Chapter 4

First Order Numerical Methods

Contents

4.1 Solving $y' = F(x)$ Numerically

Studied here is the creation of numerical tables and graphics for the solution of the initial value problem

$$(1) \qquad y' = F(x), \quad y(x_0) = y_0.$$

To illustrate, consider the initial value problem

$$y' = 3x^2 - 1, \quad y(0) = 2.$$

Quadrature gives the explicit **symbolic solution**

$$y(x) = x^3 - x + 2.$$

In Figure 1, evaluation of $y(x)$ from $x = 0$ to $x = 1$ in increments of 0.1 gives the xy-table, whose entries represent the **dots** for the **connect-the-dots** graphic.

x	y
0.0	2.000
0.1	1.901
0.2	1.808
0.3	1.727
0.4	1.664
0.5	1.625

x	y
0.6	1.616
0.7	1.643
0.8	1.712
0.9	1.829
1.0	2.000

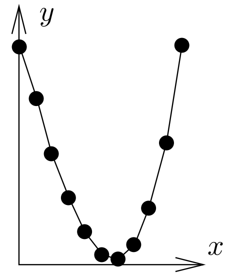


Figure 1. A table of xy-values for $y = x^3 - x + 2$.
The graphic represents the table's rows as *dots*, which are joined to make the *connect-the-dots* graphic.

The interesting case is when quadrature in (1) encounters an integral $\int_{x_0}^{x} F(t)dt$ that cannot be evaluated to provide an explicit symbolic equation for $y(x)$. Nevertheless, $y(x)$ can be computed numerically.

Applied here are numerical integration rules from calculus: *rectangular*, *trapezoidal* and *Simpson*; see page 231 for a review of the three rules. The ideas lead to the numerical methods of Euler, Heun and Runge-Kutta, which appear later in this chapter.

How to make an xy-table. Given $y' = F(x)$, $y(x_0) = y_0$, a table of xy-values is created as follows. The x-values are equally spaced a distance $h > 0$ apart. Each x, y pair in the table represents a *dot* in the *connect-the-dots* graphic of the explicit solution

$$y(x) = y_0 + \int_{x_0}^{x} F(t)dt.$$

First table entry. The *initial condition* $y(x_0) = y_0$ identifies two constants x_0, y_0 to be used for the first table pair X, Y. For example, $y(0) = 2$ identifies first table pair $X = 0$, $Y = 2$.

Second table entry. The second table pair X, Y is computed from the first table pair x_0, y_0 and a **recurrence**. The X-value is given by $X = x_0 + h$, while the Y-value is given by the numerical integration method being used, in accordance with Table 1 (the table is justified on page 234).

Table 1. Three numerical integration methods.

Rectangular Rule	$Y = y_0 + hF(x_0)$
Trapezoidal Rule	$Y = y_0 + \frac{h}{2}(F(x_0) + F(x_0 + h))$
Simpson's Rule	$Y = y_0 + \frac{h}{6}(F(x_0) + 4F(x_0 + h/2) + F(x_0 + h)))$

Third and higher table entries. They are computed by letting x_0, y_0 be the current table entry, then the next table entry X, Y is found exactly as outlined above for the second table entry.

It is expected, and normal, to compute the table entries using computer assist. In simple cases, a calculator will suffice. If F is complicated or Simpson's rule is used, then a computer algebra system or a numerical laboratory is recommended. See Example 2, page 228.

How to make a connect-the-dots graphic. To illustrate, consider the xy-pairs below, which are to represent the *dots* in the *connect-the-dots* graphic.

$$(0.0, 2.000), (0.1, 1.901), (0.2, 1.808), (0.3, 1.727), (0.4, 1.664),$$
$$(0.5, 1.625), (0.6, 1.616), (0.7, 1.643), (0.8, 1.712), (0.9, 1.829),$$
$$(1.0, 2.000).$$

Hand drawing. The method, unchanged from high school mathematics courses, is to plot the points as dots on an xy-coordinate system, then connect the dots with line segments. See Figure 2.

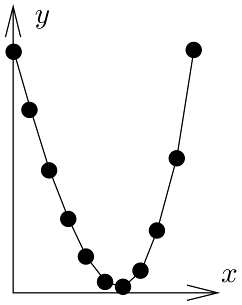


Figure 2. A Connect-the-Dots Graphic. A computer-generated graphic made to simulate a hand-drawn graphic, with straight lines between dots.

Computer Algebra System Graphic

Computer algebra system `maple`. It has a primitive syntax especially made for connect-the-dots graphics. Below, `Dots` is a list of xy-pairs.

```
Dots:=[0.0, 2.000], [0.1, 1.901], [0.2, 1.808],
      [0.3, 1.727], [0.4, 1.664], [0.5, 1.625],
      [0.6, 1.616], [0.7, 1.643], [0.8, 1.712],
      [0.9, 1.829], [1.0, 2.000]:
plot([Dots]);
```

The plotting of *points only* can be accomplished by adding options into the `plot` command: `type=point` and `symbol=circle` will suffice.

