points result in slightly different values (even though those values differ only beyond digits past those reflecting the declared tolerance).

14.19 Roots of $J_0(x)$ (MAXIMA)

Exercise: Using at least three different methods and at least two different computational tools, find the first half dozen roots of the zeroth-order Bessel function $J_0(x)$. Note that these roots are related to the radii of circular nodes in some of the vibrations of a circular membrane. The values of these roots tabulated in Abramowitz and Stegun[1] are 2.4048255577, 5.5200781103, 8.6537279129, 11.7915344391, 14.9309177086, 18.0710639679. *Hint*: Most computational tools have built-in capabilities for evaluating the Bessel functions. Consult the appropriate vendor manuals.

Solution: We wish to know the first half dozen roots of the zeroth-order Bessel function. The values of the roots as tabulated by Abramowitz and Stegun are 2.40485 55577, 5.52007 81103, 8.65372 79129, 11.79153 44391, 14.93091 77086, and 18.07106 39679. The Bessel function $J_0(x)$ we desire can be found in MAXIMA under the name `bessel_j(0,x)`. To use `find_root`, we need only call up the Bessel function and give MAXIMA a range by executing the statements

```
(%i1) find_root(bessel_j(0, x) , x, 0.0, 4.0);
(%o1)     2.404825557695773
(%i2) find_root(bessel_j(0, x) , x, 4.0, 7.0);
(%o2)     5.520078110286311
```

We can continue this way until the first half dozen roots have been found. By knowing ahead of time what the roots are, it is very easy to give good useful ranges to MAXIMA. If we did not know the roots already (or did not have access to a graph from which to make educated guesses), we could enter any range we wished. We could, for instance, guess $0.0 \leq x \leq 2.0$ or possibly $0.0 \leq x \leq 10.0$. In the first instance, MAXIMA returns the message

find_root: function has same sign at endpoints: `f(0.0)=1.0, f(2.0)= 0.2238907791412356`
– an error. To debug this try: debugmode(true);

and returns us to the top level.[2] We could then try our larger range (in which there happen to be three roots). MAXIMA returns the value 2.404825557695773, the smallest root in the range, and gives no clue that there are additional roots in that range. We could narrow the range again to see if there are any other roots between 2.5 and 10.0 (which returns a message similar to that recorded above, with the same meaning as described in the footnote), or we could choose to examine smaller ranges, like $2.0 \leq x \leq 4.0$, $4.0 \leq x \leq 6.0$, etc.

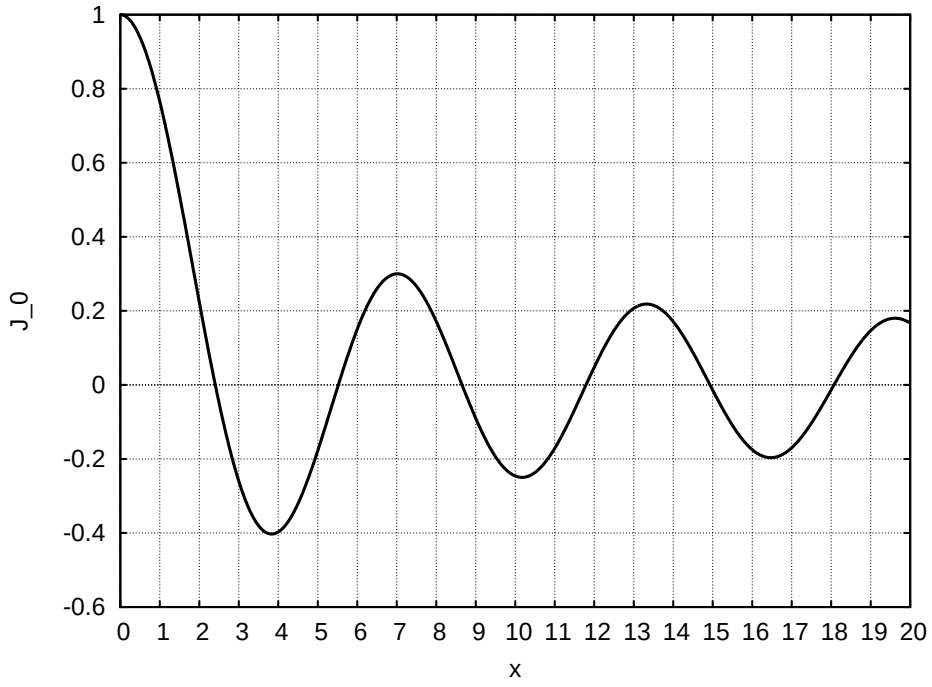
To use `newton`, we must first invoke the package `newton1` and then provide the Bessel function, supply an educated first guess. We try a succession of integers to obtain the first half dozen roots.

```
(%i3) newton(bessel_j(0,x), x, 1.0, 1.0E-10 );
(%o3)     2.404825557695773
(%i4) newton(bessel_j(0,x), x, 2.0, 1.0E-10 );
(%o4)     2.404825557695773
(%i5) newton(bessel_j(0,x), x, 3.0, 1.0E-10 );
(%o5)     2.404825557695773
(%i6) newton(bessel_j(0,x), x, 4.0, 1.0E-10 );
(%o6     33.77582021357357
(%i7) newton(bessel_j(0,x), x, 5.0, 1.0E-10 );
(%o7)     5.520078110080565
```

[1] M. Abramowitz and I. A. Stegun, *Handbook of Mathematical Functions*, U. S. Department of Commerce, Applied Mathematics Series #55, 1964.

[2] The method of bisection depends on the function having the opposite signs at the two ends of the specified interval. This message means that there are zero roots or an even number of single roots or possible some combination that involves one or more multiple roots in the interval.

Figure E14.8: Graph of $J_0(x)$ versus x.




```
(%i8) newton(bessel_j(0,x), x, 6.0, 1.0E-10 );
(%o8)     5.520078110080565
(%i9) newton(bessel_j(0,x), x, 7.0, 1.0E-10 );
(%o9)     −52.62405184111309
(%i10) newton(bessel_j(0,x), x, 8.0, 1.0E-10 );
(%o10)     8.653727912911013
```

Notice that some initial guesses (such as 1, 2, and 3) converge on the same root, which means we have to try more integers to obtain all the roots we want. Note also that some initial guesses tend to fly off to a quite remote root. The reason for that behavior is explained in the next paragraph.

If we did not know the roots already, we could create a graph to find the general area of the roots. We create this graph by executing the statement

```
(%i5) plot2d(bessel_j(0,x), [x, 0.0, 20.0], [xlabel, "x"], [ylabel, "J_0"],
        [style, [lines, 3]], [color, black], [grid, true], [xtics, 1] );
```

The resulting graph is shown in Fig. E14.8. Estimates of the roots of the displayed function can then be made from the scales on the graph. Since Newton's method begins by determining the point at which the tangent to the function at the point of each successive value intersects the horizontal axis, a starting guess close to a point where that tangent is nearly horizontal yields a next iterate that may well be quite remote from the previous iterate.

We have added an unusual number of tic marks on the horizontal axis so we can point out that at $x = 4$ and $x = 7$, the points at which the convergence of Newton's method went skittering off to a remote root, are points very close to a point at which the tangent to the curve is nearly horizontal.

14.20 Double-Welled Potential (MAXIMA)

Exercise: Suppose a particle moves in one dimension under the influence of the potential energy

$$V(x) = \frac{-V_0 a^2(a^2+x^2)}{8a^4+x^4} \qquad \Longrightarrow \qquad \frac{V(x)}{V_0} = -\frac{(1+\overline{x}^2)}{8+\overline{x}^4}$$

where $\overline{x} = x/a$. Using at least three different methods and at least two different computational tools, find the coordinates $\overline{x}$ of all turning points when the total energy E of the particle is $E = -0.2V_0$ and also when the total energy is $E = -0.1V_0$. *Optional*: Obtain graphs of the position of each turning point as a function of particle energy over the allowed range of energies for bound states. Suppose a particle moves in one dimension under the influence of the potential energy

$$V(x) = \frac{-V_0 a^2(a^2+x^2)}{8a^4+x^4} \qquad \Longrightarrow \qquad \frac{V(x)}{V_0} = -\frac{(1+\overline{x}^2)}{8+\overline{x}^4}$$

where $\overline{x} = x/a$. Using at least three different methods and at least two different computational tools, find the coordinates $\overline{x}$ of all turning points when the total energy E of the particle is $E = -0.2V_0$ and also when the total energy is $E = -0.1V_0$. *Optional*: Obtain graphs of the position of each turning point as a function of particle energy over the allowed range of energies for bound states.

Solution: We can examine the one-dimensional motion of a particle under the influence of a potential energy

$$\frac{V(x)}{V_0} = -\frac{1+X^2}{8+X^4}$$

We wish to know the coordinates of all the turning points when the total energy E is $E = -0.1V_0$ as well as when $E = -0.2V_0$. To do this we must find the roots of the equations

$$V_1 = 0.1 - \frac{1+X^2}{8+X^4} \qquad \text{and} \qquad V_2 = 0.2 - \frac{1+X^2}{8+X^4}$$

respectively. (At the roots, the kinetic energy K will be $K = 0$.)

We enter the equations with the statements

```
(%i1) v1 : 0.1-((x^2+1.0)/(x^4+8.0))$
(%i2) v2 : 0.2-((x^2+1.0)/(x^4+8.0))$
```

Then, with the statements

```
(%i3) plot2d(v1, [x, -5.0, 5.0], [color, black], [style, [lines, 4]],
       [xlabel,"X"], [ylabel, "V1"]);
(%i4) plot2d(v1, [x, -5.0, 5.0], [color, black], [style, [lines, 4]],
       [xlabel,"X"], [ylabel, "V2"]);
```

we plot V_1 and V_2 to get an idea of where we ought to look for roots. The resulting graphs are shown in Fig. E14.9 and Fig. E14.10. From these graphs, we conclude (1) that V_1 has two real roots, one in the interval $-4.0 \le x \le -3.0$ and the other in the interval $3.0 \le x \le 4.0$ and (2) that V_2 has four real roots lying, respectively, in the intervals $-3.0 \le x \le -1.5$, $-1.0 \le x \le 0.0$, $0.0 \le x \le 1.0$, and $1.5 \le x \le 3.0$.

Now we simply ask MAXIMA for the roots. To use `find_root` with V_1, we simply enter the statements

```
(%i5) find_root(v1, x, 3.0, 4.0);
(%o5)      3.193141466128087
```

Since the function is even, the two turning points are (3.193141, 0.1) and (-3.193141, 0.1). With V_2, we enter the statements

```
(%i6) find_root(v2, x, 0.0, 1.0);
(%o6)     0.8349996181244668
(%i7) find_root(v2, x, 1.5, 3.0);
(%o7)     2.074313293051943
```

Again, the function is even so the four turning poins are (-2.074313, 0.2), (-0.8349997, 0.2), (0.8349997, 0.2), and (2.074313, 0.2).

To use `newton` we install the package newton1 and examine our graphs in order to obtain initial guesses for each root. We can find them by invoking `newton` with initial guesses, as shown below.

```
(%i8) load(newton1);
(%i9) newton(v1, x, 3.0, 1.0e-10);
(%o9)     3.193141466128087
(%i9) newton(v2, x, 1.0, 1.0e-10);
(%o9)     0.834999618042134
(%i10) newton(v2, x, 3.0, 1.0e-10);
(%o10)     2.074313293051943
```

Figure E14.9: Graph of V_1 versus x.

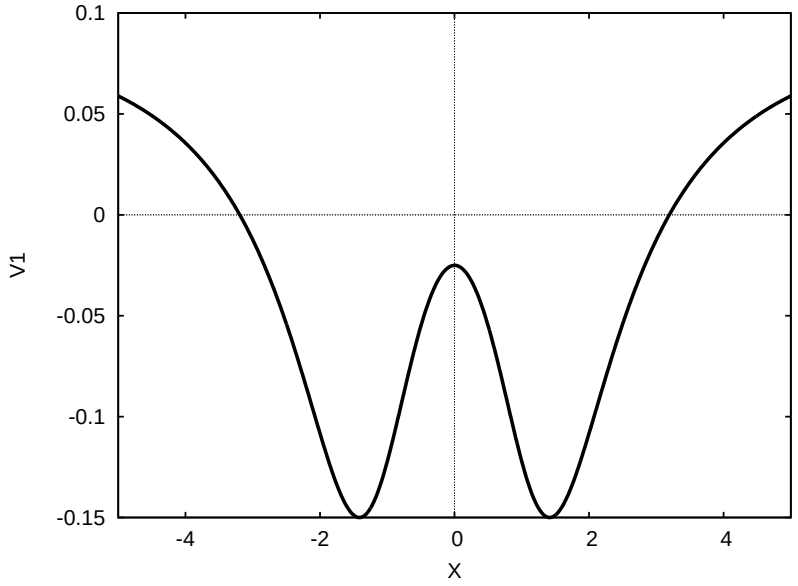
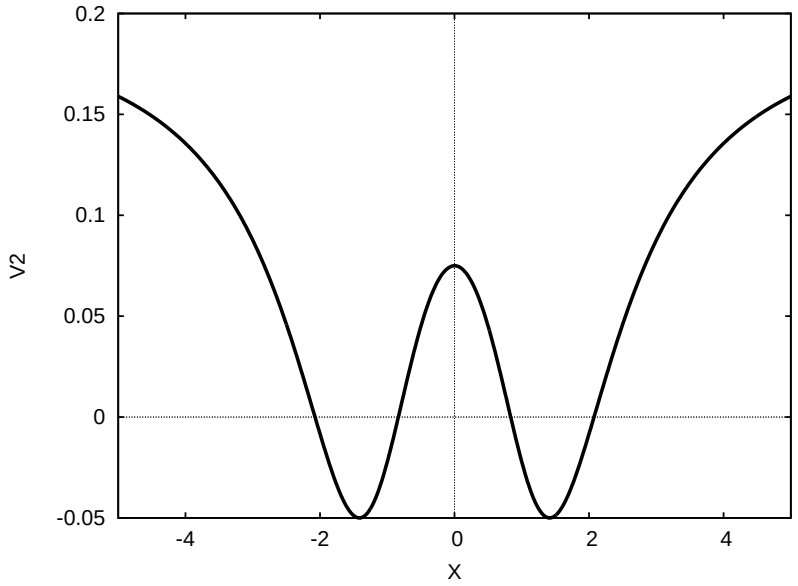


Figure E14.10: Graph of V_2 versus x.



Chapter 15

Introduction to Partial Differential Equations

15.2 A Vertically Hanging String

Exercise: Suppose the one-dimensional string of length l discussed in Section 15.1.1 hangs *vertically* and is acted on by gravity. Suppose that $u(x,t)$ and $v(x,t)$ give the transverse (horizontal) and longitudinal (vertical) displacements of the particle of the string nominally located at x, which is measured downward from the top of the string. (a) Examine the forces acting on this string, deduce the general equations for both longitudinal and transverse motion of the string, and show ultimately that, if the motion is entirely transverse and the amplitude of the motion is small, the equations reduce to

$$\rho(x)\,\frac{\partial^2 u}{\partial t^2} = \frac{\partial}{\partial x}\left(\tau(x)\,\frac{\partial u}{\partial x}\right) \quad ; \quad \tau(x) = -g\int_l^x \rho(x')\,dx'$$

where $\rho(x)$ is the mass per unit length of the string. (b) Taking ρ to be constant, show that the tension $\tau(x)$, which is simply equal to the weight of the string below x, is given by $\tau(x) = \rho g(l-x)$. With this restriction, the equation of motion then becomes

$$\frac{\partial^2 u}{\partial t^2} = \frac{\partial}{\partial x}\left((l-x)\,g\,\frac{\partial u}{\partial x}\right)$$

(c) Recast this equation in dimensionless form by introducing the variables $\overline{x} = x/l$ and $\overline{t} = t\sqrt{g/l}$. (d) Suppose you seek a sinusoidal solution for which $u(\overline{x},\overline{t}) = f(\overline{x})\cos\omega\overline{t}$. Find the ODE satisfied by $f(\overline{x})$ and then introduce the new independent variable y defined by $y^2 = 1-\overline{x}$ to find that, expressed as a function of y, f satisfies the Bessel equation

$$y^2\frac{d^2 f}{dy^2} + y\frac{df}{dy} - 4\omega^2 y^2 f = 0$$

Note: The variable transformation $y^2 = 1-x$ in effect recognizes that the vertical string is more appropriately treated by locating points on the string relative to the bottom rather than the top of the string!

Solution: (a) In order to deduce the equation of motion for a string that hangs vertically and is acted on by gravity, we first determine the forces that act on the string. A force diagram is shown in Fig. E15.1. Then, using Newton's second law, $\mathbf{F} = m\mathbf{a}$, we write the equations of motion. We take the transverse displacement to be $u(x,t)$, and the longitudinal displacement to be $v(x,t)$. Also

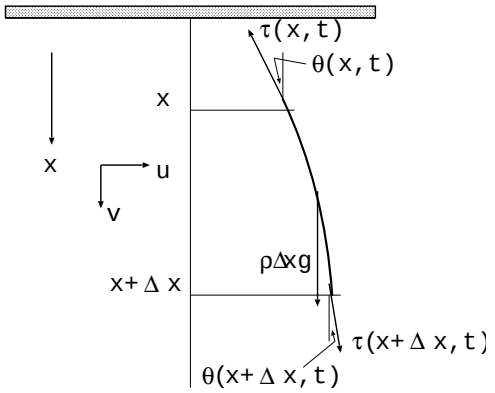


Figure E15.1: Force diagram for a vertically hanging string acted upon by gravity.

note that we have chosen the positive x direction to be down, and we have chosen to measure u positive to the right and v positive down. Then, with the mass of the element shown in the figure given by $\rho(x)\,\Delta x$, we write Newton's second law for that element to be

$$\rho(x)\,\Delta x\,\frac{\partial^2 u}{\partial t^2} = \tau(x+\Delta x,t)\sin\theta(x+\Delta x,t) - \tau(x,t)\sin\theta(x,t)$$

for the transverse motion of the string and

$$\rho(x)\,\Delta x\,\frac{\partial^2 v}{\partial t^2} = \rho(x)\,\Delta x\,g + \tau(x+\Delta x,t)\cos\theta(x+\Delta x,t) - \tau(x,t)\cos\theta(x,t)$$

for the longitudinal motion of the string. Now, we divide by Δx, and let Δx go to zero, reducing the equations of motion to

$$\rho(x)\frac{\partial^2 u}{\partial t^2} = \frac{\partial}{\partial x}\Big(\tau(x,t)\sin\theta(x,t)\Big)$$

and

$$\rho(x)\frac{\partial^2 v}{\partial t^2} = \rho(x)g + \frac{\partial}{\partial x}\Big(\tau(x,t)\cos\theta(x,t)\Big)$$

We have too many unknowns in the above equations, so we must eliminate $\theta(x,t)$. Using Fig. E15.2, we can see that

$$\sin\theta = \frac{\Delta u}{\Delta s} = \frac{\Delta u}{[\Delta u^2 + (\Delta x + \Delta v)^2]^{1/2}} = \frac{\dfrac{\partial u}{\partial x}}{\left[\left(\dfrac{\partial u}{\partial x}\right)^2 + \left(1+\dfrac{\partial v}{\partial x}\right)^2\right]^{1/2}}$$

and that

$$\cos\theta = \frac{\Delta u + \Delta v}{\Delta s} = \frac{1+\dfrac{\partial v}{\partial x}}{\left[\left(\dfrac{\partial u}{\partial x}\right)^2 + \left(1+\dfrac{\partial v}{\partial x}\right)^2\right]^{1/2}}$$