Computer algebra system maxima. The plot primitive can be invoked with x-array and y-array, or else pairs as in the `maple` example above:

```
Dots:[[0.0, 2.000], [0.1, 1.901], [0.2, 1.808],[0.3, 1.727],
 [0.4, 1.664],[0.5, 1.625],[0.6, 1.616], [0.7, 1.643],
 [0.8, 1.712],[0.9, 1.829], [1.0,2.000]];
 plot2d([discrete,Dots]);
```

Numerical Laboratory Graphic

The computer programs `matlab`, `octave` and `scilab` provide primitive plotting facilities, as follows.

```
X=[0,.1,.2,.3,.4,.5,.6,.7,.8,.9,1]
Y=[2.000, 1.901, 1.808, 1.727, 1.664, 1.625,
   1.616, 1.643, 1.712, 1.829, 2.000]
plot(X,Y)
```

1 **Example (Rectangular Rule)** Consider $y' = 3x^2 - 2x$, $y(0) = 0$. Apply the rectangular rule to make an xy-table for $y(x)$ from $x = 0$ to $x = 2$ in steps of $h = 0.2$. Graph the approximate solution and the exact solution $y(x) = x^3 - x^2$ for $0 \le x \le 2$.

Solution: The exact solution $y = x^3 - x^2$ is verified directly, by differentiation. It was obtained by quadrature applied to $y' = 3x^2 - 2x$, $y(0) = 0$.

The first table entry 0, 0 is used to obtain the second table entry $X = 0.2$, $Y = 0$ as follows.

$x_0 = 0$, $y_0 = 0$	The current table entry, row 1.
$X = x_0 + h$	The next table entry, row 2.
$\quad = 0.2,$	Use $x_0 = 0$, $h = 0.2$.
$Y = y_0 + hF(x_0)$	Rectangular rule.
$\quad = 0 + 0.2(0).$	Use $h = 0.2$, $F(x) = 3x^2 - 2x$.

The remaining 9 rows of the table are completed by calculator, following the pattern above for the second table entry. The result:

Table 2. Rectangular rule solution and exact values for $y' = 3x^2 - 2x$, $y(0) = 0$ on $0 \le x \le 2$, step size $h = 0.2$.

x	y-rect	y-exact	x	y-rect	y-exact
0.0	0.000	0.000	1.2	0.120	0.288
0.2	0.000	−0.032	1.4	0.504	0.784
0.4	−0.056	−0.096	1.6	1.120	1.536
0.6	−0.120	−0.144	1.8	2.016	2.592
0.8	−0.144	−0.128	2.0	3.240	4.000
1.0	−0.080	0.000			

The xy-values from the table are used to obtain the comparison plot in Figure 3.

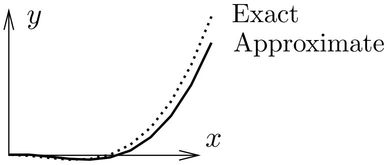


Figure 3. Comparison Plot. Rectangular rule numerical solution and the exact solution for $y = x^3 - x^2$ for $y' = 3x^2 - 2x$, $y(0) = 0$.

2 **Example (Trapezoidal Rule)** Consider $y' = \cos x + 2x$, $y(0) = 0$. Apply both the rectangular and trapezoidal rules to make an xy-table for $y(x)$ from $x = 0$ to $x = \pi$ in steps of $h = \pi/10$. Compare the two approximations in a graphic for $0 \le x \le \pi$.

Solution: The exact solution $y = \sin x + x^2$ is verified directly, by differentiation. It will be seen that the trapezoidal solution is nearly identical, graphically, to the exact solution.

The table will have 11 rows. The three columns are x, y-rectangular and y-trapezoidal. The first table entry 0, 0, 0 is used to obtain the second table entry 0.1π, 0.31415927, 0.40516728 as follows.

Rectangular rule second entry.

$$\begin{aligned} Y &= y_0 + hF(x_0) && \text{Rectangular rule.} \\ &= 0 + h(\cos 0 + 2(0)) && \text{Use } F(x) = \cos x + 2x,\ x_0 = y_0 = 0. \\ &= 0.31415927. && \text{Use } h = 0.1\pi = 0.31415927. \end{aligned}$$

Trapezoidal rule second entry.

$$\begin{aligned} Y &= y_0 + 0.5h(F(x_0) + F(x_0 + h)) && \text{Trapezoidal rule.} \\ &= 0 + 0.05\pi(\cos 0 + \cos h + 2h) && \text{Use } x_0 = y_0 = 0,\ F(x) = \cos x + 2x. \\ &= 0.40516728. && \text{Use } h = 0.1\pi. \end{aligned}$$

The remaining 9 rows of the table are completed by calculator, following the pattern above for the second table entry. The result:

Table 3. Rectangular and trapezoidal solutions for $y' = \cos x + 2x$, $y(0) = 0$ on $0 \le x \le \pi$, step size $h = 0.1\pi$.

x	y-rect	y-trap
0.000000	0.000000	0.000000
0.314159	0.314159	0.405167
0.628319	0.810335	0.977727
0.942478	1.459279	1.690617
1.256637	2.236113	2.522358
1.570796	3.122762	3.459163
1.884956	4.109723	4.496279
2.199115	5.196995	5.638458
2.513274	6.394081	6.899490
2.827433	7.719058	8.300851
3.141593	9.196803	9.869604

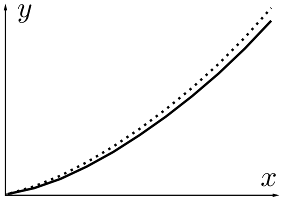


Figure 4. Comparison Plot. Rectangular (solid) and trapezoidal (dotted) numerical solutions for $y' = \cos x + 2x$, $y(0) = 0$ for $h = 0.1\pi$ on $0 \le x \le \pi$.

Computer algebra system. The `maple` implementation for Example 2 appears below. The code produces lists `Dots1` and `Dots2` which contain Rectangular (left panel) and Trapezoidal (right panel) approximations.

```
# Rectangular algorithm
# Group 1, initialize.
F:=x->evalf(cos(x) + 2*x):
x0:=0:y0:=0:h:=0.1*Pi:
Dots1:=[x0,y0]:

# Group 2, loop count = 10
for i from 1 to 10 do
Y:=y0+h*F(x0):
x0:=x0+h:y0:=evalf(Y):
Dots1:=Dots1,[x0,y0];
end do;

# Group 3, plot.
plot([Dots1]);
```

```
# Trapezoidal algorithm
# Group 1, initialize.
F:=x->evalf(cos(x) + 2*x):
x0:=0:y0:=0:h:=0.1*Pi:
Dots2:=[x0,y0]:

# Group 2, repeat 10 times
for i from 1 to 10 do
Y:=y0+h*(F(x0)+F(x0+h))/2:
x0:=x0+h:y0:=evalf(Y):
Dots2:=Dots2,[x0,y0];
end do;

# Group 3, plot.
plot([Dots2]);
```

3 Example (Simpson's Rule) Consider $y' = e^{-x^2}$, $y(0) = 0$. Apply both the rectangular and Simpson rules to make an xy-table for $y(x)$ from $x = 0$ to $x = 1$ in steps of $h = 0.1$. In the table, include values for the exact solution $y(x) = \frac{\sqrt{\pi}}{2}\operatorname{erf}(x)$. Compare the two approximations in a graphic for $0.8 \le x \le 1.0$.

Solution: The **error function** $\operatorname{erf}(x) = \frac{2}{\sqrt{\pi}}\int_0^x e^{-t^2}dt$ is a library function available in `maple`, `mathematica`, `matlab` and other computing platforms. It is known that the integral cannot be expressed in terms of elementary functions.

The xy-table. There will be 11 rows, for $x = 0$ to $x = 1$ in steps of $h = 0.1$. There are four columns: x, y-rectangular, y-Simpson, y-exact.

The first row of the table is created from $y(0) = 0$, details below.

It will be shown how to obtain the first and second rows by calculator methods, for the two algorithms *rectangular* and *Simpson*.

Rectangular rule first entry.
Let $x_0 = 0$ and $y_0 = 0$, from $y(0) = 0$, which means $y = 0$ when $x = 0$. The first table pair is (x_0, y_0).

Rectangular rule second entry. The second table pair is (x_1, y_1).

$$\begin{aligned}
x_1 &= x_0 + h && \text{Equal divisions.}\\
y_1 &= y_0 + hF(x_0) && \text{Rectangular rule.}\\
&= 0 + h(e^0) && \text{Use } F(x) = e^{-x^2},\ x_0 = y_0 = 0.\\
&= 0.1. && \text{Use } h = 0.1.
\end{aligned}$$

Simpson rule first entry.
Let $x_0 = 0$ and $y_0 = 0$, from $y(0) = 0$, which means $y = 0$ when $x = 0$. The first table pair is (x_0, y_0).

Simpson rule second entry. The second table pair is (x_1, y_1).

$$\begin{aligned}
x_1 &= x_0 + h && \text{Equal divisions.}\\
y_1 &= y_0 + \tfrac{h}{6}(F(x_0) + 4F(x_0 + h/2) + F(x_0 + h)) && \text{Simpson rule.}\\
&= 0 + \tfrac{0.1}{6}(e^0 + 4e^{.5} + e^{.1}) && \text{Use } F(x) = e^{-x^2},\ x_0 = y_0 = 0,\ h = 0.1.\\
&= 0.09966770540. && \text{Calculator.}
\end{aligned}$$

Exact solution second entry.
The numerical work requires the tabulated function erf(x). The `maple` details:

```
x0:=0:y0:=0:h:=0.1:            Given.
c:=sqrt(Pi)/2                  Conversion factor.
Exact:=x->y0+c*erf(x):         Exact solution y = y0 + ∫_0^x e^{-t^2} dt.
Y3:=Exact(x0+h);               Calculate exact answer.
# Y3 := .09966766428
```

Table 4. Rectangular and Simpson Rule.
Numerical solutions for $y' = e^{-x^2}$, $y(0) = 0$ on $0 \le x \le \pi$, step size $h = 0.1$.

x	y-rect	y-Simp	y-exact
0.0	0.00000000	0.00000000	0.00000000
0.1	0.10000000	0.09966771	0.09966766
0.2	0.19900498	0.19736511	0.19736503
0.3	0.29508393	0.29123799	0.29123788
0.4	0.38647705	0.37965297	0.37965284
0.5	0.47169142	0.46128114	0.46128101
0.6	0.54957150	0.53515366	0.53515353
0.7	0.61933914	0.60068579	0.60068567
0.8	0.68060178	0.65766996	0.65766986
0.9	0.73333102	0.70624159	0.70624152
1.0	0.77781682	0.74682418	0.74682413

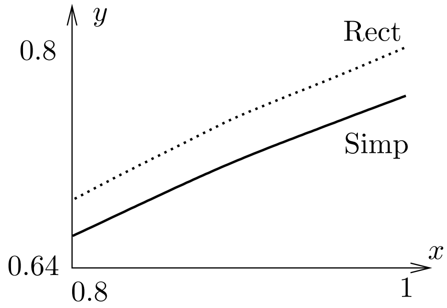


Figure 5. Comparison Plot.