We can now substitute $\sin\theta$ and $\cos\theta$ into the two equations from which we need to eliminate θ to find that

$$\rho(x)\frac{\partial^2 u}{\partial t^2} = \frac{\partial}{\partial x}\left[\frac{\tau(x,t)\dfrac{\partial u}{\partial x}}{\left[\left(\dfrac{\partial u}{\partial x}\right)^2 + \left(1+\dfrac{\partial v}{\partial x}\right)^2\right]^{1/2}}\right]$$

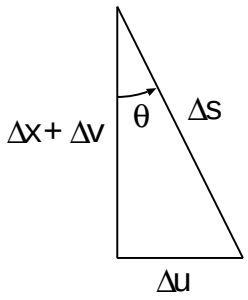


Figure E15.2: Geometry for determining $\theta(x,t)$ from $u(x,t)$ and $v(x,t)$.

$$\rho(x)\frac{\partial^2 v}{\partial t^2} = \rho(x)g + \frac{\partial}{\partial x}\left[\frac{\tau(x,t)\left(1+\dfrac{\partial v}{\partial x}\right)}{\left[\left(\dfrac{\partial u}{\partial x}\right)^2 + \left(1+\dfrac{\partial v}{\partial x}\right)^2\right]^{1/2}}\right]$$

At this point, we have only the unknowns that we are looking for, $u(x,t)$ and $v(x,t)$. In order to be able to solve these equations, we must make a few approximations. If (1) the amplitude of the motion is small so that

$$\Delta u \ll \Delta x \quad \Longrightarrow \quad \frac{\partial u}{\partial x} \ll 1$$

and (2) the motion is transverse so that $v = 0$ everywhere and always, then the equation for v reduces to

$$0 = \rho(x)\, g + \frac{\partial \tau}{\partial x} \quad \Longrightarrow \quad \tau(x) = -\int_l^x \rho(x')\, g\, dx'$$

where we find that, in this limit, τ cannot depend on t. (The limits are chosen so that τ will be zero at the bottom of the string where $x = l$ and grow to the total weight of the string at the top of the string, where $x = 0$.) In this same limit, the equation for u becomes

$$\rho(x)\frac{\partial^2 u}{\partial t^2} = \frac{\partial}{\partial x}\left[\tau(x)\frac{\partial u}{\partial x}\right]$$

In the special case that $\rho(x)$ is, in fact, constant, then the integral giving $\tau(x)$ can be quickly evaluated to give that

$$\tau(x) = \rho\, g\, (l - x)$$

and, in consequence, the equation for u becomes

$$\frac{\partial^2 u}{\partial t^2} = \frac{\partial}{\partial x}\left[g\,(l-x)\,\frac{\partial u}{\partial x}\right]$$

We next cast this result in a dimensionless form by introducing the variables $\overline{x} = x/l$ and $\overline{t} = t\sqrt{g/l}$ to find that

$$\frac{\partial^2 u}{\partial t^2} = \frac{\partial}{l\,\partial \overline{x}}\left[gl\,(1-\overline{x})\,\frac{\partial u}{l\,\partial \overline{x}}\right] = \frac{g}{l}\,\frac{\partial}{\partial \overline{x}}\left[(1-\overline{x})\,\frac{\partial u}{\partial \overline{x}}\right]$$

and then that

$$\frac{\partial^2 u}{\partial \overline{t}^2} = \frac{\partial}{\partial \overline{x}}\left[(1-\overline{x})\,\frac{\partial u}{\partial \overline{x}}\right]$$

Suppose, now, that $u(\overline{x},\overline{t}) = f(\overline{x})\cos\omega\overline{t}$. Substitution of this form into the equation we have deduced leads to the result

$$\omega^2 f = \frac{d}{d\overline{x}}\left[(1-\overline{x})\,\frac{df}{d\overline{x}}\right]$$

Finally, let us substitute the independent variable y defined by $y^2 = 1-\overline{x}$, in terms of which

$$2y\,\frac{dy}{dx} = -1 \quad\Longrightarrow\quad \frac{dy}{dx} = -\frac{1}{2y}$$

With that change, the equation becomes

$$\omega^2 f = -\frac{1}{2y}\frac{d}{dy}\left[y^2\left(-\frac{1}{2y}\right)\frac{df}{dy}\right] = \frac{1}{4y}\frac{d}{dy}\left[y\frac{df}{dy}\right] = \frac{1}{4y}\left[y\frac{d^2f}{dy^2} + \frac{df}{dy}\right]$$

Multiplying by $4y^2$ bringing all terms to the same side of the equation, we finally find that

$$y^2\frac{d^2f}{dy^2} + y\frac{df}{dy} - 4\omega^2 y^2\, f = 0$$

which is, in fact, the zeroth-order Bessel equation.

15.3 Second-Order ODEs Made Self-Adjoint

Exercise: Show that every second-order linear ODE can be cast in self-adjoint form, i.e., show that functions $\alpha(x)$, $\gamma(x)$, and $f(x)$ can be found such that the general second-order linear ODE

$$a(x)\,\frac{d^2\varphi}{dx^2} + b(x)\frac{d\varphi}{dx} + c(x)\,\varphi = g(x)$$

can be recast in self-adjoint form as defined in Section 15.1.7. *Hint*: Multiply the equation by an undetermined integrating factor $h(x)$, find a (first-order) differential equation for $h(x)$, and solve that equation—at least formally.

Solution: We want to show that the general second-order ODE

$$a(x)\,\frac{d^2\varphi}{dx^2} + b(x)\,\frac{d\varphi}{dx} + c(x)\,\varphi(x) = g(x)$$

can be recast in self-adjoint form, specifically

$$-\alpha(x)\,\frac{d^2\varphi(x)}{dx^2} - \alpha'(x)\,\frac{d\varphi(x)}{dx} + \beta(x)\,\varphi(x) = f(x)$$

We'll begin by multiplying the general equation by an undetermined function $h(x)$, which we will later find as an integrating factor, obtaining that

$$h(x)\,a(x)\,\frac{d^2\varphi}{dx^2} + h(x)\,b(x)\,\frac{d\varphi}{dx} + h(x)\,c(x)\,\varphi(x) = h(x)\,g(x)$$

We can easily see that, for this equation to be recast in self-adjoint form, we must choose $h(x)$ so that

$$h(x)\,a(x) = -\alpha(x) \quad \text{and} \quad h(x)\,b(x) = -\alpha'(x)$$

or

$$\frac{d}{dx}\Big(h(x)\,a(x)\Big) = h(x)\,b(x) \quad \Longrightarrow \quad h'(x)\,a(x) + h(x)\,a'(x) = h(x)\,b(x)$$

Equivalently, we have that

$$h'(x) = \frac{dh}{dx} = h(x)\,\left(\frac{b(x) - a'(x)}{a(x)}\right) \quad \Longrightarrow \quad \frac{dh}{h} = \left(\frac{b(x) - a'(x)}{a(x)}\right)\,dx$$

This last expression can be integrated (indefinite integral) to yield that

$$\int \frac{dh}{h} = \int \left(\frac{b(x) - a'(x)}{a(x)}\right)\,dx \quad \Longrightarrow \quad \ln h = \int \frac{b(x)}{a(x)}\,dx - \ln a + C'$$

or, on exponentiation, that

$$h(x) = C\,\frac{e^{\int (b(x)/a(x))\,dx}}{a(x)}$$

where $C = e^{C'}$ is an arbitrary constant.

Having found the integrating factor, we now multiply the original equation by this factor—though we can ignore the constant C to find that

$$e^{\int (b/a)\,dx}\,\frac{d^2\varphi(x)}{dx^2} + \frac{b}{a}\,e^{\int (b/a)\,dx}\,\frac{d\varphi(x)}{dx} + \frac{c}{a}e^{\int (b/a)dx}\varphi(x) = \frac{g}{a}e^{\int (b/a)dx}$$

If this equation is to match the desired form, we must make the identification

$$\alpha(x) = -e^{\int (b/a)\,dx}$$

With that choice, we then find that

$$\alpha'(x) = \frac{d}{dx}\left(-e^{\int b/a\,dx}\right) = -e^{\int (b/a)\,dx}\,\frac{d}{dx}\int \frac{b}{a}\,dx = -\frac{b}{a}\,e^{\int (b/a)\,dx}$$

which is exactly the coefficient of the derivative $d\varphi/dx$ in the first equation in this paragraph. To finish, we recognize by comparing the ODE just derived with the second equation in this solution that

$$\beta(x) = \frac{c}{a}\,e^{\int (b/a)\,dx}$$

and

$$f(x) = \frac{g}{a}\,e^{\int (b/a)\,dx}$$

15.7 FDM 1D Homogeneous Helmholtz Equation

Exercise: Consider the differential equation and boundary conditions

$$\frac{d^2u}{dx^2} + k^2 u = 0 \qquad ; \qquad u(0) = u(L) = 0$$

(a) Find the analytic solution to this problem and identify the special values of k that permit non-trivial solutions. (b) Show that the finite difference approach to the problem leads to the matrix eigenvalue problem $Au = -k^2L^2u/N^2$ where N is the number of equal-length segments into which the length L of the domain is divided and

$$A = \begin{pmatrix} -2 & 1 & 0 & \cdots & 0 & 0 & 0 \\ 1 & -2 & 1 & \cdots & 0 & 0 & 0 \\ 0 & 1 & -2 & \cdots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & -2 & 1 & 0 \\ 0 & 0 & 0 & \cdots & 1 & -2 & 1 \\ 0 & 0 & 0 & \cdots & 0 & 1 & -2 \end{pmatrix} \quad ; \quad u = \begin{pmatrix} u_1 \\ u_2 \\ u_3 \\ \vdots \\ u_{N-3} \\ u_{N-2} \\ u_{N-1} \end{pmatrix}$$

Note that, when the domain $0 \le x \le L$ is divided into N segments, there will be $N+1$ nodes ranging from $x_0 = 0$ to $x_{N+1} = L$. Because, along the way to a solution, the boundary conditions result in the rows and columns associated with these two nodes being deleted, these matrices will have only $N - 1$ rows and columns. (c) Taking $N = 100$, use an available numeric processing program like IDL, MATLAB, OCTAVE, or PYTHON to find the first several eigenvalues k_n and compare your results with the values found in part (a).

Solution: (a) The general solution to the differential equation is well known to be

$$u(x) = A \sin kx + B \cos kx$$

The boundary conditions require that

$$u(0) = B = 0 \quad ; \quad u(L) = A \sin kL = 0 \Longrightarrow kL = n\pi, n = 1, 2, 3, ...$$

so there is a family of solutions

$$u(x) = A \sin\left(\frac{n\pi x}{L}\right) = A \sin k_n x \quad ; \quad k_n = nk_1 = n\frac{\pi}{L}$$

The non-trivial solutions are sinusoidal with arbitrary amplitude,and the wave number k_n associated with the n-th solution is an integer multiple of the wave number $k_1 = \pi/L$ associated with the first solution.

(b) To cast the equation

$$\frac{d^2u}{dx^2} + k^2 u = 0$$

in a discretized form, we divide the interval $0 \le x \le L$ into N segments by placing nodes at the points $x_i = i\,\Delta x$ where $\Delta x = L/N$, $0 \le i \le N$, introduce the quantities $u_i = u(x_i)$, and remember that, with that grid,

$$\left.\frac{d^2u}{dx^2}\right|_{x=x_i} = \frac{u_{i+1} - 2u_i + u_{i-1}}{\Delta x^2}$$

Thus, we find that evaluating the differential equation at $x = x_i$ leads to the algebraic equation

$$\frac{u_{i+1} - 2u_i + u_{i-1}}{\Delta x^2} + k^2 u_i = 0 \quad \Longrightarrow \quad u_{i+1} - 2u_i + u_{i-1} = -k^2\,\Delta x^2\,u_i = -\frac{k^2L^2}{N^2}u_i$$

At the two ends of the domain ($i = 0$ and $i = N$), however, this equation involves quantities—u_{-1} and u_{N+1}—that are outside the domain, so we don't know them. But the boundary conditions allow us to ignore the original equations,

$$u_1 - 2u_0 + u_{-1} = -\frac{k^2L^2}{N^2}u_0 \quad ; \quad u_{N+1} - 2u_N + u_{N-1} = -\frac{k^2L^2}{N^2}u_N$$

replacing them with the equations

$$u_0 = 0 \quad ; \quad u_N = 0$$

respectively. In total, then, the equations we seek to solve are

$$\begin{aligned} u_0 &= 0 \\ u_{i+1} - 2u_i + u_{i-1} &= -\frac{k^2L^2}{N^2}u_i \quad 1 \le i \le N-1 \\ u_N &= 0 \end{aligned}$$

To recast these equations in a matrix form, we would leave out the first and last (for which we already know a solution) and introduce the vector

$$\Big(u_1, u_2, u_3, \ldots, u_{N-2}, u_{N-1}\Big)$$

(though we should probably write it as a column rather than a row). Further, we then note that we need to single out the equations for $i = 1$ and $i = N - 1$ for special treatment since, without that treatment the first of these refers to u_0 and the second refers to u_N, which we have omitted from the equations to be solved. For $i = 1$, for example, the equation becomes

$$u_2 - 2u_1 + u_0 = -\frac{k^2L^2}{N^2}u_1 \quad \Longrightarrow \quad u_2 - 2u_1 = -\frac{k^2L^2}{N^2}u_1$$

(Remember that $u_0 = 0$). Similarly, the equation for $i = N - 1$ becomes

$$u_{N-2} - 2u_{N-1} + u_N = -\frac{k^2L^2}{N^2}u_{N-1} \quad \Longrightarrow \quad u_{N-2} - 2u_{N-1} = -\frac{k^2L^2}{N^2}u_{N-1}$$

(Remember that $u_N = 0$.) Finally, we conclude that the equations to be solved are

$$\begin{aligned} -2u_1 + u_2 &= -\frac{k^2L^2}{N^2}u_1 \\ u_{i-1} - 2u_i + u_{i+1} &= -\frac{k^2L^2}{N^2}u_i \qquad 2 \le i \le N-2 \\ u_{N-2} - 2u_{N-1} &= -\frac{k^2L^2}{N^2}u_{N-1} \end{aligned}$$

Here, we have reversed the order of the terms on the left-hand side so that the indices increase across the equation rather than decrease, thus rendering more evident the equivalent—and more compact—matrix presentation

$$\begin{pmatrix} -2 & 1 & 0 & 0 & \ldots & 0 & 0 & 0 \\ 1 & -2 & 1 & 0 & \ldots & 0 & 0 & 0 \\ 0 & 1 & -2 & 1 & \ldots & 0 & 0 & 0 \\ 0 & 0 & 1 & -2 & \ldots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ldots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \ldots & -2 & 1 & 0 \\ 0 & 0 & 0 & 0 & \ldots & 1 & -2 & 1 \\ 0 & 0 & 0 & 0 & \ldots & 0 & 1 & -2 \end{pmatrix} \begin{pmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \\ \cdots \\ u_{N-3} \\ u_{N-2} \\ u_{N-1} \end{pmatrix} = -\frac{k^2L^2}{N^2} \begin{pmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \\ \cdots \\ u_{N-3} \\ u_{N-2} \\ u_{N-1} \end{pmatrix}$$

In terms of these quantities, the problem we must solve is the simple eigenvalue problem

$$Au = \lambda u = -\frac{k^2 L^2}{N^2}\, u \Longrightarrow k_n = \sqrt{-\lambda_n}\,\frac{N}{L} \Longrightarrow \frac{k_n}{k_1} = \frac{\sqrt{-\lambda_n}}{\sqrt{-\lambda_1}}$$

where A is the coefficient matrix, u is a vector of the unknown values at the nodes 1 to $N-1$, and $\lambda = -k^2L^2/N^2$ is the eigenvalue to be found. At this point, we hope that the values of λ_n turn out to be negative. Note that these matrices have dimension $(N-1)\times(N-1)$, even though there are $N+1$ nodes in the discretization on this domain.

15.8 FEM 1D Homogeneous Helmholtz Equation

Exercise: Consider the differential equation and boundary conditions

$$\frac{d^2u}{dx^2} + k^2u = 0 \qquad ; \qquad u(0) = u(L) = 0$$

(a) Find the analytic solution to this problem and identify the special values of k that permit non-trivial solutions. (b) Show that the finite element approach to the problem leads to the *generalized* eigenvalue problem $Au = -k^2L^2Bu/6N^2$ where N is the number of equal-length elements into which the length L of the domain is divided and

$$A = \begin{pmatrix} -2 & 1 & 0 & \cdots & 0 & 0 & 0 \\ 1 & -2 & 1 & \cdots & 0 & 0 & 0 \\ 0 & 1 & -2 & \cdots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & -2 & 1 & 0 \\ 0 & 0 & 0 & \cdots & 1 & -2 & 1 \\ 0 & 0 & 0 & \cdots & 0 & 1 & -2 \end{pmatrix} \quad ; \quad B = \begin{pmatrix} 4 & 1 & 0 & \cdots & 0 & 0 & 0 \\ 1 & 4 & 1 & \cdots & 0 & 0 & 0 \\ 0 & 1 & 4 & \cdots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 4 & 1 & 0 \\ 0 & 0 & 0 & \cdots & 1 & 4 & 1 \\ 0 & 0 & 0 & \cdots & 0 & 1 & 4 \end{pmatrix}$$

and

$$u = \begin{pmatrix} u_1 \\ u_2 \\ u_3 \\ \vdots \\ u_{N-3} \\ u_{N-2} \\ u_{N-1} \end{pmatrix}$$

Note that, when the domain $0 \le x \le L$ is divided into N segments, there will be $N+1$ nodes ranging from $x_0 = 0$ to $x_{N+1} = L$. Because, along the way to a solution, the boundary conditions result in the rows and columns associated with these two nodes being deleted, these matrices will have only $N - 1$ rows and columns. (c) Taking $N = 100$, use an available numeric processing program like IDL, MATLAB, OCTAVE, or PYTHON to find the first several eigenvalues k_n and compare your results with the values found in part (a).

Solution: (a) The general solution to the differential equation is well known to be

$$u(x) = A\sin kx + B\cos kx$$

The boundary conditions require that

$$u(0) = B = 0 \quad ; \quad u(L) = A\sin kL = 0 \Longrightarrow kL = n\pi, n = 1, 2, 3, ...$$

so there is a family of solutions

$$u(x) = A\sin\left(\frac{n\pi x}{L}\right) = A\sin k_n x \quad ; \quad k_n = nk_1 = n\frac{\pi}{L}$$

The non-trivial solutions are sinusoidal with arbitrary amplitude,and the wave number k_n associated with the n-th solution is an integer multiple of the wave number $k_1 = \pi/L$ associated with the first solution.

(b) We begin by identifying the values of the parameters in Eq. (15.67) that will reduce that general equation to the equation of current interest. Specifically, we must identify $\alpha = -1$, $\beta = k^2$, and $f = 0$. Further, to yield the boundary conditions in this exercise, we must set $p = 0$, $\gamma = 0$, and $q = 0$.

Following the pattern laid out in Section 15.9, we begin by discretizing the domain $0 \le x \le L$ into equally-length elements whose end points lie at

$$x_i = i\left(\frac{L}{N}\right) \quad , \quad i = 0, 1, 2, \ldots, N$$

so that, for the e-th element,

$$x_1^{(e)} = x_e \quad ; \quad x_2^{(e)} = x_{e+1} = x_e + L/N \quad , \quad e = 0, 1, 2, \ldots, N-1$$

With these choices, the interpolation functions in Section 15.9.2 reduce to

$$N_1^{(e)}(x) = \frac{x_2^{(e)} - x}{L/N} \quad ; \quad N_2^{(e)}(x) = \frac{x - x_1^{(e)}}{L/N}$$

and the approximate solution over the e-th element is given by

$$\tilde{u}^{(e)}(x) = \sum_{j=1}^{2} N_1^{(e)}(x)\, \tilde{u}_j^{(e)}$$

where the $\tilde{u}_j^{(e)}$s for $j = 1, 2$ are *constants* to be determined. The residual $r(x)$ as defined in Section 15.9.3 is for this exercise given by

$$r(x) = \frac{d^2u}{dx^2} + k^2 u$$

The weighted averages of this residual over the e-th element when the weights are the interpolation functions then are defined by

$$\begin{aligned} R_i^{(e)} &= \int_{x_1^{(e)}}^{x_2^{(e)}} N_i^{(e)} \left(\frac{d^2\tilde{u}^{(e)}}{dx^2} + k^2\tilde{u}^{(e)}\right) dx \\ &= -\int_{x_1^{(e)}}^{x_2^{(e)}} \left(\frac{dN_i^{(e)}}{dx}\frac{d\tilde{u}^{(e)}}{dx} - k^2 N_i^{(e)}\tilde{u}^{(e)}\right) dx + \left(N_i^{(e)}\frac{d\tilde{u}^{(e)}}{dx}\right)\Bigg|_{x_1^{(e)}}^{x_2^{(e)}} \quad ; \quad i = 1, 2 \end{aligned}$$

Here, the second form is calculated by evaluating the integral of the second derivative in the first form by parts. Substituting from the above equation for $u^{(e)}(x)$, we next learn that

$$R_i^{(e)} = -\left[\sum_{j=1}^{2} \tilde{u}_j^{(e)} \int_{x_1^{(e)}}^{x_2^{(e)}} \left(\frac{dN_i^{(e)}}{dx}\frac{dN_j^{(e)}}{dx} - k^2 N_i^{(e)}\, N_j^{(e)}\right) dx\right] + \left(N_i^{(e)}\frac{d\tilde{u}^{(e)}}{dx}\right)\Bigg|_{x_1^{(e)}}^{x_2^{(e)}}$$

In matrix form, these several equations for $1 \le i, j \le 2$ have the form

$$\left\{ \begin{array}{c} R_1^{(e)} \\ R_2^{(e)} \end{array} \right\} = \left[\begin{array}{cc} K_{11}^{(e)} & K_{12}^{(e)} \\ K_{21}^{(e)} & K_{22}^{(e)} \end{array} \right] \left\{ \begin{array}{c} \tilde{u}_1^{(e)} \\ \tilde{u}2^{(e)} \end{array} \right\} + \left\{ \begin{array}{c} g_1^{(e)} \\ g_2^{(e)} \end{array} \right\}$$

or more compactly as

$$\{R^{(e)}\} = [K^{(e)}]\,\{\tilde{u}^{(e)}\} + \{g^{(e)}\}$$

where

$$K_{ij}^{(e)} = -\int_{x_1^{(e)}}^{x_2^{(e)}} \left(\frac{dN_i^{(e)}}{dx}\,\frac{dN_j^{(e)}}{dx} - k^2\, N_i^{(e)}\, N_j^{(e)}\right) dx$$

and

$$g_i^{(e)} = \left(N_i^{(e)}\,\frac{d\tilde{u}^{(e)}}{dx}\right)\Bigg|_{x_1^{(e)}}^{x_2^{(e)}} = \left(N_i^{(e)}\,\frac{d\tilde{u}}{dx}\right)\Bigg|_{x=x_2^{(e)}} - \left(N_i^{(e)}\,\frac{d\tilde{u}}{dx}\right)\Bigg|_{x=x_1^{(e)}}$$

We have several integrals to evaluate. Because $dN_1^{(e)}/dx = -L/N$ and $dN_2^{(e)}/dx = L/N$ or, more compactly $dN_k^{(e)} = (-1)^k\, L/N$, we conclude quickly that

$$\int_{x_1^{(e)}}^{x_2^{(e)}} \frac{dN_i^{(e)}}{dx}\,\frac{dN_j^{(e)}}{dx}\,dx = \int_{x_1^{(e)}}^{x_2^{(e)}} (-1)^i\,\frac{L}{N}\,(-1)^j\,\frac{L}{N}\,dx = (-1)^{i+j}\frac{L^2}{N^2}(x_2^{(e)} - x_1^{(e)}) = (-1)^{i+j}\frac{L^3}{N^3}$$

Next, note that[1]

$$\int_{x_1^{(e)}}^{x_2^{(e)}} N_1^{(e)}\,N_1^{(e)}\,dx = \frac{L^2}{N^2}\int_{x_1^{(e)}}^{x_2^{(e)}} (x_2^{(e)} - x)^2\,dx = \frac{L^5}{3N^5}$$

$$\int_{x_1^{(e)}}^{x_2^{(e)}} N_2^{(e)}\,N_2^{(e)}\,dx = \frac{L^2}{N^2}\int_{x_1^{(e)}}^{x_2^{(e)}} (x - x_1^{(e)})^2\,dx = \frac{L^5}{3N^5}$$

$$\int_{x_1^{(e)}}^{x_2^{(e)}} N_1^{(e)}\,N_2^{(e)}\,dx = \int_{x_1^{(e)}}^{x_2^{(e)}} N_2^{(e)}\,N_1^{(e)}\,dx = \frac{L^2}{N^2}\int_{x_1^{(e)}}^{x_2^{(e)}} (x_2^{(e)} - x)(x - x_1^{(e)})\,dx = \frac{L^5}{6N^5}$$

(Remember that $x_2^{(e)} = x_1^{(e)} + L/N$.) We conclude that

$$\begin{aligned}\left[K^{(e)}\right] &= -\begin{bmatrix} L^3/N^3 & -L^3/N^3 \\ -L^3/N^3 & L^3/N^3 \end{bmatrix} + k^2\begin{bmatrix} L^5/3N^5 & L^5/6N^5 \\ L^5/6N^5 & L^5/3N^5 \end{bmatrix} \\ &= \frac{L^3}{N^3}\begin{bmatrix} -1 & 1 \\ 1 & -1 \end{bmatrix} + \frac{k^2L^5}{6N^5}\begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix}\end{aligned}$$

To complete the evaluation, we note that[2]

$$g_1^{(e)} = \left(N_1^{(e)}\,\frac{d\tilde{u}}{dx}\right)\Big|_{x=x_2^{(e)}} - \left(N_1^{(e)}\,\frac{d\tilde{u}}{dx}\right)\Big|_{x=x_1^{(e)}} = -\tilde{u}'(x_1^{(e)}) = -\tilde{u}'(x_e)$$

and

$$g_2^{(e)} = \left(N_2^{(e)}\,\frac{d\tilde{u}}{dx}\right)\Big|_{x=x_2^{(e)}} - \left(N_2^{(e)}\,\frac{d\tilde{u}}{dx}\right)\Big|_{x=x_1^{(e)}} = \tilde{u}'(x_2^{(e)}) = \tilde{u}'(x_{e+1})$$

so

$$g^{(e)} = \begin{Bmatrix} -\tilde{u}'(x_e) \\ \tilde{u}'(x_{e+1}) \end{Bmatrix}$$

Setting the average residual on the e-th element to zero, we conclude that the elemental equation for the e-th element is

$$\{R^{(e)}\} = \left(\frac{L^3}{N^3}\begin{bmatrix} -1 & 1 \\ 1 & -1 \end{bmatrix} + \frac{k^2L^5}{6N^5}\begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix}\right)\begin{Bmatrix} \tilde{u}_e \\ \tilde{u}_{e+1} \end{Bmatrix} + \begin{Bmatrix} -\tilde{u}'(x_e) \\ \tilde{u}'(x_{e+1}) \end{Bmatrix} = \begin{Bmatrix} 0 \\ 0 \end{Bmatrix}$$

or

$$\left(\begin{bmatrix} -1 & 1 \\ 1 & -1 \end{bmatrix} + \frac{k^2L^2}{6N^2}\begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix}\right)\begin{Bmatrix} \tilde{u}_e \\ \tilde{u}_{e+1} \end{Bmatrix} = \frac{N^3}{L^3}\begin{Bmatrix} -\tilde{u}'(x_e) \\ \tilde{u}'(x_{e+1}) \end{Bmatrix}$$

or, with $\beta = k^2L^2/6N^2$,

$$\begin{bmatrix} -1+2\beta & 1+\beta \\ 1+\beta & -1+2\beta \end{bmatrix}\begin{Bmatrix} \tilde{u}_e \\ \tilde{u}_{e+1} \end{Bmatrix} = \frac{N^3}{L^3}\begin{Bmatrix} -\tilde{u}'(x_e) \\ \tilde{u}'(x_{e+1}) \end{Bmatrix}$$

[1] These integrals are simple enough to be evaluated by hand. Alternatively, they can be evaluated with an available symbolic manipulating program like MAXIMA, MAPLE, or *Mathematica*. The details of those evaluations are omitted.

[2] Remember that $N_1^{(e)}(x_1^{(e)}) = 1$, $N_1^{(e)}(x_2^{(e)}) = 0$, $N_2^{(e)}(x_1^{(e)}) = 0$, and $N_2^{(e)}(x_2^{(e)}) = 1$.

As guidance for inferring the general case, we write out these equation for a system in which there are six nodes ($N = 6, i = 0, 1, 2, 3, 4, 5$) and five elements $e = 0, 1, 2, 3, 4$. For $e = 0$, the equation, expanded to involve 6×6 matrices and six-element vectors, is[3]

$$\begin{bmatrix} -1+2\beta & 1+\beta & 0 & 0 & 0 & 0 \\ 1+\beta & -1+2\beta & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{Bmatrix} \tilde{u}_0 \\ \tilde{u}_1 \\ \tilde{u}_2 \\ \tilde{u}_3 \\ \tilde{u}_4 \\ \tilde{u}_5 \end{Bmatrix} = \frac{N^3}{L^3} \begin{Bmatrix} -\tilde{u}'(x_0) \\ \tilde{u}'(x_1) \\ 0 \\ 0 \\ 0 \\ 0 \end{Bmatrix}$$

For $e = 1$,

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1+2\beta & 1+\beta & 0 & 0 & 0 \\ 0 & 1+\beta & -1+2\beta & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{Bmatrix} \tilde{u}_0 \\ \tilde{u}_1 \\ \tilde{u}_2 \\ \tilde{u}_3 \\ \tilde{u}_4 \\ \tilde{u}_5 \end{Bmatrix} = \frac{N^3}{L^3} \begin{Bmatrix} 0 \\ -\tilde{u}'(x_1) \\ \tilde{u}'(x_2) \\ 0 \\ 0 \\ 0 \end{Bmatrix}$$

For $e = 2$,

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1+2\beta & 1+\beta & 0 & 0 \\ 0 & 0 & 1+\beta & -1+2\beta & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{Bmatrix} \tilde{u}_0 \\ \tilde{u}_1 \\ \tilde{u}_2 \\ \tilde{u}_3 \\ \tilde{u}_4 \\ \tilde{u}_5 \end{Bmatrix} = \frac{N^3}{L^3} \begin{Bmatrix} 0 \\ 0 \\ -\tilde{u}'(x_2) \\ \tilde{u}'(x_3) \\ 0 \\ 0 \end{Bmatrix}$$

For $e = 3$

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1+2\beta & 1+\beta & 0 \\ 0 & 0 & 0 & 1+\beta & -1+2\beta & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{Bmatrix} \tilde{u}_0 \\ \tilde{u}_1 \\ \tilde{u}_2 \\ \tilde{u}_3 \\ \tilde{u}_4 \\ \tilde{u}_5 \end{Bmatrix} = \frac{N^3}{L^3} \begin{Bmatrix} 0 \\ 0 \\ 0 \\ -\tilde{u}'(x_3) \\ \tilde{u}'(x_4) \\ 0 \end{Bmatrix}$$

[3] We have also rearranged the terms a bit to reduce the "width" of the equation.

For $e = 4$

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1+2\beta & 1+\beta \\ 0 & 0 & 0 & 0 & 1+\beta & -1+2\beta \end{bmatrix} \begin{Bmatrix} \tilde{u}_0 \\ \tilde{u}_1 \\ \tilde{u}_2 \\ \tilde{u}_3 \\ \tilde{u}_4 \\ \tilde{u}_5 \end{Bmatrix} = \frac{N^3}{L^3} \begin{Bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ -\tilde{u}'(x_4) \\ \tilde{u}'(x_5) \end{Bmatrix}$$

The next step is to add these equations to find the single matrix equation

$$\begin{bmatrix} -1+2\beta & 1+\beta & 0 & 0 & 0 & 0 \\ 1+\beta & -2+4\beta & 1+\beta & 0 & 0 & 0 \\ 0 & 1+\beta & -2+4\beta & 1+\beta & 0 & 0 \\ 0 & 0 & 1+\beta & -2+4\beta & 1+\beta & 0 \\ 0 & 0 & 0 & 1+\beta & -2+4\beta & 1+\beta \\ 0 & 0 & 0 & 0 & 1+\beta & -1+2\beta \end{bmatrix} \begin{Bmatrix} \tilde{u}_0 \\ \tilde{u}_1 \\ \tilde{u}_2 \\ \tilde{u}_3 \\ \tilde{u}_4 \\ \tilde{u}_5 \end{Bmatrix} = \frac{N^3}{L^3} \begin{Bmatrix} -\tilde{u}'(x_0) \\ 0 \\ 0 \\ 0 \\ 0 \\ \tilde{u}'(x_5) \end{Bmatrix}$$

Finally, we have to incorporate the boundary conditions, which essentially require that $\tilde{u}_0 = \tilde{u}_5 = 0$. We change the first row and last rows in the coefficient matrix to reflect these conditions:

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 1+\beta & -2+4\beta & 1+\beta & 0 & 0 & 0 \\ 0 & 1+\beta & -2+4\beta & 1+\beta & 0 & 0 \\ 0 & 0 & 1+\beta & -2+4\beta & 1+\beta & 0 \\ 0 & 0 & 0 & 1+\beta & -2+4\beta & 1+\beta \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{Bmatrix} \tilde{u}_0 \\ \tilde{u}_1 \\ \tilde{u}_2 \\ \tilde{u}_3 \\ \tilde{u}_4 \\ \tilde{u}_5 \end{Bmatrix} = \frac{N^3}{L^3} \begin{Bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{Bmatrix}$$

Further, since the second element in the first column of the coefficient matrix is multiplied by $\tilde{u}_0 = 0$ and the next to last element in the last column of the coefficient matrix is multiplied by $\tilde{u}_5 = 0$, we can replace each of those elements by 0. Further, since the right hand side of this equation is entirely zero, we can remove the factor N^3/L^3. We find that

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & -2+4\beta & 1+\beta & 0 & 0 & 0 \\ 0 & 1+\beta & -2+4\beta & 1+\beta & 0 & 0 \\ 0 & 0 & 1+\beta & -2+4\beta & 1+\beta & 0 \\ 0 & 0 & 0 & 1+\beta & -2+4\beta & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{Bmatrix} \tilde{u}_0 \\ \tilde{u}_1 \\ \tilde{u}_2 \\ \tilde{u}_3 \\ \tilde{u}_4 \\ \tilde{u}_5 \end{Bmatrix} = \begin{Bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{Bmatrix}$$

Recast in the form

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & -2 & 1 & 0 & 0 & 0 \\ 0 & 1 & -2 & 1 & 0 & 0 \\ 0 & 0 & 1 & -2 & 1 & 0 \\ 0 & 0 & 0 & 1 & -2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{Bmatrix} \tilde{u}_0 \\ \tilde{u}_1 \\ \tilde{u}_2 \\ \tilde{u}_3 \\ \tilde{u}_4 \\ \tilde{u}_5 \end{Bmatrix} = -\beta \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 4 & 1 & 0 & 0 & 0 \\ 0 & 1 & 4 & 1 & 0 & 0 \\ 0 & 0 & 1 & 4 & 1 & 0 \\ 0 & 0 & 0 & 1 & 4 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{Bmatrix} \tilde{u}_0 \\ \tilde{u}_1 \\ \tilde{u}_2 \\ \tilde{u}_3 \\ \tilde{u}_4 \\ \tilde{u}_5 \end{Bmatrix}$$

we conclude that the problem is actually a bit simpler, since we already know $\tilde{u}_0$ and $\tilde{u}_5$. We can therefore delete the first and last rows and columns to find the essential problem we must solve to be

$$\begin{bmatrix} 2 & -1 & 0 & 0 \\ -1 & 2 & -1 & 0 \\ 0 & -1 & 2 & -1 \\ 0 & 0 & -1 & 2 \end{bmatrix} \begin{Bmatrix} \tilde{u}_1 \\ \tilde{u}_2 \\ \tilde{u}_3 \\ \tilde{u}_4 \end{Bmatrix} = -\beta \begin{bmatrix} 4 & 1 & 0 & 0 \\ 1 & 4 & 1 & 0 \\ 0 & 1 & 4 & 1 \\ 0 & 0 & 1 & 4 \end{bmatrix} \begin{Bmatrix} \tilde{u}_1 \\ \tilde{u}_2 \\ \tilde{u}_3 \\ \tilde{u}_4 \end{Bmatrix}$$

We can now readily extend this result to apply to an arbitrary number of elements by introducing the quantities[4]

$$A = \begin{pmatrix} -2 & 1 & 0 & \cdots & 0 & 0 & 0 \\ 1 & -2 & 1 & \cdots & 0 & 0 & 0 \\ 0 & 1 & -2 & \cdots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 2 & -1 & 0 \\ 0 & 0 & 0 & \cdots & -1 & 2 & -1 \\ 0 & 0 & 0 & \cdots & 0 & -1 & 2 \end{pmatrix} \quad ; \quad B = \begin{pmatrix} 4 & 1 & 0 & \cdots & 0 & 0 & 0 \\ 1 & 4 & 1 & \cdots & 0 & 0 & 0 \\ 0 & 1 & 4 & \cdots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 4 & 1 & 0 \\ 0 & 0 & 0 & \cdots & 1 & 4 & 1 \\ 0 & 0 & 0 & \cdots & 0 & 1 & 4 \end{pmatrix}$$

and

$$u = \begin{pmatrix} u_1 \\ u_2 \\ u_3 \\ \vdots \\ u_{N-3} \\ u_{N-2} \\ u_{N-1} \end{pmatrix}$$

In terms of these quantities, the problem we must solve is the generalized eigenvalue problem

$$Au = -\beta B\, u = -\frac{k^2 L^2}{6N^2}\, B\, u \Longrightarrow k_n = \sqrt{6\beta_n}\, \frac{N}{L} \Longrightarrow \frac{k_n}{k_1} = \frac{\sqrt{\beta_n}}{\sqrt{\beta_1}}$$

where A and B are matrices, u is a vector of the unknown values at the nodes 1 to $N-1$, and $\beta = k^2L^2/6N^2$ is the eigenvalue to be found. At this point, we hope that the values of β_n are positive. Note that these matrices have dimension $(N-1) \times (N-1)$, even though there are $N+1$ nodes in the discretization on this domain.

[4] We now drop each tilde, understanding that u now stands for the approximate solution.