Rectangular (dotted) and Simpson (solid) numerical solutions for $y' = e^{-x^2}$, $y(0) = 0$ for $h = 0.1$ on $0.8 \le x \le 1.0$.

Computer algebra system. The `maple` implementation for Example 3 appears below. The code produces two lists `Dots1` and `Dots2` which contain Rectangular (left panel) and Simpson (right panel) approximations.

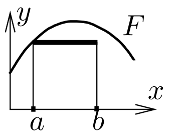
```
# Rectangular algorithm
# Group 1, initialize.
F:=x->evalf(exp(-x*x)):
x0:=0:y0:=0:h:=0.1:
Dots1:=[x0,y0]:

# Group 2, repeat 10 times
for i from 1 to 10 do
Y:=evalf(y0+h*F(x0)):
x0:=x0+h:y0:=Y:
Dots1:=Dots1,[x0,y0];
end do;


# Group 3, plot.
plot([Dots1]);
```

```
# Simpson algorithm
# Group 1, initialize.
F:=x->evalf(exp(-x*x)):
x0:=0:y0:=0:h:=0.1:
Dots2:=[x0,y0]:

# Group 2, loop count = 10
for i from 1 to 10 do
Y:=evalf(y0+h*(F(x0)+
   4*F(x0+h/2)+F(x0+h))/6):
x0:=x0+h:y0:=Y:
Dots2:=Dots2,[x0,y0];
end do;

# Group 3, plot.
plot([Dots2]);
```

Review of Numerical Integration

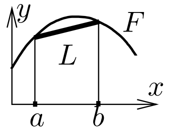
Reproduced here are calculus topics: the **rectangular rule**, the **trapezoidal rule** and **Simpson's rule** for the numerical approximation of an integral $\int_a^b F(x)dx$. The approximations are valid for $b - a$ small. Larger intervals must be subdivided, then the rule applies to the small subdivisions.

Rectangular Rule. The approximation uses Euler's idea of replacing the integrand by a constant. The value of the integral is approximately the area of a rectangle of width $b - a$ and height $F(a)$.

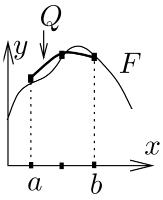


(2)
$$\int_a^b F(x)dx \approx (b-a)F(a).$$

Trapezoidal Rule. The rule replaces the integrand $F(x)$ by a linear function $L(x)$ which connects the planar points $(a, F(a))$, $(b, F(b))$. The value of the integral is approximately the area under the curve L, which is the area of a trapezoid.



(3)
$$\int_a^b F(x)dx \approx \frac{b-a}{2}\left(F(a)+F(b)\right).$$



Simpson's Rule. The rule replaces the integrand $F(x)$ by a quadratic polynomial $Q(x)$ which connects the planar points $(a, F(a))$, $((a+b)/2, F((a+b)/2))$, $(b, F(b))$. The value of the integral is approximately the area under the quadratic curve Q.

$$\int_a^b F(x)dx \approx \frac{b-a}{6}\left(F(a) + 4F\left(\frac{a+b}{2}\right) + F(b)\right). \tag{4}$$

Simpson's Polynomial Rule. If $Q(x)$ is constant, or a linear, quadratic or cubic polynomial, then (proof on page 232)

$$\int_a^b Q(x)dx = \frac{b-a}{6}\left(Q(a) + 4Q\left(\frac{a+b}{2}\right) + Q(b)\right). \tag{5}$$

Integrals of linear, quadratic and cubic polynomials can be evaluated *exactly* using Simpson's polynomial rule (5); see Example 4, page 232.

Remarks on Simpson's Rule. The right side of (4) is exactly the integral of $Q(x)$, which is evaluated by equation (5). The appearance of F instead of Q on the right in equation (4) is due to the relations $Q(a) = F(a)$, $Q((a+b)/2) = F((a+b)/2)$, $Q(b) = F(b)$, which arise from the requirement that Q connect three points along curve F.

The quadratic interpolation polynomial $Q(x)$ is determined uniquely from the three data points; see *Quadratic Interpolant*, page 233, for a formula for Q and a derivation. It is interesting that Simpson's rule depends only upon the uniqueness and not upon the actual formula for Q!

4 Example (Polynomial Quadrature) Apply Simpson's polynomial rule (5) to verify $\int_1^2 (x^3 - 16x^2 + 4)dx = -355/12$.

Solution: The application proceeds as follows:

$I = \int_1^2 Q(x)dx$	Evaluate integral I using $Q(x) = x^3 - 16x^2 + 4$.
$= \dfrac{2-1}{6}\left(Q(1) + 4Q(3/2) + Q(2)\right)$	Apply Simpson's polynomial rule (5).
$= \dfrac{1}{6}\left(-11 + 4(-229/8) - 52\right)$	Use $Q(x) = x^3 - 16x^2 + 4$.
$= -\dfrac{355}{12}$.	Equality verified.