15.9 1-D Inhomogeneous Helmholtz Equation

Exercise: When α, β, and f are constants, Eqs. (15.67), (15.68), and (15.69) can be solved analytically. Show, for example, that the solution to this boundary value problem is given by

$$\varphi(x) = A\sin\lambda x + \left(p - \frac{f}{\beta}\right)\cos\lambda x + \frac{f}{\beta}$$

where

$$A = \frac{(q - \gamma f/\beta) + (p - f/\beta)(\alpha\lambda\sin\lambda L - \gamma\cos\lambda L)}{\alpha\lambda\cos\lambda L + \gamma\sin\lambda L}$$

and $\lambda = \sqrt{-\beta/\alpha}$ when $\beta/\alpha < 0$. Using graphical displays in particular, compare the analytic solution in this exercise with the solution obtained numerically by finite difference and finite element approaches. *Optional*: Find corresponding solutions when $\beta/\alpha = 0$ and $\beta/\alpha > 0$.

Solution: Assuming that α, β, and f are constants, we can rewrite Eqs. (15.67), (15.68), and (15.69) as

$$-\alpha\frac{d^2\varphi(x)}{dx^2} + \beta\varphi(x) = f \tag{E15.9.1}$$

where α' is zero,

$$\varphi(0) = p \tag{E15.9.2}$$

and

$$\left[\alpha\frac{d\varphi(x)}{dx} + \gamma\varphi(x)\right]\Bigg|_{x=L} = q \tag{E15.9.3}$$

If we divide the differential equation by $-\alpha$, and let $\lambda = \sqrt{-\beta/\alpha}$ or, equivalently, $\lambda^2 = -\beta/\alpha$ (assuming that $\beta/\alpha < 0$), then the differential equation becomes

$$\frac{d^2\varphi(x)}{dx^2} + \lambda^2\varphi(x) = -\frac{f}{\alpha} \tag{E15.9.4}$$

which is an inhomogeneous Helmholtz equation in one dimension.

We know that the general solution to this inhomogeneous equation consists of the sum of a general solution to the homogeneous equation ($f = 0$) and a particular solution which, when substituted into the equation, yields the inhomogeneity. While the independent variable is a space variable rather than time, the homogeneous equation is basically the equation for simple harmonic oscillation and has the general solution

$$\varphi(x) = A\sin(\lambda x) + B\cos(\lambda x) \tag{E15.9.5}$$

We find the particular solution by noting that, if we substitute $\varphi = C$, where C is an as yet undetermined constant, the differential equation Eq. (E15.9.4) becomes

$$\lambda^2 C = -\frac{f}{\alpha} \quad\Longrightarrow\quad C = -\frac{f}{\lambda^2\alpha} = \frac{f}{\beta} \tag{E15.9.6}$$

Thus, having identified the particular solution $\varphi_p = f/\beta$, we conclude that the general solution to the inhomogeneous equation is

$$\varphi(x) = A\sin(\lambda x) + B\cos(\lambda x) + \frac{f}{\beta} \tag{E15.9.7}$$

Now we are left with the task of using the boundary conditions to solve for the constants A and B. We first use Eq. (E15.9.2) to find that

$$\varphi(0) = B + \frac{f}{\beta} = p \quad \text{or} \quad B = p - \frac{f}{\beta} \tag{E15.9.8}$$

Then we use Eq. (E15.9.3) to find A. We substitute the solution for $\varphi(x)$ from Eq. (E15.9.7) to find that

$$\left[\alpha\Big(A\lambda\cos(\lambda x) - B\lambda\sin(\lambda x)\Big) + \gamma\Big(A\sin(\lambda x) + B\cos(\lambda x) + \frac{f}{\beta}\Big)\right]\Bigg|_{x=L} = q \tag{E15.9.9}$$

If we substitute $x = L$, rearrange the equation to solve for A, and substitute the result for B from Eq. (E15.9.8), we find that

$$A = \frac{(q - \gamma f/\beta) + (p - f/\beta)(\alpha\lambda\sin\lambda L - \gamma\cos\lambda L)}{\alpha\lambda\cos\lambda L + \gamma\sin\lambda L} \tag{E15.9.10}$$

The final solution to Eq. (E15.9.1), is

$$\varphi(x) = \frac{(q - \gamma f/\beta) + (p - f/\beta)(\alpha\lambda\sin\lambda L - \gamma\cos\lambda L)}{\alpha\lambda\cos\lambda L + \gamma\sin\lambda L}\sin\lambda x + \left(p - \frac{f}{\beta}\right)\cos\lambda x + \frac{f}{\beta} \tag{E15.9.11}$$

Optional Part

To address the optional part of this exercise, we look next at what happens when $\beta/\alpha = 0$. The differential equation then becomes

$$\frac{d^2\varphi(x)}{dx^2} = -\frac{f}{\alpha} \tag{E15.9.12}$$

and the general solution, which we can obtain simply by integrating the equation twice and inserting appropriate integration constants, is

$$\varphi(x) = -\frac{fx^2}{2\alpha} + Ax + B \tag{E15.9.13}$$

Once again we use the boundary conditions to solve for A and B.

$$\varphi(0) = B = p \tag{E15.9.14}$$

$$\left[\alpha\frac{d\varphi(x)}{dx} + \gamma\varphi(x)\right]\Big|_{x=L} = \alpha\left(-\frac{fL}{\alpha} + A\right) + \gamma\left(-\frac{fL^2}{2\alpha} + AL + B\right) = q \tag{E15.9.15}$$

$$A = \frac{q - p\gamma + fL + \dfrac{fL^2\gamma}{2\alpha}}{\alpha + \gamma L} \tag{E15.9.16}$$

So the final solution for the case $\beta/\alpha = 0$ is

$$\varphi(x) = -\frac{fx^2}{2\alpha} + \frac{q - p\gamma + fL + \dfrac{fL^2\gamma}{2\alpha}}{\alpha + \gamma L}x + p \tag{E15.9.17}$$

Finally, we deduce the solution when $\beta/\alpha > 0$. In this case, we would set $\xi^2 = \beta/\alpha$ and write the differential equation in the form

$$\frac{d^2\varphi(x)}{dx^2} - \xi^2\varphi(x) = -\frac{f}{\alpha} \tag{E15.9.18}$$

The solution to the homogeneous equation this time is a linear combination of the exponentials $e^{\xi x}$ and $e^{-\xi x}$ and a suitable particular solution is still $\phi_p = f/\beta$. Thus, the general solution to this equation is

$$\varphi(x) = Ae^{\xi x} + Be^{-\xi x} + \frac{f}{\beta} \tag{E15.9.19}$$

Once again we use the boundary conditions to find that the constants A and B must satisfy both

$$\varphi(0) = A + B + \frac{f}{\beta} = p \quad \Longrightarrow \quad A + B = p - \frac{f}{\beta} \tag{E15.9.20}$$

and

$$\left[\alpha(A\xi e^{\xi x} - B\xi e^{-\xi x}) + \gamma(Ae^{\xi x} + Be^{-\xi x} + \frac{f}{\beta})\right]\Big|_{x=L} = q$$

$$\Longrightarrow \quad (\gamma + \alpha\xi)e^{\xi L}A + (\gamma - \alpha\xi)e^{-\xi L}B = q - \frac{\gamma f}{\beta} \tag{E15.9.21}$$

Solved for A and B, these two equations, yield the results

$$A = \frac{q - \gamma f/\beta + e^{-\xi L}\Big((\alpha\xi - \gamma)(p - f/\beta)\Big)}{(\alpha\xi + \gamma)e^{\xi L} + (\alpha\xi - \gamma)e^{-\xi L}} \tag{E15.9.22}$$

and

$$B = \frac{-(q - \gamma f/\beta) + e^{\xi L}\Big((\alpha\xi + \gamma)(p - f/\beta)\Big)}{(\alpha\xi + \gamma)e^{\xi L} + (\alpha\xi - \gamma)e^{-\xi L}} \tag{E15.9.23}$$

With these values, the final solution for the case $\beta/\alpha > 0$ is

$$\begin{aligned}\varphi(x) &= \frac{q-\gamma f/\beta + e^{-\xi L}\Big((\alpha\xi-\gamma)(p-f/\beta)\Big)}{(\alpha\xi+\gamma)e^{\xi L}+(\alpha\xi-\gamma)e^{-\xi L}}e^{\xi x} \\ &\quad + \frac{-(q-\gamma f/\beta) + e^{\xi L}\Big((\alpha\xi+\gamma)(p-f/\beta)\Big)}{(\alpha\xi+\gamma)e^{\xi L}+(\alpha\xi-\gamma)e^{-\xi L}}e^{-\xi x} + \frac{f}{\beta} \qquad \text{(E15.9.24)}\end{aligned}$$

15.12 Assembly for Three-Element Linear System

Exercise: Continuing with the circumstances of Exercises 15.10 and 15.11, suppose the region of interest is divided into three three-noded elements, $e = 1, 2, 3$ with global nodes 1, 2, 3, 4, 5, 6, 7, nodes 1, 2, and 3 in element 1, nodes 3, 4, and 5 in element 2, and nodes 5, 6, and 7 in element 3. Suppose node 2 is midway between nodes 1 and 3, node 4 is midway between nodes 3 and 5, and node 6 is midway between nodes 5 and 7, but do not suppose the lengths of the three elements are the same. Following the pattern in Sections 15.9.4 and 15.9.5, assemble the elemental equations for these three elements into an equation for the whole system analogous to the lasst equation in Section (15.9.5) if the solution is required to satisfy the boundary conditions of Eqs. (15.68) and (15.69).

Solution: The specific geometry of the system in this exercise is clarified in Fig. E15.3. The system of interest has three elements and seven distinct nodes.

In Exercise 15.11, we found that the elemental equation applicable to a three-noded linear element (e) is

$$\begin{bmatrix} K_{11}^{(e)} & K_{12}^{(e)} & K_{13}^{(e)} \\ K_{21}^{(e)} & K_{22}^{(e)} & K_{23}^{(e)} \\ K_{31}^{(e)} & K_{32}^{(e)} & K_{33}^{(e)} \end{bmatrix} \begin{Bmatrix} \tilde{\varphi}_1^{(e)} \\ \tilde{\varphi}_2^{(e)} \\ \tilde{\varphi}_3^{(e)} \end{Bmatrix} = \begin{Bmatrix} b_1^{(e)} \\ b_2^{(e)} \\ b_3^{(e)} \end{Bmatrix} + \begin{Bmatrix} g_1^{(e)} \\ 0 \\ g_3^{(e)} \end{Bmatrix}$$

For the system shown in Fig. E15.3, we want to assemble the elemental equations for each element into an equation for the whole system if the solution is required to satisfy the boundary conditions

$$\varphi(x)|_{x=0} = p \quad ; \quad \left[\alpha(x)\frac{d\varphi(x)}{dx} + \gamma\varphi(x)\right]\Big|_{x=L} = q$$

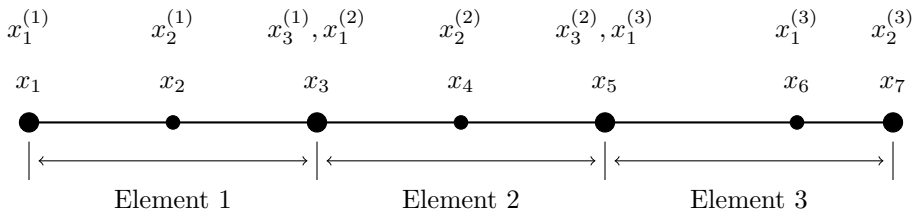
We write the elemental equation for the first element as

$$\begin{bmatrix} K_{11}^{(1)} & K_{12}^{(1)} & K_{13}^{(1)} \\ K_{21}^{(1)} & K_{22}^{(1)} & K_{23}^{(1)} \\ K_{31}^{(1)} & K_{32}^{(1)} & K_{33}^{(1)} \end{bmatrix} \begin{Bmatrix} \tilde{\varphi}_1^{(1)} \\ \tilde{\varphi}_2^{(1)} \\ \tilde{\varphi}_3^{(1)} \end{Bmatrix} = \begin{Bmatrix} b_1^{(1)} \\ b_2^{(1)} \\ b_3^{(1)} \end{Bmatrix} + \begin{Bmatrix} g_1^{(1)} \\ 0 \\ g_3^{(1)} \end{Bmatrix}$$

or equivalently,

$$\begin{aligned} K_{11}^{(1)}\tilde{\varphi}_1^{(1)} + K_{12}^{(1)}\tilde{\varphi}_2^{(1)} + K_{13}^{(1)}\tilde{\varphi}_3^{(1)} &= b_1^{(1)} + g_1^{(1)} \\ K_{21}^{(1)}\tilde{\varphi}_1^{(1)} + K_{22}^{(1)}\tilde{\varphi}_2^{(1)} + K_{23}^{(1)}\tilde{\varphi}_3^{(1)} &= b_2^{(1)} \\ K_{31}^{(1)}\tilde{\varphi}_1^{(1)} + K_{32}^{(1)}\tilde{\varphi}_2^{(1)} + K_{33}^{(1)}\tilde{\varphi}_3^{(1)} &= b_3^{(1)} + g_3^{(1)} \end{aligned}$$

Figure E15.3: Figure for Exercise 15.12.



Similarly, we can write out the equations for the remaining two elements, finding that

$$\begin{aligned} K_{11}^{(2)}\tilde{\varphi}_1^{(2)} + K_{12}^{(2)}\tilde{\varphi}_2^{(2)} + K_{13}^{(2)}\tilde{\varphi}_3^{(2)} &= b_1^{(2)} + g_1^{(2)} \\ K_{21}^{(2)}\tilde{\varphi}_1^{(2)} + K_{22}^{(2)}\tilde{\varphi}_2^{(2)} + K_{23}^{(2)}\tilde{\varphi}_3^{(2)} &= b_2^{(2)} \\ K_{31}^{(2)}\tilde{\varphi}_1^{(2)} + K_{32}^{(2)}\tilde{\varphi}_2^{(2)} + K_{33}^{(2)}\tilde{\varphi}_3^{(2)} &= b_3^{(2)} + g_3^{(2)} \end{aligned}$$

and

$$\begin{aligned} K_{11}^{(3)}\tilde{\varphi}_1^{(3)} + K_{12}^{(3)}\tilde{\varphi}_2^{(3)} + K_{13}^{(3)}\tilde{\varphi}_3^{(3)} &= b_1^{(3)} + g_1^{(3)} \\ K_{21}^{(3)}\tilde{\varphi}_1^{(3)} + K_{22}^{(3)}\tilde{\varphi}_2^{(3)} + K_{23}^{(3)}\tilde{\varphi}_3^{(3)} &= b_2^{(3)} \\ K_{31}^{(3)}\tilde{\varphi}_1^{(3)} + K_{32}^{(3)}\tilde{\varphi}_2^{(3)} + K_{33}^{(3)}\tilde{\varphi}_3^{(3)} &= b_3^{(3)} + g_3^{(3)} \end{aligned}$$

For this region, the relationship between the local and global numbering systems is defined as

$$x_1^{(e)} = x_{2e-1} \quad ; \quad x_2^{(e)} = x_{2e} \quad ; \quad x_3^{(e)} = x_{2e+1} \quad ; \quad e = 1, 2, 3$$

The condition of continuity at nodes 2 and 3 then requires that $\tilde{\varphi}_3^{(1)} = \tilde{\varphi}_1^{(2)}$ and $\tilde{\varphi}_3^{(2)} = \tilde{\varphi}_1^{(3)}$. Further, as evidenced in Fig. E15.3,

$$\tilde{\varphi}_1^{(1)} = \tilde{\varphi}_1 \quad ; \quad \tilde{\varphi}_2^{(1)} = \tilde{\varphi}_2 \quad ; \quad \tilde{\varphi}_3^{(1)} = \tilde{\varphi}_1^{(2)} = \tilde{\varphi}_3 \quad ; \quad \tilde{\varphi}_2^{(2)} = \tilde{\varphi}_4$$
$$\tilde{\varphi}_3^{(2)} = \tilde{\varphi}_1^{(3)} = \tilde{\varphi}_5 \quad ; \quad \tilde{\varphi}_2^{(3)} = \tilde{\varphi}_6 \quad ; \quad \tilde{\varphi}_3^{(3)} = \tilde{\varphi}_7$$

Thus, we rewrite the individual elemental equations in the form

$$\begin{aligned} K_{11}^{(1)}\tilde{\varphi}_1 + K_{12}^{(1)}\tilde{\varphi}_2 + K_{13}^{(1)}\tilde{\varphi}_3 &= b_1^{(1)} + g_1^{(1)} \\ K_{21}^{(1)}\tilde{\varphi}_1 + K_{22}^{(1)}\tilde{\varphi}_2 + K_{23}^{(1)}\tilde{\varphi}_3 &= b_2^{(1)} \\ K_{31}^{(1)}\tilde{\varphi}_1 + K_{32}^{(1)}\tilde{\varphi}_2 + K_{33}^{(1)}\tilde{\varphi}_3 &= b_3^{(1)} + g_3^{(1)} \\ K_{11}^{(2)}\tilde{\varphi}_3 + K_{12}^{(2)}\tilde{\varphi}_4 + K_{13}^{(2)}\tilde{\varphi}_5 &= b_1^{(2)} + g_1^{(2)} \\ K_{21}^{(2)}\tilde{\varphi}_3 + K_{22}^{(2)}\tilde{\varphi}_4 + K_{23}^{(2)}\tilde{\varphi}_5 &= b_2^{(2)} \\ K_{31}^{(2)}\tilde{\varphi}_3 + K_{32}^{(2)}\tilde{\varphi}_4 + K_{33}^{(2)}\tilde{\varphi}_5 &= b_3^{(2)} + g_3^{(2)} \\ K_{11}^{(3)}\tilde{\varphi}_5 + K_{12}^{(3)}\tilde{\varphi}_6 + K_{13}^{(3)}\tilde{\varphi}_7 &= b_1^{(3)} + g_1^{(3)} \\ K_{21}^{(3)}\tilde{\varphi}_5 + K_{22}^{(3)}\tilde{\varphi}_6 + K_{23}^{(3)}\tilde{\varphi}_7 &= b_2^{(3)} \\ K_{31}^{(3)}\tilde{\varphi}_5 + K_{32}^{(3)}\tilde{\varphi}_6 + K_{33}^{(3)}\tilde{\varphi}_7 &= b_3^{(3)} + g_3^{(3)} \end{aligned}$$

Unfortunately, we now have nine equations but only seven unknowns. We know that some of these equations must be redundant, and we therefore need to reduce the number of equations. In order to do this, we can replace the third and fourth equations by their sum, as well as the sixth and seventh equations by their sum. We then find that

$$\begin{bmatrix} K_{11}^{(1)} & K_{12}^{(1)} & K_{13}^{(1)} & 0 & 0 & 0 & 0 \\ K_{21}^{(1)} & K_{22}^{(1)} & K_{23}^{(1)} & 0 & 0 & 0 & 0 \\ K_{31}^{(1)} & K_{32}^{(1)} & K_{33}^{(1)} + K_{11}^{(2)} & K_{12}^{(2)} & K_{13}^{(2)} & 0 & 0 \\ 0 & 0 & K_{21}^{(2)} & K_{22}^{(2)} & K_{23}^{(2)} & 0 & 0 \\ 0 & 0 & K_{31}^{(2)} & K_{32}^{(2)} & K_{33}^{(2)} + K_{11}^{(3)} & K_{12}^{(3)} & K_{13}^{(3)} \\ 0 & 0 & 0 & 0 & K_{21}^{(3)} & K_{22}^{(3)} & K_{23}^{(3)} \\ 0 & 0 & 0 & 0 & K_{31}^{(3)} & K_{32}^{(3)} & K_{33}^{(3)} \end{bmatrix} \begin{Bmatrix} \tilde{\varphi}_1 \\ \tilde{\varphi}_2 \\ \tilde{\varphi}_3 \\ \tilde{\varphi}_4 \\ \tilde{\varphi}_5 \\ \tilde{\varphi}_6 \\ \tilde{\varphi}_7 \end{Bmatrix} = \begin{Bmatrix} b_1^{(1)} \\ b_2^{(1)} \\ b_3^{(1)} + b_1^{(2)} \\ b_2^{(2)} \\ b_3^{(2)} + b_1^{(3)} \\ b_2^{(3)} \\ b_3^{(3)} \end{Bmatrix} + \begin{Bmatrix} g_1^{(1)} \\ g_2^{(1)} \\ g_3^{(1)} + g_1^{(2)} \\ g_2^{(2)} \\ g_3^{(2)} + g_1^{(3)} \\ g_2^{(3)} \\ g_3^{(3)} \end{Bmatrix}$$

In Exercise 15.11, we found that

$$\begin{aligned} g_1^{(e)} &= -\alpha(x_1^{(e)})\frac{d\tilde{\varphi}^{(e)}}{dx}\bigg|_{x_1^{(e)}} \\ g_2^{(e)} &= 0 \\ g_3^{(e)} &= \alpha(x_3^{(e)})\frac{d\tilde{\varphi}^{(e)}}{dx}\bigg|_{x_3^{(e)}} \end{aligned}$$

With these values, we conclude also that

$$g_3^{(1)} + g_1^{(2)} = \alpha(x_3^{(1)})\frac{d\tilde{\varphi}^{(1)}}{dx}\bigg|_{x_3^{(1)}} - \alpha(x_1^{(2)})\frac{d\tilde{\varphi}^{(2)}}{dx}\bigg|_{x_1^{(2)}} = 0$$

and

$$g_3^{(2)} + g_1^{(3)} = \alpha(x_3^{(2)})\frac{d\tilde{\varphi}^{(2)}}{dx}\bigg|_{x_3^{(2)}} - \alpha(x_1^{(3)})\frac{d\tilde{\varphi}^{(3)}}{dx}\bigg|_{x_1^{(3)}} = 0$$

because $x_3^{(1)} = x_1^{(2)} = x_3$, $x_3^{(2)} = x_1^{(3)} = x_5$, and the derivative of $\tilde{\varphi}$ must be continuous across the boundary between elements. Hence, all elements except the first and last in $\{g\}$ are zero, and our system of equations now becomes

$$\begin{bmatrix} K_{11}^{(1)} & K_{12}^{(1)} & K_{13}^{(1)} & 0 & 0 & 0 & 0 \\ K_{21}^{(1)} & K_{22}^{(1)} & K_{23}^{(1)} & 0 & 0 & 0 & 0 \\ K_{31}^{(1)} & K_{32}^{(1)} & K_{33}^{(1)} + K_{11}^{(2)} & K_{12}^{(2)} & K_{13}^{(2)} & 0 & 0 \\ 0 & 0 & K_{21}^{(2)} & K_{22}^{(2)} & K_{23}^{(2)} & 0 & 0 \\ 0 & 0 & K_{31}^{(2)} & K_{32}^{(2)} & K_{33}^{(2)} + K_{11}^{(3)} & K_{12}^{(3)} & K_{13}^{(3)} \\ 0 & 0 & 0 & 0 & K_{21}^{(3)} & K_{22}^{(3)} & K_{23}^{(3)} \\ 0 & 0 & 0 & 0 & K_{31}^{(3)} & K_{32}^{(3)} & K_{33}^{(3)} \end{bmatrix} \begin{Bmatrix} \tilde{\varphi}_1 \\ \tilde{\varphi}_2 \\ \tilde{\varphi}_3 \\ \tilde{\varphi}_4 \\ \tilde{\varphi}_5 \\ \tilde{\varphi}_6 \\ \tilde{\varphi}_7 \end{Bmatrix} = \begin{Bmatrix} b_1^{(1)} \\ b_2^{(1)} \\ b_3^{(1)} + b_1^{(2)} \\ b_2^{(2)} \\ b_3^{(2)} + b_1^{(3)} \\ b_2^{(3)} \\ b_3^{(3)} \end{Bmatrix} + \begin{Bmatrix} g_1^{(1)} \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ g_3^{(3)} \end{Bmatrix}$$

More explicit values for all entries in this equation except the unknowns in the vector $\{\tilde{\varphi}\}$ can be inferred from the results in Exercise 15.11, though those values apply only if the interior nodes in each element are halfway between the end nodes of that element.

15.20 FDM 2D Inhomogeneous Helmholtz Equation

Exercise: The inhomogeneous Helmholtz equation in two-dimensional Cartesian coordinates is

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} + k^2 u = r(x,y)$$

where k^2 is a constant and $r(x,y)$ is the inhomogeneity. Apply finite difference methods to show that

$$u_{i,j} \approx \frac{u_{i+1,j} + u_{i-1,j} + u_{i,j+1} + u_{i,j-1} - \Delta x^2\, r_{i,j}}{4 - k^2\, \Delta x^2}$$

Here, $u_{i,j} = u(x_i, y_j)$, the spacing between consecutive values of x is Δx, the spacing between consecutive values of y is Δy and $\Delta y = \Delta x$. This result could be used in an iterative approach to solving the inhomogeneous Helmholtz equation. *Note*: If $k^2 = 0$, the equation of this exercise reduces to the inhomogeneous Laplace equation, i.e., to the Poisson equation. If, on the other hand, $r(x,y) = 0$, then this equation reduces to the (homogeneous) Helmholtz equation.

Solution: Paralleling our argument on Eq. (15.75) in the text, we infer that

$$\begin{aligned}
\left.\frac{d\varphi^2(x,y)}{dx^2}\right|_{x_i,y_j} &\approx \frac{\left.\dfrac{d\varphi(x,y)}{dx}\right|_{x_i+\Delta x/2,y_j} - \left.\dfrac{d\varphi(x,y)}{dx}\right|_{x_i-\Delta x/2,y_j}}{\Delta x} \\
&\approx \frac{\dfrac{\varphi_{i+1,j}-\varphi_{i,j}}{\Delta x} - \dfrac{\varphi_{i,j}-\varphi_{i-1,j}}{\Delta x}}{\Delta x} \\
&\approx \frac{\varphi_{i+1,j} - 2\varphi_{i,j} + \varphi_{i-1,j}}{\Delta x^2}
\end{aligned}$$

Similarly, we conclude that

$$\left.\frac{d\varphi^2(x,y)}{dy^2}\right|_{x_i,y_j} \approx \frac{\varphi_{i,j+1} - 2\varphi_{i,j} + \varphi_{i,j-1}}{\Delta x^2}$$

Finally, evaluating the original differential equation at (x_i, y_j), we conclude that

$$\frac{\varphi_{i+1,j} - 2\varphi_{i,j} + \varphi_{i-1,j}}{\Delta x^2} + \frac{\varphi_{i,j+1} - 2\varphi_{i,j} + \varphi_{i,j-1}}{\Delta x^2} + k^2 u_{i,j} = r_{i,j}$$

which, on solution for $u_{i,j}$ yields that

$$u_{i,j} \approx \frac{u_{i+1,j} + u_{i-1,j} + u_{i,j+1} + u_{i,j-1} - \Delta x^2\, r_{i,j}}{4 - k^2\, \Delta x^2}$$

Q.E.D.

15.25 Finite Difference Formulas via Taylor Series

Exercise: (a) Starting with the Taylor series

$$\tilde{\varphi}(x+\Delta x) = \tilde{\varphi}(x) + \Delta x\,\tilde{\varphi}'(x) + O(\Delta x^2)$$

evaluate

$$\tilde{\varphi}(x+\Delta x) - \tilde{\varphi}(x)$$

to show that

$$\tilde{\varphi}'(x) = \frac{\tilde{\varphi}(x+\Delta x) - \tilde{\varphi}(x)}{\Delta x} + O(\Delta x)$$

That is, derive the forward difference formula in Eq. (15.74). (b) Deduce the more accurate forward difference formula

$$\tilde{\varphi}'(x) \approx \frac{-\tilde{\varphi}(x+2\,\Delta x) + 4\tilde{\varphi}(x+\Delta x) - 3\tilde{\varphi}(x)}{2\,\Delta x} + O(\Delta x^2)$$

Hint: Start by using the Taylor expansion

$$f(x+h) = f(x) + h\,f'(x) + \frac{1}{2}\,h^2\,f''(x) + O(h^3)$$

to expand $\tilde{\varphi}(x+\Delta x)$ and $\tilde{\varphi}(x+2\,\Delta x)$.

Solution: (a) This first part involves little more than a rearrangement of the terms in the Taylor expansion provided in part (a), first to

$$\tilde{\varphi}(x+\Delta x) - \tilde{\varphi}(x) = \Delta x\,\tilde{\varphi}'(x) + O(\Delta x^2)$$

and then to

$$\frac{\tilde{\varphi}(x+\Delta x) - \tilde{\varphi}(x)}{\Delta x} = \tilde{\varphi}'(x) + \frac{O(\Delta x^2)}{\Delta x}$$

and finally to[5]

$$\tilde{\varphi}'(x) = \frac{\tilde{\varphi}(x+\Delta x) - \tilde{\varphi}(x)}{\Delta x} + O(\Delta x)$$

Introducing $x = x_i$ and $x + \Delta x = x_{i+1}$ in the above Taylor expansion to find that

$$\left.\frac{d\tilde{\varphi}(x)}{dx}\right|_{x_i} = \frac{\tilde{\varphi}_{x_{i+1}} - \tilde{\varphi}_i}{\Delta x} + O(\Delta x)$$

(b) Based on the Taylor series provided in part (b), we note that

$$\tilde{\varphi}(x+\Delta x) = \tilde{\varphi}(x) + \Delta x\,\tilde{\varphi}'(x) + \frac{1}{2}\,\Delta x^2\,\tilde{\varphi}''(x) + O(\Delta x^3)$$

and that

$$\tilde{\varphi}(x+2\,\Delta x) = \tilde{\varphi}(x) + 2\,\Delta x\,\tilde{\varphi}'(x) + \frac{1}{2}\,(2\,\Delta x)^2\,\tilde{\varphi}''(x) + O(\Delta x^3)$$

Solving the first of these equations for $\tilde{\varphi}''(x)\,\Delta x^2$ yields

$$\tilde{\varphi}''(x)\,\Delta x^2 = 2\,\tilde{\varphi}(x+\Delta x) - 2\,\tilde{\varphi}(x) - 2\,\Delta x\,\tilde{\varphi}'(x) + O(\Delta x^3)$$

Substituting this result into the second equation then yields that

$$\tilde{\varphi}(x+2\,\Delta x) = \tilde{\varphi}(x) + 2\,\Delta x\,\tilde{\varphi}'(x) + 2\Big(2\,\tilde{\varphi}(x+\Delta x) - 2\,\tilde{\varphi}(x) - 2\,\Delta x\,\tilde{\varphi}'(x)\Big) + O(\Delta x^3)$$

[5] We can retain the plus sign before $O(\Delta x)$ because it is of no consequence in specifying the magnitude of the potential error.

Combining those terms that can be combined, we learn that

$$\tilde{\varphi}(x + 2\,\Delta x) = -3\,\tilde{\varphi}(x) - 2\,\Delta x\,\tilde{\varphi}'(x) + 4\,\tilde{\varphi}(x + \Delta x) + O(\Delta x^3)$$

Finally, solving for $\tilde{\varphi}'(x)$, we find that

$$\tilde{\varphi}'(x) = \frac{-3\,\tilde{\varphi}(x) + 4\,\tilde{\varphi}(x + \Delta x) - \tilde{\varphi}(x + 2\,\Delta x)}{2\,\Delta x} + O(\Delta x^2)$$

With $x = x_i$, $x + \Delta x = x_{i+1}$, and $x + 2\,\Delta x = x_{i+2}$, this result has the expression

$$\left.\frac{d\tilde{\varphi}(x)}{dx}\right|_{x_i} = \frac{-3\,\tilde{\varphi}_i + 4\,\tilde{\varphi}_{i+1} - \tilde{\varphi}_{i+2}}{2\,\Delta x} + O(\Delta x^2)$$

15.26 Shape Functions for Two-Node Linear Element

Exercise: Starting with the first two numbered equations in Section 15.9.2, work through the details in that section to derive the results

$$N_1^{(e)}(x) = \frac{x_2^{(e)} - x}{l^{(e)}} \quad ; \quad N_2^{(e)}(x) = \frac{x - x_1^{(e)}}{l^{(e)}}$$

for the two interpolation functions. Here, $l^{(e)} = x_2^{(e)} - x_1^{(e)}$.

Solution: We begin by presuming that the solution in the interval $x_1^{(e)} \leq x \leq x_2^{(e)}$ can be approximated with the linear function

$$\tilde{\varphi}^{(e)} = a^{(e)} + b^{(e)}\, x$$

We require that this function yield the known values at the ends of the interval, i.e., that

$$\begin{aligned} \tilde{\varphi}_1^{(e)} &= a^{(e)} + b^{(e)}\, x_1^{(e)} \\ \tilde{\varphi}_2^{(e)} &= a^{(e)} + b^{(e)}\, x_2^{(e)} \end{aligned}$$

Subtracting the first of these equations from the second yields that

$$\tilde{\varphi}_2^{(e)} - \tilde{\varphi}_1^{(e)} = b^{(e)}(x_2^{(e)} - x_1^{(e)}) \quad \Longrightarrow \quad b^{(e)} = \frac{\tilde{\varphi}_2^{(e)} - \tilde{\varphi}_1^{(e)}}{x_2^{(e)} - x_1^{(e)}} = \frac{\tilde{\varphi}_2^{(e)} - \tilde{\varphi}_1^{(e)}}{l^{(e)}}$$

Then, the first equation yields that

$$\tilde{\varphi}_1^{(e)} = a^{(e)} + \left(\frac{\tilde{\varphi}_2^{(e)} - \tilde{\varphi}_1^{(e)}}{x_2^{(e)} - x_1^{(e)}} \right) x_1^{(e)} \quad \Longrightarrow \quad a^{(e)} = \tilde{\varphi}_1^{(e)} - \left(\frac{\tilde{\varphi}_2^{(e)} - \tilde{\varphi}_1^{(e)}}{x_2^{(e)} - x_1^{(e)}} \right) x_1^{(e)}$$

$$\Longrightarrow \quad a^{(e)} = \frac{\tilde{\varphi}_1^{(e)} x_2^{(e)} - \tilde{\varphi}_2^{(e)} x_1^{(e)}}{x_2^{(e)} - x_1^{(e)}} = \frac{\tilde{\varphi}_1^{(e)} x_2^{(e)} - \tilde{\varphi}_2^{(e)} x_1^{(e)}}{l^{(e)}}$$

Substituting these results for the two constants into the first equation in this solution, we learn that

$$\tilde{\varphi}^{(e)} = \frac{\tilde{\varphi}_1^{(e)} x_2^{(e)} - \tilde{\varphi}_2^{(e)} x_1^{(e)}}{l^{(e)}} + \left(\frac{\tilde{\varphi}_2^{(e)} - \tilde{\varphi}_1^{(e)}}{l^{(e)}} \right) x = \tilde{\varphi}_1^{(e)} \left(\frac{x_2^{(e)} - x}{l^{(e)}} \right) + \tilde{\varphi}_2^{(e)} \left(\frac{x - x_1^{(e)}}{l^{(e)}} \right)$$

from which we learn, finally, that the interpolation functions are

$$N_1^{(e)}(x) = \frac{x_2^{(e)} - x}{l^{(e)}} \quad ; \quad N_2^{(e)}(x) = \frac{x - x_1^{(e)}}{l^{(e)}}$$

15.6 Better Approximation to First Derivative (MAXIMA)

Exercise: Suppose you know three points (x_0, f_0), $(x_1 = x_0 + \Delta x, f_1)$, and $(x_2 = x_0 + 2\,\Delta x, f_2)$ on a curve and you wish to estimate the derivative of the corresponding function at $x = x_0$ by a method that is more accurate than simply the forward difference formula $(f_1 - f_0)/\Delta x$. You might fit the parabola $f(x) = Ax^2 + Bx + C$ through the three points and then approximate the derivative of the actual function as the derivative of this parabola at the point $x = x_0$. Show that this approach yields the formula

$$\left.\frac{df}{dx}\right|_{x=x_0} = \frac{-3f_0 + 4f_1 - f_2}{2\,\Delta x}$$

which is a higher-order forward difference approximation than the one used in the text. *Hint*: Set up the three equations $f_i = Ax_i^2 + Bx_i + C$, solve those equations for A, B, and C, and note that $df/dx = 2Ax + B$. You may find a symbolic manipulating program of substantial assistance.

Solution: To fit a parabola to the given three points, we would require that

$$\begin{aligned} Ax_0^2 + Bx_0 + C &= f_0 \\ A(x_0 + \Delta x)^2 + B(x_0 + \Delta x) + C &= f_1 \\ A(x_0 + 2\Delta x)^2 + B(x_0 + 2\Delta x) + C &= f_2 \end{aligned}$$

Rather than solving these equations for A, b, and C by hand, we elect to invoke MAXIMA with the statements

```
eq0 : A*x0^2 + B*x0 + C = f0$                  /* Define equations,     */
eq1 : A*(x0+dx)^2 + B*(x0+dx) + C = f1$        /*    suppressing output */
eq2 : A*(x0+2*dx)^2 + B*(x0+2*dx) + C = f2$
soln : solve([eq0,eq1,eq2], [A,B,C]);          /* Solve equations       */
```

Translated into a more readable form, the output for `soln` is

$$\begin{aligned} A &= \frac{f_2 - 2f_1 + f_0}{2\,\Delta x^2} \\ B &= -\frac{(2f_2 - 4f_1 + 2f_0)\,x_0 + \Delta x\,f_2 - 4\,\Delta x\,f_1 + 3\,\Delta x\,f_0}{2\,\Delta x^2} \\ C &= \frac{(f_2 - 2f_1 + f_0)\,x_0^2 + (\Delta x\,f_2 - 4\,\Delta x\,f_1 + 3\,\Delta x\,f_0)\,x_0 + 2\,\Delta x^2\,f_0}{2\,\Delta x^2} \end{aligned}$$

This result is a bit of a mess. The derivative we need, however, is a bit simpler and can be evaluated with the statement

```
deriv : ev(2*A*x0+B, soln);
```

yielding

$$\frac{(f_2 - 2f_1 + f_0)\,x_0}{\Delta x^2} - \frac{(2f_2 - 4f_1 + 2f_0)\,x_0 + \Delta x\,f_2 - 4\,\Delta x\,f_1 + 3\,\Delta x\,f_0}{2\,\Delta x^2}$$

Clearly, the first term in this expression is exactly cancelled by the first part of the second term. Persuading MAXIMA to recognize that fact, however, requires that we temporarily remove the denominators by multiplying the expression by $2\,\Delta x^2$ with the statement

```
dx2deriv : expand(2*dx^2*deriv);
```

yielding

$$-\Delta\, x\, f_2 + 4\,\Delta x\, f_1 - 3\,\Delta x\, f_0$$

Finally, we undo the multiplication by $2\,\Delta x^2$ with the statement

```
ans : collectterms(dx2deriv,dx)/(2*dx^2);
```

which yields the result that the first derivative of $f(x)$ at the point $x = x_0$ can be estimated by the forward difference formula

$$\left.\frac{df(x)}{dx}\right|_{x_0} \approx \frac{-f_2 + 4f_1 - 3f_0}{2\,\Delta x}$$

Q.E.D.