Simpson's Polynomial Rule Proof. Let $Q(x)$ be a linear, quadratic or cubic polynomial. It will be verified that

$$\int_a^b Q(x)dx = \frac{b-a}{6}\left(Q(a) + 4Q\left(\frac{a+b}{2}\right) + Q(b)\right). \tag{6}$$

If the formula holds for polynomial Q and c is a constant, then the formula also holds for the polynomial cQ. Similarly, if the formula holds for polynomials Q_1 and Q_2, then it also holds for $Q_1 + Q_2$. Consequently, it suffices to show that the formula is true for the special polynomials 1, x, x^2 and x^3, because then it holds for all combinations $Q(x) = c_0 + c_1x + c_2x^2 + c_3x^3$.

Only the special case $Q(x) = x^3$ will be treated here. The other cases are left to the exercises. The details:

$\text{RHS} = \frac{b-a}{6}\left(Q(a) + 4Q\left(\frac{a+b}{2}\right) + Q(b)\right)$	Evaluate the right side of equation (6).
$= \frac{b-a}{6}\left(a^3 + \frac{1}{2}(a+b)^3 + b^3\right)$	Substitute $Q(x) = x^3$.
$= \frac{b-a}{6}\left(\frac{3}{2}\right)\left(a^3 + a^2b + ab^2 + b^3\right)$	Expand $(a+b)^3$. Simplify.
$= \frac{1}{4}\left(b^4 - a^4\right),$	Multiply and simplify.
$\text{LHS} = \int_a^b Q(x)dx$	Evaluate the left hand side (LHS) of equation (6).
$= \int_a^b x^3dx$	Substitute $Q(x) = x^3$.
$= \frac{1}{4}\left(b^4 - a^4\right)$	Evaluate.
$= \text{RHS}.$	Compare with the RHS.

This completes the proof of Simpson's polynomial rule.

Quadratic Interpolant Q.

Given $a < b$ and the three data points (a, Y_0), $((a+b)/2, Y_1))$, $(b, Y_2))$, then there is a unique quadratic curve $Q(X)$ which connects the points, given by

$$Q(X) = Y_0 + (4Y_1 - Y_2 - 3Y_0)\frac{X-a}{b-a} + (2Y_2 + 2Y_0 - 4Y_1)\frac{(X-a)^2}{(b-a)^2}. \tag{7}$$

Proof: The term *quadratic* is meant loosely: it can be a constant or linear function as well.

Uniqueness of the interpolant Q is established by subtracting two candidates to obtain a polynomial P of degree at most two which vanishes at three distinct points. By Rolle's theorem, P' vanishes at two distinct points and hence P'' vanishes at one point. Writing $P(X) = c_0 + c_1X + c_2X^2$ shows $c_2 = 0$ and then $c_1 = c_0 = 0$, or briefly, $P \equiv 0$. Hence the two candidates are identical.

It remains to verify the given formula (7). The details are presented as two lemmas.[1] The first lemma contains the essential ideas. The second simply translates the variables.

[1] What's a lemma? It's a helper theorem, used to dissect long proofs into short pieces.

Lemma 1 Given y_1 and y_2, define $A = y_2 - y_1$, $B = 2y_1 - y_2$. Then the quadratic $y = x(Ax + B)$ fits the data items $(0, 0)$, $(1, y_1)$, $(2, 2y_2)$.

Lemma 2 Given Y_0, Y_1 and Y_2, define $y_1 = Y_1 - Y_0$, $y_2 = \frac{1}{2}(Y_2 - Y_0)$, $A = y_2 - y_1$, $B = 2y_1 - y_2$ and $x = 2(X - a)/(b - a)$. Then quadratic $Y(X) = Y_0 + x(Ax + B)$ fits the data items (a, Y_0), $((a + b)/2, Y_1)$, (b, Y_2).

To verify the first lemma, the formula $y = x(Ax + B)$ is tested to go through the given data points $(0, 0)$, $(1, y_1)$ and $(2, 2y_2)$. For example, the last pair is tested by the steps

$y(2) = 2(2A + B)$	Apply $y = x(Ax + B)$ with $x = 2$.
$= 4y_2 - 4y_1 + 4y_1 - 2y_2$	Use $A = y_2 - y_1$ and $B = 2y_1 - y_2$.
$= 2y_2$.	Therefore, the quadratic fits data item $(2, 2y_2)$.

The other two data items are tested similarly, details omitted here.

To verify the second lemma, observe that it is just a change of variables in the first lemma, $Y = Y_0 + y$. The data fit is checked as follows:

$Y(b) = Y_0 + y(2)$	Apply formulas $Y(X) = Y_0 + y(x)$, $y(x) = x(Ax + B)$ with $X = b$ and $x = 2$.
$= Y_0 + 2y_2$	Apply data fit $y(2) = 2y_2$.
$= Y_2$.	The quadratic fits the data item (b, Y_2).

The other two items are checked similarly, details omitted here. This completes the proof of the two lemmas. The formula for Q is obtained from the second lemma as $Q = Y_0 + Bx + Ax^2$ with substitutions for A, B and x performed to obtain the given equation for Q in terms of Y_0, Y_1, Y_2, a, b and X.