15.10 Shape Functions for Three-Node Linear Element (MAXIMA)

Exercise: Suppose a linear element e is characterized by *three* nodes at $x_1^{(e)}$, $x_2^{(e)}$, and $x_3^{(e)}$. Further, let the solution $\tilde{\varphi}^{(e)}$ on that element be approximated by the quadratic function

$$\tilde{\varphi}^{(e)} = a^{(e)} + b^{(e)}x + c^{(e)}x^2$$

(a) Find the constants $a^{(e)}$, $b^{(e)}$, and $c^{(e)}$ that will make this function match the specific values $\tilde{\varphi}_1^{(e)}$, $\tilde{\varphi}_2^{(e)}$, and $\tilde{\varphi}_3^{(e)}$ at the points $x = x_1^{(e)}$, $x_2^{(e)}$, and $x_3^{(e)}$, respectively. (b) Substituting these values into the above expression and grouping terms appropriately, cast the result in the form

$$\tilde{\varphi}^{(e)} = \sum_{i=1}^{3} \tilde{\varphi}_i^{(e)}\, N_i^{(e)}(x)$$

and show that the shape functions $N_i^{(e)}(x)$ appropriate to this three-noded linear element are

$$N_1^{(e)}(x) = \frac{1}{\Delta}\begin{vmatrix} 1 & x & x^2 \\ 1 & x_2^{(e)} & (x_2^{(e)})^2 \\ 1 & x_3^{(e)} & (x_3^{(e)})^2 \end{vmatrix} = \frac{(x_2^{(e)} - x)(x_3^{(e)} - x)}{(x_2^{(e)} - x_1^{(e)})(x_3^{(e)} - x_1^{(e)})}$$

$$N_2^{(e)}(x) = \frac{1}{\Delta}\begin{vmatrix} 1 & x_1^{(e)} & (x_1^{(e)})^2 \\ 1 & x & x^2 \\ 1 & x_3^{(e)} & (x_3^{(e)})^2 \end{vmatrix} = \frac{(x - x_1^{(e)})(x_3^{(e)} - x)}{(x_2^{(e)} - x_1^{(e)})(x_3^{(e)} - x_2^{(e)})}$$

$$N_3^{(e)}(x) = \frac{1}{\Delta}\begin{vmatrix} 1 & x_1^{(e)} & (x_1^{(e)})^2 \\ 1 & x_2^{(e)} & (x_2^{(e)})^2 \\ 1 & x & x^2 \end{vmatrix} = \frac{(x - x_1^{(e)})(x - x_2^{(e)})}{(x_3^{(e)} - x_1^{(e)})(x_3^{(e)} - x_2^{(e)})}$$

where

$$\Delta = \begin{vmatrix} 1 & x_1^{(e)} & (x_1^{(e)})^2 \\ 1 & x_2^{(e)} & (x_2^{(e)})^2 \\ 1 & x_3^{(e)} & (x_3^{(e)})^2 \end{vmatrix}$$

Finally, (c) show that—with $\xi = (x - x_1^{(e)})/l^{(e)}$—these functions reduce to

$$N_1^{(e)}(x) = 2(\xi - 1)\left(\xi - \frac{1}{2}\right) \quad ; \quad N_2^{(e)}(x) = 4\xi(1 - \xi) \quad ; \quad N_3^{(e)}(x) = 2\xi\left(\xi - \frac{1}{2}\right)$$

when $x_2^{(e)}$ is midway between $x_1^{(e)}$ and $x_3^{(e)}$, i.e., when $x_2^{(e)} = x_1^{(e)} + \frac{1}{2}l^{(e)}$ and $x_3^{(e)} = x_1^{(e)} + l^{(e)}$, and (d) obtain graphs of these three shape functions over the interval $0 < \xi < 1$. *Hint*: You may find a symbolic manipulating program to be useful at many points in this problem.

Solution: This exercise follows the same procedure as is laid out in the text for the finite element method in one dimension. The only difference here is that the element is characterized by three nodes rather than just two, and we therefore approximate the solution, $\tilde{\varphi}^{(e)}$, on this element with a quadratic function

$$\tilde{\varphi}^{(e)} = a^{(e)} + b^{(e)}x + c^{(e)}x^2 \tag{E15.10.1}$$

rather than a linear function.

(a) We want to find the constants[6] a, b, and c that will make this function match the specific values $\tilde{\varphi}_1$, $\tilde{\varphi}_2$, and $\tilde{\varphi}_3$ at the points $x = x_1$, x_2, and x_3, respectively. We need to solve the equations

$$\tilde{\varphi}_1 = a + bx_1 + cx_1^2 \quad ; \quad \tilde{\varphi}_2 = a + bx_2 + cx_2^2 \quad ; \quad \tilde{\varphi}_3 = a + bx_3 + cx_3^2$$

Using MAXIMA, we create these equation and solve them with the statements

[6] To simplify the expressions, we will suppress the superscripts (e), though we can reinstate them at the end.

```
eq1 : phi1 = A + B*x1 + C*x1^2 $
eq2 : phi2 = A + B*x2 + C*x2^2 $
eq3 : phi3 = A + B*x3 + C*x3^2 $
soln : solve( [eq1, eq2, eq3], [A, B, C] );
```

(soln) $$\left[\left[A = \frac{A_n}{D}, B = \frac{B_n}{D}, C = \frac{C_n}{D}\right]\right]$$

where we have indicated only the form of the solution. The several numerators A_n, B_n and C_n and the common denominator D can be extracted from this complicated result with the statements

```
An : num( ev(A,soln) );
```

(An) $\tilde{\varphi}_1(x_2^2 x_3 - x_2 x_3^2) + x_1(\tilde{\varphi}_2 x_3^2 - \tilde{\varphi}_3 x_2^2) + x_1^2(\tilde{\varphi}_3 x_2 - \tilde{\varphi}_2 x_3)$

```
Bn : num( ev(B,soln) );
```

(Bn) $\tilde{\varphi}_1(x_3^2 - x_2^2) - \tilde{\varphi}_2 x_3^2 + \tilde{\varphi}_3 x_2^2 + (\tilde{\varphi}_2 - \tilde{\varphi}_3)x_1^2$

```
Cn : num( ev(C,soln) );
```

(Cn) $-\tilde{\varphi}_1(x_3 - x_2) + \tilde{\varphi}_2 x_3 - \tilde{\varphi}_3 x_2 - (\tilde{\varphi}_2 - \tilde{\varphi}_3)x_1$

```
D : denom( ev(A,soln) );
```

(D) $x_1(x_3^2 - x_2^2) - x_2 x_3^2 + x_2^2 x_3 + x_1^2(x_2 - x_3)$

```
DD : factor(expand(D));
```

(DD) $-(x_2 - x_1)(x_3 - x_1)(x_3 - x_2)$

(b) With these results, the approximating function Eq. (E15.10.1) for element e is given by the expression

$$\tilde{\varphi} = a + bx + cx^2 = \frac{a_n + b_n\, x + c_n\, x^2}{d}$$

The *numerator* in this last expression is given and then the coefficient of each $\tilde{\varphi}_i$ in that numerator collected by the statements

```
phin : An + Bn*x + Cn*x^2$
phin : collectterms( expand(phin), phi1, phi2, phi3);
```

(phin) $\tilde{\varphi}_1\, n_1 + \tilde{\varphi}_2\, n_2 + \tilde{\varphi}_3\, n_3$

where

```
 n1 : factor( part(phin,1,2) );
```

(n1) $-(x_2 - x)(x_3 - x)(x_3 - x_2)$

```
 n2 : factor( part(phin,2,2) );
```

(n2) $(x_1 - x)(x_3 - x)(x_3 - x_1)$

```
 n3 : factor( part(phin,3,2) );
```

(n3) $-(x_1 - x)(x_2 - x)(x_2 - x_1)$

With these results, the shape functions themselves are given by the statements

```
N1 : n1/DD;
```

(N1) $$N_1^{(e)}(x) = \frac{(x_2^{(e)} - x)(x_3^{(e)} - x)}{(x_2^{(e)} - x_1^{(e)})(x_3^{(e)} - x_1^{(e)})}$$

```
N2 : n2/DD;
```

(N2) $$N_2^{(e)}(x) = \frac{(x - x_1^{(e)})(x_3^{(e)} - x)}{(x_2^{(e)} - x_1^{(e)})(x_3^{(e)} - x_2^{(e)})}$$

```
N3 : n3/DD;
```

(N3) $$N_3^{(e)}(x) = \frac{(x - x_1^{(e)})(x - x_2^{(e)})}{(x_3^{(e)} - x_1^{(e)})(x_3^{(e)} - x_2^{(e)})}$$

Here, we have restored the superscript indication of the element and rearranged the actual output to have the form of the functions given in the problem statement. The functions in the problem statement clearly agree with those we have just deduced.

We still have to demonstrate that these results emerge from the determinants given in the problem statement. To do so, we evaluate the determinants with the statements

```
Dmat : matrix( [1, x1, x1^2], [1,x2,x2^2], [1,x3,x3^2] ) $
Ddet : factor( determinant(Dmat) );
```

(Ddet) $(x_2 - x_1)(x_3 - x_1)(x_3 - x_2)$

```
N1mat : matrix( [1, x, x^2], [1,x2,x2^2], [1,x3,x3^2] ) $
N1det : factor( determinant(N1mat) );
```

(N1det) $(x_2 - x)(x_3 - x)(x_3 - x_2)$

```
N2mat : matrix( [1, x1, x1^2], [1,x,x^2], [1,x3,x3^2] ) $
N2det : factor( determinant(N2mat) );
```

(N2det) $-(x_1 - x)(x_3 - x)(x_3 - x_1)$

```
N3mat : matrix( [1, x1, x1^2], [1,x2,x2^2], [1,x,x^2] ) $
N3det : factor( determinant(N3mat) );
```

(N3det) $(x_1 - x)(x_2 - x)(x_2 - x_1)$

Then, we evaluate the shape functions with the statements

```
N1 : N1det/Ddet; N2 : N2det/Ddet; N3 : N3det/Ddet;
```

yielding the results

$$\begin{aligned} N_1^{(e)}(x) &= \frac{(x_2^{(e)} - x)(x_3^{(e)} - x)}{(x_2^{(e)} - x_1^{(e)})(x_3^{(e)} - x_1^{(e)})} \\ N_2^{(e)}(x) &= -\frac{(x_1^{(e)} - x)(x_3^{(e)} - x)}{(x_2^{(e)} - x_1^{(e)})(x_3^{(e)} - x_2^{(e)})} \\ N_3^{(e)}(x) &= \frac{(x_1^{(e)} - x)(x_2^{(e)} - x)}{(x_3^{(e)} - x_1^{(e)})(x_3^{(e)} - x_2^{(e)})} \end{aligned}$$

wherein we have restored the superscript indication of the specific element. The different evaluations of these shape functions all agree.

(c) To transform these functions into the new variable[7] $\xi = (x - x_1)/l$, we start defining the substitutions

$$x_2 = x_1 + \frac{1}{2}l \quad ; \quad x_3 = x_1 + l \quad ; \quad x = l\,\xi + x_1$$

with the statements

```
x2 : x1 + l/2$ x3 : x1 + l$ x : l*xi + x1$
```

Then, we make the substitutions into the shape functions and simplify the resulting forms with the statements

```
N1xi : factor( ev(N1) );
```

(N1xi) $(\xi - 1)(2\xi - 1)$

```
N2xi : factor( ev(N2) );
```

(N2xi) $-4(\xi - 1)\xi$

```
N3xi : factor( ev(N3) );
```

(N3xi) $\xi(2\xi - 1)$

all of which agree with the expected results laid out in the problem statement.

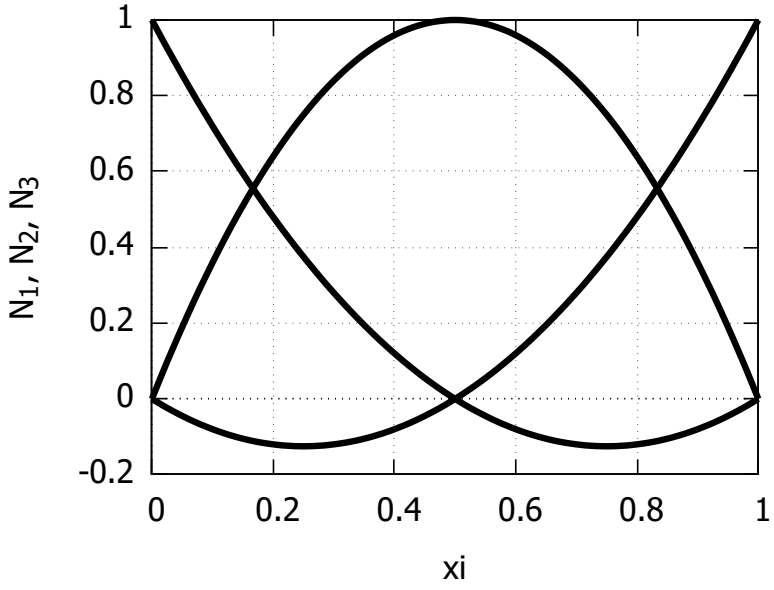
(d) We next obtain graphs of these three shape functions over the interval $0 < \xi < 1$. We can do this in MAXIMA with the statements

```
plot2d([N1xi,N2xi,N3xi], [xi, 0.0,1.0], [color,black], [grid2d,true],
  [xlabel,"xi"], [ylabel,"N_1, N_2, N_3"], [legend,false], [style,[lines,4]] )$
```

The resulting graph is shown in Fig. E15.4. Note that each function has the value 1 at the node to which it is attached and the value 0 at the other two nodes in the element.

[7] We again temporarily drop the superscripts.

Figure E15.4: Graph of the shape functions. Note that $N_1^{(e)}(0) = 1.0$, $N_2^{(e)}(0.5) = 1.0$, and $N_3^{(e)}(1) = 1.0$. This graph was made with MAXIMA.



15.11 Elemental Stiffness Matrix for Three-Node Linear Element (MAXIMA)

Exercise: Accepting the shape functions given in part (c) of Exercise 15.10, (a) find the 3×3 matrices

$$ {}^1K_{ij}^{(e)} = \int_{x_1^{(e)}}^{x_3^{(e)}} \frac{dN_i}{dx}\,\frac{dN_j}{dx}\,dx \qquad \text{and} \qquad {}^2K_{ij}^{(e)} = \int_{x_1^{(e)}}^{x_3^{(e)}} N_i\,N_j\,dx $$

and then construct the 3×3 matrix whose elements are

$$ K_{ij}^{(e)} = \alpha^{(e)}\ {}^1K_{ij}^{(e)} + \beta^{(e)}\ {}^2K_{ij}^{(e)} $$

to produce the (elemental) stiffness matrix analogous to the eighth numbered equation in Section 15.9.3. (b) Find the three-element vector whose elements are

$$ b_i^{(e)} = f^{(e)} \int_{x_1^{(e)}}^{x_3^{(e)}} N_i^{(e)}\,dx $$

analogous to the ninth numbered equation in Section 15.9.3 and the three element vector whose elements are

$$ g_i^{(e)} = \alpha^{(e)}\,N_i^{(e)}\,\frac{d\tilde{\varphi}}{dx}\bigg|_{x=x_3^{(e)}} - \alpha^{(e)}\,N_i^{(e)}\,\frac{d\tilde{\varphi}}{dx}\bigg|_{x=x_1^{(e)}}. $$

analogous to tenth numbered equation in Section 15.9.3. By the time you are done, you should discover that the elemental equation applicable to this three-node element is

$$ \begin{bmatrix} K_{11}^{(e)} & K_{12}^{(e)} & K_{13}^{(e)} \\ K_{21}^{(e)} & K_{22}^{(e)} & K_{22}^{(e)} \\ K_{31}^{(e)} & K_{32}^{(e)} & K_{32}^{(e)} \end{bmatrix} \begin{Bmatrix} \tilde{\varphi}_1^{(e)} \\ \tilde{\varphi}_2^{(e)} \\ \tilde{\varphi}_3^{(e)} \end{Bmatrix} = \begin{Bmatrix} b_1^{(e)} \\ b_2^{(e)} \\ b_3^{(e)} \end{Bmatrix} + \begin{Bmatrix} g_1^{(e)} \\ 0 \\ g_3^{(e)} \end{Bmatrix} $$

where

$$ g_1^{(e)} = -\alpha^{(e)}\ \frac{d\tilde{\varphi}^{(e)}}{dx}\bigg|_{x_1^{(e)}} \qquad ; \qquad g_3^{(e)} = \alpha^{(e)}\ \frac{d\tilde{\varphi}^{(e)}}{dx}\bigg|_{x_3^{(e)}}. $$

Except for factors of $\alpha^{(e)}$, $\beta^{(e)}$, $f^{(e)}$, and $l^{(e)}$, you should have explicit values for $K_{ij}^{(e)}$ and $b_i^{(e)}$. *Hints*: (1) Begin by recasting the integrals as integrals on the variable ξ. (2) You may find a symbolic manipulating program to be useful at many points in this problem. (3) Assume as in Exercise 15.10 that the point $x - x_2$ is midway between x_1 and x_3.

Solution: This exercise takes us through the process of formulating a system of equations (as laid out in Section 15.9.3). The first step is to define the residual, $r(x)$, which in our case, is the same as that given in the text by the first numbered equation in Section 15.9.3,

$$ r(x) = -\frac{d}{dx}\left(\alpha\,\frac{d\tilde{\varphi}}{dx}\right) + \beta\,\tilde{\varphi} - f $$

We then can determine R_i, the weighted residual integrals, with the weighting functions equal to the shape functions, N_i, as in the second numbered equation in Section 15.9.3,[8]

$$ R_i = \int_{x_1}^{x_3} W_i(x)\,r(x)\,dx = 0 \qquad ; \qquad i = 1, 2, 3 $$

Taking all of these factors into account, we are left with third numbered equation in Section 15.9.3, specifically

$$ R_i = \int_{x_1}^{x_3} N_i\,\left[-\frac{d}{dx}\,\left(\alpha\,\frac{d\tilde{\varphi}}{dx}\right) + \beta\,\tilde{\varphi}\right]\,dx - \int_{x_1}^{x_3} N_i\,f\,dx \qquad i = 1, 2, 3 $$

[8] To simplify the expressions, we will suppress the superscripts (e), though we can reinstate them at the end.

When we integrate the first term by parts we find that

$$R_i = \int_{x_1}^{x_3} \left(\alpha \frac{dN_i}{dx} \frac{d\tilde{\varphi}}{dx} + \beta N_i \tilde{\varphi} \right) dx - \int_{x_1}^{x_3} N_i f \, dx - \alpha N_i \frac{d\tilde{\varphi}}{dx}\bigg|_{x_1}^{x_3} \quad ; \quad i = 1, 2, 3$$

Note that the unintegrated term involves *only* values at the ends (boundaries) of the element. Now, if we substitute the approximate solution

$$\tilde{\varphi} = \sum_{j=1}^{3} \tilde{\varphi}_j \, N_j(x)$$

from Exercise 15.10, we find that

$$R_i = \sum_{j=1}^{3} \tilde{\varphi}_j \int_{x_1}^{x_3} \left(\alpha \frac{dN_i}{dx} \frac{dN_j}{dx} + \beta N_i N_j \right) dx - \int_{x_1}^{x_3} N_i f \, dx - \alpha N_i \frac{d\tilde{\varphi}}{dx}\bigg|_{x_1}^{x_3}$$

or

$$R_i = \sum_{j=1}^{3} \tilde{\varphi}_j \int_{x_1}^{x_3} \alpha \frac{dN_i}{dx} \frac{dN_j}{dx} \, dx + \sum_{j=1}^{3} \tilde{\varphi}_j \int_{x_1}^{x_3} \beta N_i N_j \, dx - \int_{x_1}^{x_3} N_i f \, dx - \alpha N_i \frac{d\tilde{\varphi}}{dx}\bigg|_{x_1}^{x_3}$$

We need one more simplification before we are ready to evaluate these integrals. We assume the elements are all quite small, so that we can assume α, β, and f to be approximately constant over each interval of integration, though that constant value may vary with the element (so, in the end, we will have to supply a superscript (e) on these symbols). With that assumption, the integrals become

$$R_i = \sum_{j=1}^{3} \tilde{\varphi}_j \alpha \int_{x_1}^{x_3} \frac{dN_i}{dx} \frac{dN_j}{dx} \, dx + \sum_{j=1}^{3} \tilde{\varphi}_j \beta \int_{x_1}^{x_3} N_i N_j \, dx - f \int_{x_1}^{x_3} N_i \, dx - \alpha N_i \frac{d\tilde{\varphi}}{dx}\bigg|_{x_1}^{x_3}$$

More simply, if we introduce the quantities

$$\begin{aligned} {}^1K_{ij} &= \int_{x_1}^{x_3} \frac{dN_i}{dx} \frac{dN_j}{dx} \, dx \quad ; \quad {}^2K_{ij} = \int_{x_1}^{x_3} N_i N_j \, dx \\ b_i &= \int_{x_1}^{x_3} N_i \, dx \quad ; \quad g_i = \alpha N_i \frac{d\tilde{\varphi}}{dx}\bigg|_{x_1}^{x_3} \end{aligned} \tag{E15.11.1}$$

we can then write

$$R_i = \sum_{j=1}^{3} (\alpha \, {}^1K_{ij} + \beta \, {}^2K_{ij}) \tilde{\varphi}_j - f \, b_i - g_i \tag{E15.11.2}$$

Our task now is to evaluate the several integrals in Eq. (E15.11.1) for the shape functions given by the expressions

$$N_1(x) = \frac{(x_2 - x)(x_3 - x)}{(x_2 - x_1)(x_3 - x_1)} \quad ; \quad N_2(x) = -\frac{(x_1 - x)(x_3 - x)}{(x_2 - x_1)(x_3 - x_2)} \quad ; \quad N_3(x) = \frac{(x_1 - x)(x_2 - x)}{(x_3 - x_1)(x_3 - x_2)}$$

deduced in Exercise 15.10.

(a) Our task is much simplified if we recast the integrals in terms of the variable ξ defined implicitly by the equation $x = l\xi + x_1$. With this transformation (and with the assumptions that $x_2 = x_1 + \frac{1}{2}l$ and $x_3 = x_1 + l$),[9]

$$dx \longrightarrow l \, d\xi \quad ; \quad \frac{dN_i}{dx} \longrightarrow \frac{1}{l} \frac{dN_i}{d\xi} \quad ; \quad x = x_1 \rightarrow \xi = 0 \quad ; \quad x = x_2 \rightarrow \xi = \frac{1}{2} \quad ; \quad x = x_3 \rightarrow \xi = 1$$

[9] Remember that, along with items in this expression already identified, l depends on the element, though we continue to suppress the superscript (e) that would remind us of that dependence.

With these substitutions, the integrals we must evaluate become

$$ {}^1K_{ij} = \int_0^1 \frac{dN_i}{l\,d\xi}\,\frac{dN_j}{l\,d\xi}\,l\,d\xi = l\int_0^1 \frac{dN_i}{d\xi}\,\frac{dN_j}{d\xi}\,d\xi $$

$$ {}^2K_{ij} = \int_0^1 N_i\,N_j\,l\,d\xi = l\int_0^1 N_i\,N_j\,d\xi $$

and

$$ b_i = \int_0^1 N_i\,d\xi = l\int_0^1 N_i d\xi $$

At this point the integrands are all explicitly known, so the integrals can be evaluated. With the shape functions,

$$ N_1(\xi) = (\xi-1)(2\xi-1) \quad ; \quad N_2(\xi) = -4(\xi-1)\xi \quad ; \quad N_3(\xi) = \xi(2\xi-1) $$

as in Exercise 15.10, we are now ready to evaluate these integrals.

We begin in MAXIMA by defining the shape functions and their derivatives as *vectors* with the statements

```
N : [(xi-1)*(2*xi-1), -4*(xi-1)*xi, xi*(2*xi-1) ]$
dN : diff(N,xi)$
```

We can then evaluate b_i directly with the statement

```
b : l*integrate(N,xi, 0, 1);
```

yielding the result that

$$ b = \left[\frac{l}{6}, \frac{2l}{3}, \frac{l}{6}\right] $$

To evaluate ${}^1K_{ij}$, we need first to create a two-dimensional matrix containing the products N_iN_j. Since N is a row vector, we can create that two-dimensional matrix with the expression[10] `transpose(N).N`. The integrals of interest are then evaluated as a matrix. For ${}^1K$, we use the statement

```
K1 : l * integrate(transpose(dN).dN, xi, 0, 1);
```

$$ {}^1K = \begin{bmatrix} 7l/3 & -8l/3 & l/3 \\ -8l/3 & 16l/3 & -8l/3 \\ l/3 & -8l/3 & 7l/3 \end{bmatrix} $$

Similarly, we find ${}^2K_{ij}$ with the statement

```
K2 : l * integrate( transpose(N).N, xi, 0, 1);
```

with the result

$$ {}^2K = \begin{bmatrix} 2l/15 & l/15 & -l/30 \\ l/15 & 8l/15 & l/15 \\ -l/30 & l/15 & 2l/15 \end{bmatrix} $$

Only evaluation of g_i remains. Remember from Exercise 15.10 that N_i has the value 1 at $x_i^{(e)}$ and the value zero at the other two nodes in the element. In particular (and in terms of the variable

[10] Remember that ordinary matrix multiplication is effected with the operator `.`.

ξ), the values of concern here are $N_1(0) = 1$, $N_3(1) = 1$, and $N_2(0) = N_2(1) = 0$. With g_i defined by

$$g_i = \left(\alpha(x)\, N_i\, \frac{d\tilde{\varphi}}{dx} \right) \Bigg|_{x_1}^{x_3}$$

Since $N_1(x_3) = 0$, we conclude that[11]

$$g_1^{(e)} = \alpha(x_3^{(e)})\, N_1(1) \frac{d\tilde{\varphi}}{dx}\bigg|_{x_3^{(e)}} - \alpha(x_1^{(e)})\, N_1(0) \frac{d\tilde{\varphi}}{dx}\bigg|_{x_1^{(e)}} = -\,\alpha(x_1^{(e)})\, \frac{d\tilde{\varphi}}{dx}\bigg|_{x_1^{(e)}}$$

Similarly

$$g_2^{(e)} = \alpha(x_3^{(e)})\, N_2(1) \frac{d\tilde{\varphi}}{dx}\bigg|_{x_3^{(e)}} - \alpha(x_1^{(e)})\, N_2(0) \frac{d\tilde{\varphi}}{dx}\bigg|_{x_1^{(e)}} = 0$$

and

$$g_3^{(e)} = \alpha(x_3^{(e)})\, N_3(1) \frac{d\tilde{\varphi}}{dx}\bigg|_{x_3^{(e)}} - \alpha(x_1^{(e)})\, N_3(0) \frac{d\tilde{\varphi}}{dx}\bigg|_{x_1^{(e)}} = \alpha(x_3^{(e)})\, \frac{d\tilde{\varphi}}{dx}\bigg|_{x_3^{(e)}}$$

We are now ready to put all these results together into the equation that characterizes the solution in the single element (e). Specifically, that equation emerges if we require the residuals in Eq. (E15.11.2) all to be zero. We learn that we must find the unknown quantities $\tilde{\varphi}_i, i = 1, 2, 3$ to satisfy the equation

$$\sum_{j=1}^{3} (\alpha\, {}^1K_{ij} + \beta\, {}^2K_{ij})\, \tilde{\varphi}_j = f\, b_i + g_i$$

If we introduce the vectors

$$\tilde{\varphi}^{(e)} = \begin{Bmatrix} \tilde{\varphi}_1^{(e)} \\ \tilde{\varphi}_2^{(e)} \\ \tilde{\varphi}_3^{(e)} \end{Bmatrix} \quad , \quad b^{(e)} = \begin{Bmatrix} b_1^{(e)} \\ b_2^{(e)} \\ b_3^{(e)} \end{Bmatrix} \quad \text{and} \quad g^{(e)} = \begin{Bmatrix} g_1^{(e)} \\ g_2^{(e)} \\ g_3^{(e)} \end{Bmatrix}$$

and the matrix $[K^{(e)}]$ whose ij element is

$$K_{ij}^{(e)} = \alpha(e)\, {}^1K_{ij}^{(e)} + \beta(e)\, {}^2K_{ij}^{(e)}$$

we can then write the essential equation in the compact form

$$K^{(e)} \tilde{\varphi}^{(e)} = f^{(e)}\, b^{(e)} + g^{(e)}$$

From the results obtained in the several previous paragraphs, we have learned that

$$K^{(e)} = \alpha^{(e)}\, l^{(e)} \begin{bmatrix} 7/3 & -8/3 & 1/3 \\ -8/3 & 16/3 & -8/3 \\ 1/3 & -8/3 & 7/3 \end{bmatrix} + \beta^{(e)}\; l^{(e)} \begin{bmatrix} 2/15 & 1/15 & -1/30 \\ 1/15 & 8/15 & 1/15 \\ -1/30 & 1/15 & 2/15 \end{bmatrix}$$

and

$$b^{(e)} = l^{(e)} \begin{Bmatrix} 1/6 \\ 2/3 \\ 1/6 \end{Bmatrix} \quad ; \quad g^{(e)} = \begin{Bmatrix} -\,\alpha(x_1^{(e)})\, \dfrac{d\tilde{\varphi}}{dx}\Big|_{x_1^{(e)}} \\ 0 \\ \alpha(x_3^{(e)})\, \dfrac{d\tilde{\varphi}}{dx}\Big|_{x_3^{(e)}} \end{Bmatrix}$$

Assembly of these elemental equations into a system with three elements is the subject of Exercise 15.12.

[11] We also reinstate the superscripts identifying the element involved, and we recognize that $x_1^{(e)}$ and $x_3^{(e)}$ correspond to $\xi = 0$ and $\xi = 1$, respectively, and we think of N_i as a function of ξ.

15.15 Shape Functions for Triangular Element (MAXIMA)

Exercise: (a) Solve the equations

$$\begin{aligned} \tilde{\varphi}_0^{(e)} &= a^{(e)} + b^{(e)}\, x_0^{(e)} + c^{(e)}\, y_0^{(e)} \\ \tilde{\varphi}_1^{(e)} &= a^{(e)} + b^{(e)}\, x_1^{(e)} + c^{(e)}\, y_1^{(e)} \\ \tilde{\varphi}_2^{(e)} &= a^{(e)} + b^{(e)}\, x_2^{(e)} + c^{(e)}\, y_2^{(e)} \end{aligned} \tag{E15.15.1}$$

in Section 15.23.2 for the constants $a^{(e)}$, $b^{(e)}$, and $c^{(e)}$ and then verify the expressions given in that section for the shape functions $N_i^{(e)}(x, y)$. Remember that symbol manipulating programs are available. (b) For each node i, verify that the function $N_i^{(e)}(x, y)$ is zero not only at the two nodes not identified by its index but also along the entire line joining those two nodes. *Hint*: Set the determinant in the numerator of the expression giving $N_i^{(e)}$ to zero, thereby obtaining the equation of a line in the plane. Verify that that line is, in fact, the line joining the two described nodes. (c) Find the functions to which these functions reduce when $(x_0, y_0) = (-1, 0)$, $(x_1, y_1) = (1, 0)$, $(x_2, y_2) = (0, 1)$ and demonstrate that each is zero where it is supposed to be zero and one where it is supposed to be one.

Solution: (a) To define and solve Eq. (E15.15.1) for the constants $a^{(e)}$, $b^{(e)}$, and $c^{(e)}$ using MAXIMA, we invoke the statements[12]

```
eq0 : phi0 = a + b*x0 + c*y0 $
eq1 : phi1 = a + b*x1 + c*y1 $
eq2 : phi2 = a + b*x2 + c*y2 $
soln : solve([eq0,eq1,eq2], [a,b,c] )
```

(soln) $\left[\left[a = \frac{a_n}{d}, \frac{b_n}{d}, \frac{c_n}{d}\right]\right]$

where we have indicated only the form of the solution. The several numerators a_n, b_n, and c_n and the common denominator d can be extracted from this complicated result with the statements

```
an : num( ev(a,soln) )$
an : collectterms(expand(an), phi0,phi1,phi2 );
```

(an) $\tilde{\varphi}_0(x_2y_1 - x_1y_2) + \tilde{\varphi}_1(x_0y_2 - x_2y_0) + \tilde{\varphi}_2(x_1y_0 - x_0y_1)$

```
bn : num( ev(b,soln) )$
bn : collectterms(expand(bn), phi0,phi1,phi2 );
```

(bn) $\tilde{\varphi}_0(y_2 - y_1) + \tilde{\varphi}_1(y_0 - y_2) + \tilde{\varphi}_2(y_1 - y_0)$

```
cn : num( ev(c,soln) )$
cn : collectterms(expand(cn), phi0,phi1,phi2 );
```

(cn) $\tilde{\varphi}_1(x_2 - x_0) + \tilde{\varphi}_0(x_1 - x_2) + \tilde{\varphi}_2(x_0 - x_1)$

```
d : denom( ev(a,soln) )$
d : collectterms(expand(d), x0,x1,x2 );
```

(d) $x_0(y_2 - y_1) + x_1(y_0 - y_2) + x_2(y_1 - y_0)$

[12] Note that the order in which the coefficients appear in the solution will vary. Note also that we are suppressing the superscript (e), at least temporarily.

With these results, the approximating function becomes

$$\tilde{\varphi} = a + bx + cy = \frac{a_n + b_n x + c_n y}{d}$$

The *numerator* in this last expression is given and then the coefficient of each $\tilde{\varphi}_i$ in that numerator collected by the statements

```
phin : an + bn*x + cn*y$
phin : collectterms( expand(phin), phi0, phi1, phi2 );
```

(phin) $\tilde{\varphi}_0 n_0 + \tilde{\varphi}_1 n_1 + \tilde{\varphi}_2 n_2$

where we have only indicated the form of the result. The several functions n_0, n_1, and n_2 can then be extracted and massaged with the statements

```
n0 : collectterms(part(phin,1,2),x,y);
```

(n0) $x(y_2 - y_1) - x_1 y_2 + x_2 y_1 + (x_1 - x_2)y$

```
n1 : collectterms(part(phin,2,2),x,y);
```

(n1) $x_0 y_2 + x(y_0 - y_2) - x_2 y_0 + (x_2 - x_0)y$

```
n2 : collectterms(part(phin,3,2),x,y);
```

(n2) $x(y_1 - y_0) - x_0 y_1 + x_1 y_0 + (x_0 - x_1)y$

With these results, we quickly find that

```
N0 : n0/d;
```

$$\text{(N0)} \quad \frac{x(y_2 - y_1) - x_1 y_2 + x_2 y_1 + (x_1 - x_2)y}{x_0(y_2 - y_1) + x_1(y_0 - y_2) + x_2(y_1 - y_0)}$$

```
N1 : n1/d;
```

$$\text{N1)} \quad \frac{x_0 y_2 + x(y_0 - y_2) - x_2 y_0 + (x_2 - x_0)y}{x_0(y_2 - y_1) + x_1(y_0 - y_2) + x_2(y_1 - y_0)}$$

```
N2 : n2/d;
```

$$\text{(N2)} \quad \frac{x(y_1 - y_0) - x_0 y_1 + x_1 y_0 + (x_0 - x_1)y}{x_0(y_2 - y_1) + x_1(y_0 - y_2) + x_2(y_1 - y_0)}$$

To verify that these expressions agree with those given in Section 15.23.2, we simply evaluate the determinants given there with the statements

```
Delta :  matrix( [1,x0,y0], [1,x1,y1], [1,x2,y2] );
```

$$\text{(Delta)} \quad \begin{pmatrix} 1 & x_0 & y_0 \\ 1 & x_1 & y_1 \\ 1 & x_2 & y_2 \end{pmatrix}$$

```
Ddet :  collectterms( expand(determinant(Delta)), x0,x1,x2 );
```

(Ddet) $x_1(y_2 - y_0) + x_0(y_1 - y_2) + x_2(y_0 - y_1)$

```
Anum :  matrix( [1,x,y], [1,x1,y1], [1,x2,y2] );
```

$$\text{(Anum)} \quad \begin{pmatrix} 1 & x & y \\ 1 & x_1 & y_1 \\ 1 & x_2 & y_2 \end{pmatrix}$$

```
Adet :  collectterms( expand(determinant(Anum)), x,y );
```

(Adet) $x_1y_2 + x(y_1 - y_2) - x_2y_1 + (x_2 - x_1)y$

```
Bnum :  matrix( [1,x0,y0], [1,x,y], [1,x2,y2] );
```

(Bnum) $\begin{pmatrix} 1 & x_0 & y_0 \\ 1 & x & y \\ 1 & x_2 & y_2 \end{pmatrix}$

```
(
```

(Bdet) $x(y_2 - y_0) - x_0y_2 + x_2y_0 + (x_0 - x_2)y$

```
Cnum :  matrix( [1,x0,y0], [1,x1,y1], [1,x,y] );
```

(Cnum) $\begin{pmatrix} 1 & x_0 & y_0 \\ 1 & x_1 & y_1 \\ 1 & x & y \end{pmatrix}$

```
Cdet :  collectterms( expand(determinant(Cnum)), x,y );
```

(Cdet) $x_0y_1 + x(y_0 - y_1) - x_1y_0 + (x_1 - x_0)y$

Intriguingly, the statement

```
[is(equal(n0,-Adet)), is(equal(n1,-Bdet)), is(equal(n2,-Cdet))];
```

$[true, true, true]$

```
is(equal(d,-Ddet));
```

$true$

reveals that the numerators and denominator just determined from determinants are all the negatives of the corresponding results obtained by direct solution of the defining equation earlier in this solution. Hence, the *ratios* of numerators to the denominator are the same in both cases. Reinstating the superscript (e) stipulation, we conclude that

$$N_0^{(e)}(x,y) = \frac{1}{\Delta^{(e)}} \begin{vmatrix} 1 & x & y \\ 1 & x_1^{(e)} & y_1^{(e)} \\ 1 & x_2^{(e)} & y_2^{(e)} \end{vmatrix} \quad ; \quad N_1^{(e)}(x,y) = \frac{1}{\Delta^{(e)}} \begin{vmatrix} 1 & x_0^{(e)} & y_0^{(e)} \\ 1 & x & y \\ 1 & x_2^{(e)} & y_2^{(e)} \end{vmatrix}$$

$$N_2^{(e)}(x,y) = \frac{1}{\Delta^{(e)}} \begin{vmatrix} 1 & x_0^{(e)} & y_0^{(e)} \\ 1 & x_1^{(e)} & y_1^{(e)} \\ 1 & x & y \end{vmatrix} \quad \text{where} \quad \Delta^{(e)} = \begin{vmatrix} 1 & x_0^{(e)} & y_0^{(e)} \\ 1 & x_1^{(e)} & y_1^{(e)} \\ 1 & x_2^{(e)} & y_2^{(e)} \end{vmatrix}$$

confirming the expressions provided in Section 15.23.2.