Justification of Table 1: The method of quadrature applied to $y' = F(x)$, $y(x_0) = y_0$ gives an explicit solution $y(x)$ involving the integral of F. Specialize this solution formula to $x = x_0 + h$ where $h > 0$. Then

$$y(x_0 + h) = y_0 + \int_{x_0}^{x_0+h} F(t)dt.$$

All three methods in Table 1 are derived by replacment of the integral above by the corresponding approximation taken from the rectangular, trapezoidal or Simpson method on page 231. For example, the trapezoidal method gives

$$\int_{x_0}^{x_0+h} F(t)dt \approx \frac{h}{2}\left(F(x_0) + F(x_0 + h)\right),$$

whereupon replacement into the formula for y gives the entry in Table 1 as

$$Y \approx y(x_0 + h) \approx y_0 + \frac{h}{2}\left(F(x_0) + F(x_0 + h)\right).$$

This completes the justification of Table 1.

Exercises 4.1

Connect-the-Dots. Make a numerical table of 6 rows and a connect-the-dots graphic for the following.

1. $y = 2x + 5$, $x = 0$ to $x = 1$

2. $y = 3x + 5$, $x = 0$ to $x = 2$

3. $y = 2x^2 + 5$, $x = 0$ to $x = 1$

4. $y = 3x^2 + 5$, $x = 0$ to $x = 2$

5. $y = \sin x$, $x = 0$ to $x = \pi/2$

6. $y = \sin 2x$, $x = 0$ to $x = \pi/4$

7. $y = x \ln|1 + x|$, $x = 0$ to $x = 2$

8. $y = x \ln|1 + 2x|$, $x = 0$ to $x = 1$

9. $y = xe^x$, $x = 0$ to $x = 1$

10. $y = x^2 e^x$, $x = 0$ to $x = 1/2$

Rectangular Rule. Apply the rectangular rule to make an xy-table for $y(x)$ with 11 rows and step size $h = 0.1$. Graph the approximate solution and the exact solution. Follow example 1.

11. $y' = 2x$, $y(0) = 5$.

12. $y' = 3x^2$, $y(0) = 5$.

13. $y' = 3x^2 + 2x$, $y(0) = 4$.

14. $y' = 3x^2 + 4x^3$, $y(0) = 4$.

15. $y' = \sin x$, $y(0) = 1$.

16. $y' = 2\sin 2x$, $y(0) = 1$.

17. $y' = \ln(1 + x)$, $y(0) = 1$. Exact $(1 + x)\ln|1 + x| + 1 - x$.

18. $y' = 2\ln(1 + 2x)$, $y(0) = 1$. Exact $(1 + 2x)\ln|1 + 2x| + 1 - 2x$.

19. $y' = xe^x$, $y(0) = 1$. Exact $xe^x - e^x + 2$.

20. $y' = 2x^2 e^{2x}$, $y(0) = 4$. Exact $2\,x^2 e^x - 4\,xe^x + 4\,e^x$.

Trapezoidal Rule. Apply the trapezoidal rule to make an xy-table for $y(x)$ with 6 rows and step size $h = 0.2$. Graph the approximate solution and the exact solution. Follow example 2.

21. $y' = 2x$, $y(0) = 1$.

22. $y' = 3x^2$, $y(0) = 1$.

23. $y' = 3x^2 + 2x$, $y(0) = 2$.

24. $y' = 3x^2 + 4x^3$, $y(0) = 2$.

25. $y' = \sin x$, $y(0) = 4$.

26. $y' = 2\sin 2x$, $y(0) = 4$.

27. $y' = \ln(1 + x)$, $y(0) = 1$. Exact $(1 + x)\ln|1 + x| + 1 - x$.

28. $y' = 2\ln(1 + 2x)$, $y(0) = 1$. Exact $(1 + 2x)\ln|1 + 2x| + 1 - 2x$.

29. $y' = xe^x$, $y(0) = 1$. Exact $xe^x - e^x + 2$.

30. $y' = 2x^2 e^{2x}$, $y(0) = 4$. Exact $2\,x^2 e^x - 4\,xe^x + 4\,e^x$.

Simpson Rule. Apply Simpson's rule to make an xy-table for $y(x)$ with 6 rows and step size $h = 0.2$. Graph the approximate solution and the exact solution. Follow example 3.

31. $y' = 2x$, $y(0) = 2$.

32. $y' = 3x^2$, $y(0) = 2$.

33. $y' = 3x^2 + 2x$, $y(0) = 3$.

34. $y' = 3x^2 + 4x^3$, $y(0) = 3$.

35. $y' = \sin x$, $y(0) = 5$.

36. $y' = 2\sin 2x$, $y(0) = 5$.

37. $y' = \ln(1 + x)$, $y(0) = 1$. Exact $(1 + x)\ln|1 + x| + 1 - x$.

38. $y' = 2\ln(1 + 2x)$, $y(0) = 1$. Exact $(1 + 2x)\ln|1 + 2x| + 1 - 2x$.

39. $y' = xe^x$, $y(0) = 1$. Exact $xe^x - e^x + 2$.

40. $y' = 2x^2e^{2x}$, $y(0) = 4$. Exact $2\,x^2e^x - 4\,xe^x + 4\,e^x$.

Simpson's Rule. The following exercises use formulas and techniques found in the proof on page 232 and in Example 4, page 232.