(b) We quickly demonstrate that each shape function has the value 1 at the node with which it is associated and the value zero at the other two nodes. For example,

$$N_0^{(e)}(x_0^{(e)}, y_0^{(e)}) = \frac{1}{\Delta^{(e)}} \begin{vmatrix} 1 & x_0^{(e)} & y_0^{(e)} \\ 1 & x_1^{(e)} & y_1^{(e)} \\ 1 & x_2^{(e)} & y_2^{(e)} \end{vmatrix}$$

With this substitution, the determinant in the numerator of this expression is, in fact, $\Delta^{(e)}$, so $N_0^{(e)}(x_0^{(e)}, y_0^{(e)}) = 1$. In contrast,

$$N_0^{(e)}(x_1^{(e)}, y_1^{(e)}) = \frac{1}{\Delta^{(e)}} \begin{vmatrix} 1 & x_0^{(e)} & y_0^{(e)} \\ 1 & x_1^{(e)} & y_1^{(e)} \\ 1 & x_2^{(e)} & y_2^{(e)} \end{vmatrix}$$

This time, the determinant in the numerator has two rows the same and its value is therefore 0, so $N_0^{(e)}(x_1^{(e)}, y_1^{(e)}) = 0$. Further,

$$N_0^{(e)}(x_2^{(e)}, y_2^{(e)}) = \frac{1}{\Delta^{(e)}} \begin{vmatrix} 1 & x_2^{(e)} & y_2^{(e)} \\ 1 & x_1^{(e)} & y_1^{(e)} \\ 1 & x_2^{(e)} & y_2^{(e)} \end{vmatrix}$$

Again, the determinant has two identical rows, so $N_0^{(e)}(x_2^{(e)}, y_2^{(e)}) = 0$. Similar arguments will demonstrate that

$$N_1^{(e)}(x_0^{(e)}, y_0^{(e)}) = 0 \quad ; \quad N_1^{(e)}(x_1^{(e)}, y_1^{(e)}) = 1 \quad ; \quad N_1^{(e)}(x_2^{(e)}, y_2^{(e)}) = 0$$
$$N_2^{(e)}(x_0^{(e)}, y_0^{(e)}) = 0 \quad ; \quad N_2^{(e)}(x_1^{(e)}, y_1^{(e)}) = 0 \quad ; \quad N_2^{(e)}(x_2^{(e)}, y_2^{(e)}) = 1$$

Demonstrating that each of these functions is zero everywhere along a line joining the two nodes to which the function is *not* attached is more difficult. For example, $N_0^{(e)}(x, y)$ will be zero if its numerator n_0 is zero, i.e., if

$$n_0(x, y) = x(y_2 - y_1) - x_1 y_2 + x_2 y_1 + (x_1 - x_2)y = 0$$

This condition requires that

$$y = \frac{y_2 - y_1}{x_2 - x_1}\, x + \frac{x_2 y_1 - x_1 y_2}{x_2 - x_1}$$

Compare this with the equation of the line joining the two points (x_1, y_1) and (x_2, y_2), which characterized by slope m and intercept b satisfying

$$y_1 = m x_1 + b \quad ; \quad y_2 = m x_2 + b$$

Subtracting the first of these equations from the second yields that

$$(y_2 - y_1) = m(x_2 - x_1) \quad \Longrightarrow \quad m = \frac{y_2 - y_1}{x_2 - x_1}$$

and then the first equation, say, yields that

$$b = y_1 - \frac{y_2 - y_1}{x_2 - x_1}\, x_1 = \frac{y_1(x_2 - x_1) - x_1(y_2 - y_1)}{x_2 - x_1} = \frac{y_1 x_2 - x_1 y_2}{x_2 - x_1}$$

Substituting these values for m and b into the equation $y = mx + b$ yields that the equation of the line joining the two points is

$$y = \frac{y_2 - y_1}{x_2 - x_1}\, x + \frac{y_1 x_2 - x_1 y_2}{x_2 - x_1}$$

The equation of the line joining the two points in question is identical to the equation emerging by requiring $n_0(x, y) = 0$. Q.E.D. Similar arguments will reveal the corresponding property for the functions $n_1(x, y)$ and $n_2(x, y)$.

(c) Finally, we wish to find the three shape functions for the element shown in Fig. E15.5, where $(x_0, y_0) = (-1, 0)$, $(x_1, y_1) = (1, 0)$. The statements

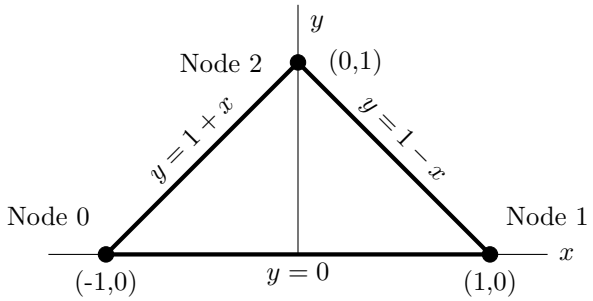
```
x0:-1$ y0:0$ x1:1$ y1:0$ x2:0$ y2:1$
ev(N0); ev(N1); ev(N2);
```

yield that

$$N_0 = -\frac{y + x - 1}{2} \quad ; \quad N_1 = -\frac{y - x - 1}{2} \quad ; \quad N_2 = y$$

Since

Figure E15.5: Figure for Exercise 15.15.



- $y = 0$ everywhere along the line from Node 0 to Node 1, $N_2 = 0$ along that line.
- $y = 1$ at Node 2, $N_2 = 1$ at Node 2.
- $y = 1 - x$ everywhere along the line from Node 1 to Node 2, $N_0 = 0$ along that line.
- $x = -1$ and $y = 0$ at Node 0, $N_0 = 1$ at that node.
- $y = 1 + x$ everywhere along the line from Node 0 to Node 2, $N_1 = 0$ along that line.
- $x = 1$ and $y = 0$ at Node 1, $N_1 = 1$ at that node.

15.16 Shape Functions for Four-Node 2D Element (MAXIMA)

Exercise: Consider a four-node element in two dimensions. Let the nodes be at (x_0, y_0), (x_1, y_1), (x_2, y_2), and (x_3, y_3), and take the interpolating function to be $\tilde{\varphi} = a + bx + cy + dxy$. (a) Find the four shape functions in general terms, showing that those functions can be expressed in the form

$$N_0(x,y) = \frac{1}{\Delta}\begin{vmatrix} 1 & x & y & xy \\ 1 & x_1 & y_1 & x_1y_1 \\ 1 & x_2 & y_2 & x_2y_2 \\ 1 & x_3 & y_3 & x_3y_3 \end{vmatrix} \quad \text{where} \quad \Delta = \begin{vmatrix} 1 & x_0 & y_0 & x_0y_0 \\ 1 & x_1 & y_1 & x_1y_1 \\ 1 & x_2 & y_2 & x_2y_2 \\ 1 & x_3 & y_3 & x_3y_3 \end{vmatrix}$$

and in similar forms for $N_1(x,y)$, $N_2(x,y)$, and $N_3(x,y)$. (b) Then, show that, if the nodes lie at the corners of a square of side s, i.e., the nodes are—in order—at $(0,0)$, $(s,0)$, (s,s), and $(0,s)$, the shape functions expressed in terms of the variables $\xi = x/s$ and $\eta = y/s$ are

$$N_0 = (1-\eta)(1-\xi) \quad ; \quad N_1 = \xi(1-\eta) \quad ; \quad N_2 = \xi\eta \quad ; \quad N_3 = \eta(1-\xi)$$

(c) Finally, generate surface graphs of the four functions and verify that the shape function associated with each node is zero everywhere along the two edges that intersect at the diagonally opposite node. *Hints*: (1) Remember that symbol manipulating programs are available. (2) Some results along the way to a solution are very involved. Presentation of the intermediate results can certainly be suppressed.

Solution: (a) To define and solve the basic equations for the constants a, b, c, and d using MAXIMA, we invoke the statements[13]

```
eq0 : phi0 = a + b*x0 + c*y0 + d*x0*y0 $
eq1 : phi1 = a + b*x1 + c*y1 + d*x1*y1 $
eq2 : phi2 = a + b*x2 + c*y2 + d*x2*y2 $
eq3 : phi3 = a + b*x3 + c*y3 + d*x3*y3 $
soln : solve([eq0,eq1,eq2,eq3], [a,b,c,d] );
```

(soln) $\left[\left[a = \frac{a_n}{den}, \frac{b_n}{den}, \frac{c_n}{den}, \frac{d_n}{den}\right]\right]$

We suppress presentation of the *very* lengthy and complicated result in favor of indicating only the general form of that result. Instead, we work to show agreement with the determinants shown in the statement of the exercise without actually showing the detailed results at all.

We verify first that the denominators in all four constants in `soln` are the same with the statements

```
is( equal( denom(ev(a,soln)), denom(ev(b,soln)) ));
```

true

```
is( equal( denom(ev(a,soln)), denom(ev(c,soln)) ));
```

true

```
is( equal( denom(ev(a,soln)), denom(ev(d,soln)) ));
```

true

[13] Note that the order in which the coefficients appear in the solution *may* vary, though the method here adopted to extract those coefficients will work whatever the order.

We thus defend the use of the same symbol for all four denominators in `soln`.

We next verify that the common denominator in `soln` agrees with the denominator Δ given in the statement of the exercise. To that end, we evaluate Δ with the statements

```
Drow0 : [1,x0,y0,x0*y0]$ Drow1 : [1,x1,y1,x1*y1]$
Drow2 : [1,x2,y2,x2*y2]$ Drow3 : [1,x3,y3,x3*y3]$
Delta : matrix( Drow0, Drow1, Drow2, Drow3 )$
```

The statement

```
is( equal( denom(ev(a,soln)), determinant(Delta) ) );
```

true

then confirms the agreement of the denominator in `soln` with the determinant in the statement of the exercise.

The remaining task in this part of the exercise is to massage the numerator in the approximating function

$$\tilde{\varphi} = a + bx + cy + dxy = \frac{a_n + b_n x + c_n y + d_n xy}{\Delta}$$

into the form

$$\tilde{\varphi} = \frac{1}{\Delta}\Big(n_0\tilde{\varphi}_0 + n_1\tilde{\varphi}_1 + n_2\tilde{\varphi}_2 + n_3\tilde{\varphi}_3\Big)$$

To that end, we (1) execute the statements

```
an : num(ev(a,soln))$
bn : num(ev(b,soln))$
cn : num(ev(c,soln))$
dn : num(ev(d,soln))$
```

to extract a_n, b_n, c_n, and d_n, (2) execute the statements

```
phin : expand( an + bn*x + cn*y + dn*x*y )$
phin : collectterms( phin, phi0, phi1, phi2, phi3 );
```

`(phin)` $\tilde{\varphi}_0 n_0 + \tilde{\varphi}_1 n_1 + \tilde{\varphi}_2 n_2 + \tilde{\varphi}_3 n_3$

to recast the numerator of the approximating function to reveal the numerators n_0, n_1, n_2, and n_3 of the shape functions, and (3) execute the statements

```
n0 : part(phin,1,2)$
n1 : part(phin,2,2)$
n2 : part(phin,3,2)$
n3 : part(phin,4,2)$
```

to extract the numerators of the individual shape functions.[14]

Turning now to the numerators in the functions N_0, N_1, N_2, and N_3 as given in the statement of the exercise, we execute the statements

[14] Since the order in which the terms in `phin` are presented may vary, you will need to inspect *your* value for `phin` to determine which parts should be extracted as the several quantities n_0, n_1, n_2, and n_3.

```
N0row0 : [1,x,y,x*y]$ N0row1 : [1,x1,y1,x1*y1]$
N0row2 : [1,x2,y2,x2*y2]$ N0row3 : [1,x3,y3,x3*y3]$
N0mat : matrix( N0row0, N0row1, N0row2, N0row3 )$
is( equal( determinant(N0mat), n0 ) );
```

true

to define the matrix in the numerator of $N_0(x, y)$ and compare it with the numerator extracted from `soln`. Their agreement confirms the correctness of the determinant given in the statement of the exercise. Similarly, the statements

```
N1row0 : [1,x0,y0,x0*y0]$ N1row1 : [1,x,y,x*y]$
N1row2 : [1,x2,y2,x2*y2]$ N1row3 : [1,x3,y3,x3*y3]$
N1mat : matrix( N1row0, N1row1, N1row2, N1row3 )$
is( equal( determinant(N1mat), n1 ) );
```

true

```
N2row0 : [1,x0,y0,x0*y0]$ N2row1 : [1,x1,y1,x1*y1]$
N2row2 : [1,x,y,x*y]$ N2row3 : [1,x3,y3,x3*y3]$
N2mat : matrix( N2row0, N2row1, N2row2, N2row3 )$
is( equal( determinant(N2mat), n2 ) );
```

true

```
N3row0 : [1,x0,y0,x0*y0]$ N3row1 : [1,x1,y1,x1*y1]$
N3row2 : [1,x2,y2,x2*y2]$ N3row3 : [1,x,y,x*y]$
N3mat : matrix( N3row0, N3row1, N3row2, N3row3 )$
is( equal( determinant(N3mat), n3 ) );
```

true

confirm the correctness of the determinants given in the statement of the exercise.

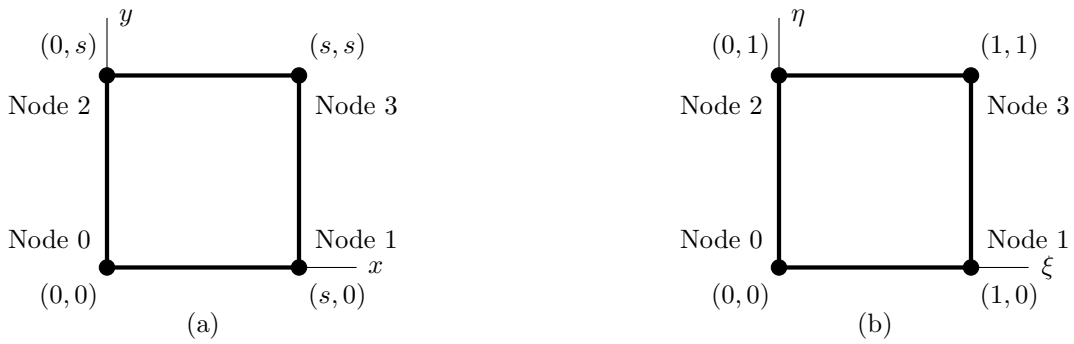
The shape functions determined to this point apply to *any* four-noded, 2D element, regardless of how the nodes are situated relative to one another. The expressions, however, are extremely involved. To simplify the expressions, suppose the element is square and is situated as shown in Fig. E15.6(a). We restrict the coordinates of the nodes as shown in Fig. E15.6(b) with the statements

```
x0:0$ x1:s$ x2:s$x3:0$  y0:0$ y1:0$ y2:s$ y3:s$
```

Then we evaluate the critical determinants with the statements

```
Dden : ev(determinant(Delta));
(Dden)   -s^4
N0num : ev(determinant(N0mat));
(N0num)   -s^2*x*y+s^3*y+s^3*x-s^4
```

Figure E15.6: Figure for Exercise 15.16.




```
N1num : ev(determinant(N1mat));
```

(N1num) $s^2xy - s^3x$

```
N2num : ev(determinant(N2mat));
```

(N2num) $-s^2xy$

```
N3num : expand(ev(determinant(N3mat)));
```

(N2num) $s^2xy - s^3y$

We next introduce the variables $\xi = x/s$ and $\eta = y/s$ and clean up the results with the statements

```
x : s*xi$ y : s*eta$
[ev(N0num), ev(N1num), ev(N2num), ev(N3num)]$
N : map( factor, % )/Dden;
```

finding that

$$N_0(\xi,\eta) = (\eta-1)(\xi-1) \quad ; \quad N_1(\xi,\eta) = -(\eta-1)\xi \quad ; \quad N_2(\xi,\eta) = \eta\xi \quad ; \quad N_3(\xi,\eta) = -\eta(\xi-1)$$

in agreement with the results included in the statement of the exercise.

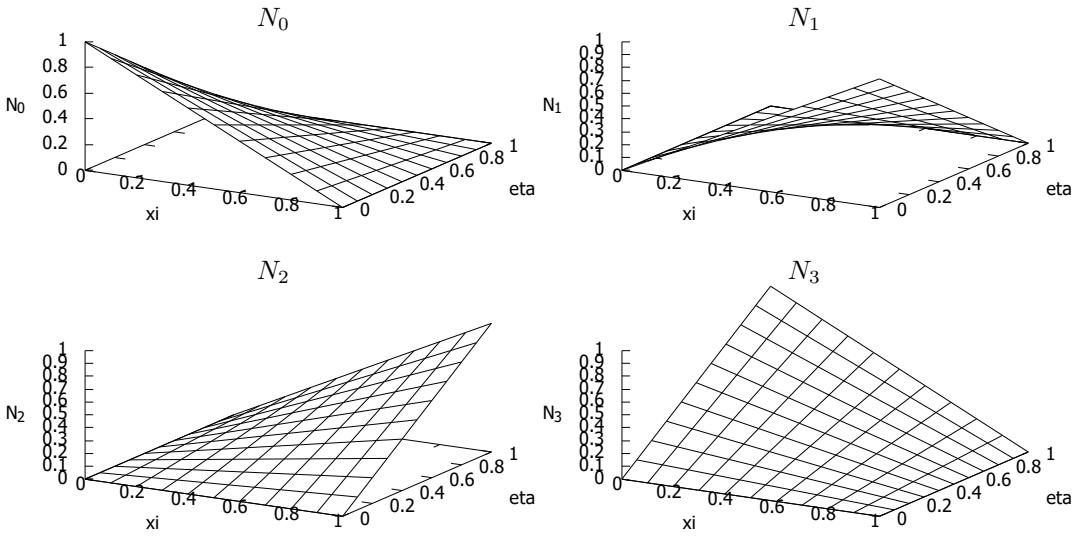
(c) Finally, we seek surface plots of the four shape functions over the $\xi\eta$ plane. We recognize that the region of interest is defined by $0 \le \xi, \eta \le 1$. The statements

```
plot3d( (eta-1)*(xi-1), [xi,0,1], [eta,0,1], [grid,10,10],
    [legend,""], [zlabel, "N_0"], [palette, false], [color,black] )$
plot3d( -(eta-1)*xi, [xi,0,1], [eta,0,1], [grid,10,10],
    [legend,""], [zlabel, "N_1"], [palette, false], [color,black] )$
plot3d( eta*xi, [xi,0,1], [eta,0,1], [grid,10,10],
    [legend,""], [zlabel, "N_2"], [palette, false], [color,black] )$
plot3d( -eta*(xi-1), [xi,0,1], [eta,0,1], [grid,10,10],
    [legend,""], [zlabel, "N_3"], [palette, false], [color,black] )$
```

yields the graphs shown in Fig. E15.7. Clearly each function has the value 1 at the corresponding node and each has the value 0 at all other nodes. That N_3 is zero along the line joining Node 0 to Node 1 and also zero along the line joining Node 1 to Node 2, i.e., along the two sides neither of which includes Node 3, is evident. That Nodes 0, 1, and 2 are similarly characterized is less evident in these graphs, though the conclusion is still correct. We illustrate with an argument that applies to Node 0, for which, in a somewhat easier arrangement than shown above,

$$N_0 = (1-\eta)(1-\xi)$$

Figure E15.7: Graphs of the four shape functions in Exercise 15.16. This graph was produced in MAXIMA.



At node 0, both η and ξ are both zero, so $N_0 = 1$ at that node. Either η or ξ has or both have the value one at the other three nodes, so at least one factor in N_0 is zero at each of those nodes, and N_0 itself is therefore zero at each of those nodes. Note, in addition, that ξ has the value 1 along the *entire* line joining Node 1 to Node 2, so $N_0 = 0$ along that entire line. Similarly, η has the value 1 along the entire line joining Node 2 to Node 3, so $N_0 = 0$ along that line as well. Similar arguments will confirm the corresponding properties at each of Nodes 1, 2, and 3.