41. Verify with Simpson's rule (5) for cubic polynomials the equality $\int_1^2(x^3 + 16x^2 + 4)dx = 541/12$.

42. Verify with Simpson's rule (5) for cubic polynomials the equality $\int_1^2(x^3 + x + 14)dx = 77/4$.

43. Let $f(x)$ satisfy $f(0) = 1$, $f(1/2) = 6/5$, $f(1) = 3/4$. Apply Simpson's rule with one division to verify that $\int_0^1 f(x)dx \approx 131/120$.

44. Let $f(x)$ satisfy $f(0) = -1$, $f(1/2) = 1$, $f(1) = 2$. Apply Simpson's rule with one division to verify that $\int_0^1 f(x)dx \approx 5/6$.

45. Verify Simpson's equality (5), assuming $Q(x) = 1$ and $Q(x) = x$.

46. Verify Simpson's equality (5), assuming $Q(x) = x^2$.

Quadratic Interpolation. The following exercises use formulas and techniques from the proof on page 233.

47. Verify directly that the quadratic polynomial $y = x(7 - 4x)$ goes through the points $(0, 0)$, $(1, 3)$, $(2, -2)$.

48. Verify directly that the quadratic polynomial $y = x(8 - 5x)$ goes through the points $(0, 0)$, $(1, 3)$, $(2, -4)$.

49. Compute the quadratic interpolation polynomial $Q(x)$ which goes through the points $(0, 1)$, $(0.5, 1.2)$, $(1, 0.75)$.

50. Compute the quadratic interpolation polynomial $Q(x)$ which goes through the points $(0, -1)$, $(0.5, 1)$, $(1, 2)$.

51. Verify the remaining cases in Lemma 1, page 234.

52. Verify the remaining cases in Lemma 2, page 234.

4.2 Solving $y' = f(x, y)$ Numerically

The numerical solution of the initial value problem

$$y'(x) = f(x, y(x)), \quad y(x_0) = y_0 \tag{1}$$

is studied here by three basic methods. In each case, the current table entry x_0, y_0 plus step size h is used to find the next table entry X, Y. *Define* $X = x_0 + h$ and let Y be defined below, according to the algorithm selected (Euler, Heun, RK4)[2]. The *motivation* for the three methods appears on page 242.

Euler's method.

$$Y = y_0 + hf(x_0, y_0). \tag{2}$$

Heun's method.

$$\begin{aligned} y_1 &= y_0 + hf(x_0, y_0), \\ Y &= y_0 + \frac{h}{2}\left(f(x_0, y_0) + f(x_0 + h, y_1)\right). \end{aligned} \tag{3}$$

Runge-Kutta RK4 method.

$$\begin{aligned} k_1 &= hf(x_0, y_0), \\ k_2 &= hf(x_0 + h/2, y_0 + k_1/2), \\ k_3 &= hf(x_0 + h/2, y_0 + k_2/2), \\ k_4 &= hf(x_0 + h, y_0 + k_3), \\ Y &= y_0 + \frac{k_1 + 2k_2 + 2k_3 + k_4}{6}. \end{aligned} \tag{4}$$

The last quantity Y contains an average of six terms, where two appear in duplicate: $(k_1 + k_2 + k_2 + k_3 + k_3 + k_4)/6$. A similar average appears in Simpson's rule.

Relationship to calculus methods. If the differential equation (1) is specialized to the equation $y'(x) = F(x)$, $y(x_0) = y_0$, to agree with the previous section, then $f(x, y) = F(x)$ is independent of y and the three methods of Euler, Heun and RK4 reduce to the rectangular, trapezoidal and Simpson rules.

To justify the reduction in the case of Heun's method, start with the assumption $f(x, y) = F(x)$ and observe that by independence of y, variable y_1 is never used. Compute as follows:

$$\begin{aligned} Y &= y_0 + \tfrac{h}{2}\left(f(x_0, y_0) + f(x_0 + h, y_1)\right) && \text{Apply equation (3).} \\ &= y_0 + \tfrac{h}{2}\left(F(x_0) + F(x_0 + h)\right). && \text{Use } f(x, y) = F(x). \end{aligned}$$

The right side of the last equation is exactly the trapezoidal rule.

[2] Euler is pronounced *oiler*. Heun rhymes with *coin*. Runge rhymes with *run key*.

Examples and Methods

5 **Example (Euler's Method)** Solve $y' = -y + 1 - x$, $y(0) = 3$ by Euler's method for $x = 0$ to $x = 1$ in steps of $h = 0.1$. Produce a table of values which compares approximate and exact solutions. Graph both the exact solution $y = 2 - x + e^{-x}$ and the approximate solution.

Solution: **Exact solution**. The homogeneous solution is $y_h = ce^{-x}$. A particular solution $y_p = 2 - x$ is found by the method of undetermined coefficients or the linear integrating factor method. The general solution $y_h + y_p$ is then $y(x) = ce^{-x} + 2 - x$. Initial condition $y(0) = 3$ gives $c = 1$ and then $y = 2 - x + e^{-x}$.

Approximate Solution. The table of xy-values starts because of $y(0) = 3$ with the two values $X = 0$, $Y = 3$. Throughout, $f(x, y) = -y+1-x$ = RHS of the differential equation. The X-values will be $X = 0$ to $X = 1$ in increments of $h = 1/10$, making 11 rows total. The Y-values are computed from

$$\begin{aligned} Y &= y_0 + hf(x_0, y_0) && \text{Euler's method.} \\ &= y_0 + h(-y_0 + 1 - x_0) && \text{Use } f(x,y) = -y + 1 - x. \\ &= 0.9y_0 + 0.1(1 - x_0) && \text{Use } h = 0.1. \end{aligned}$$

The pair x_0, y_0 represents the two entries in the current row of the table. The next table pair X, Y is given by $X = x_0+h$, $Y = 0.9y_0+0.1(1-x_0)$. It is normal in a computation to do the *second pair* by hand, then use computing machinery to reproduce the hand result and finish the computation of the remaining table rows. Here's the second pair:

$$\begin{aligned} X &= x_0 + h && \text{Definition of } X\text{-values.} \\ &= 0.1, && \text{Substitute } x_0 = 0 \text{ and } h = 0.1. \\ Y &= 0.9y_0 + 0.1(1 - x_0), && \text{The simplified recurrence.} \\ &= 0.9(3) + 0.1(1 - 0) && \text{Substitute for row 1, } x_0 = 0,\ y_0 = 3. \\ &= 2.8. && \text{Second row found: } X = 0.1,\ Y = 2.8. \end{aligned}$$

By the same process, the third row is $X = 0.2$, $Y = 2.61$. This gives the xy-table below, in which the exact values from $y = 2 - x + e^{-x}$ are also tabulated.

Table 5. Euler's method applied with $h = 0.1$ on $0 \le x \le 1$ to the problem $y' = -y + 1 - x$, $y(0) = 3$.

x	y-Euler	y-Exact	x	y-Euler	y-Exact
0.0	3.00000	3.0000000	0.6	1.93144	1.9488116
0.1	2.80000	2.8048374	0.7	1.77830	1.7965853
0.2	2.61000	2.6187308	0.8	1.63047	1.6493290
0.3	2.42900	2.4408182	0.9	1.48742	1.5065697
0.4	2.25610	2.2703200	1.0	1.34868	1.3678794
0.5	2.09049	2.1065307			

See page 240 for `maple` code which automates Euler's method. The approximate solution graphed in Figure 6 is nearly identical to the exact solution $y = 2 - x + e^{-x}$. The `maple` plot code for Figure 6:

```
L:=[0.0,3.00000],[0.1,2.80000],[0.2,2.61000],[0.3,2.42900],
   [0.4,2.25610],[0.5,2.09049],[0.6,1.93144],[0.7,1.77830],
   [0.8,1.63047],[0.9,1.48742],[1.0,1.34868]:
plot({[L],2-x+exp(-x)},x=0..1);
```

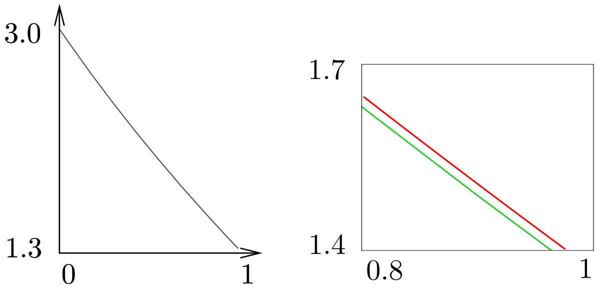


Figure 6. Euler approximate solution on $[0, 1]$ for $y' = -y + 1 - x$, $y(0) = 3$ is the curve on the left. The exact solution $y = 2 - x + e^{-x}$ is the upper curve on the right. The approximate solution is the lower curve on the right.

6 **Example (Euler and Heun Methods)** Solve $y' = -y + 1 - x$, $y(0) = 3$ by both Euler's method and Heun's method for $x = 0$ to $x = 1$ in steps of $h = 0.1$. Produce a table of values which compares approximate and exact solutions.

Solution: **Table of xy-values**. The Euler method was applied in Example 5. Heun's method will be documented here. The first pair is 0, 3. The second pair X, Y will be computed by hand calculation below. Throughout, $f(x, y) = -y + 1 - x =$ RHS of the differential equation.

$$\begin{aligned}
X &= x_0 + h && \text{Definition of } X\text{-values.} \\
&= 0.1, && \text{Substitute } x_0 = 0 \text{ and } h = 0.1. \\
Y_1 &= y_0 + hf(x_0, y_0) && \text{First Heun formula.} \\
&= y_0 + 0.1(-y_0 + 1 - x_0) && \text{Use } f(x, y) = -y + 1 - x. \\
&= 2.8, && \text{Row 1 gives } x_0,\ y_0. \text{ Same as the Euler method value.} \\
Y &= y_0 + h(f(x_0, y_0) + f(x_0 + h, Y_1))/2, && \text{Second Heun formula.} \\
&= 3 + 0.05(-3 + 1 - 0 - 2.8 + 1 - 0.1) && \text{Use } x_0 = 0,\ y_0 = 3,\ Y_1 = 2.8. \\
&= 2.805. &&
\end{aligned}$$

Therefore, the second row is $X = 0.1$, $Y = 2.805$. By the same process, the third row is $X = 0.2$, $Y = 2.619025$. This gives the xy-table below, in which the Euler approximate values and the exact values from $y = 2 - x + e^{-x}$ are also tabulated, taken from the preceding example.

Table 6. Euler and Heun methods applied with $h = 0.1$ on $0 \le x \le 1$ to the problem $y' = -y + 1 - x$, $y(0) = 3$.

x	y-Euler	y-Heun	y-Exact
0.0	3.00000	3.00000	3.0000000
0.1	2.80000	2.80500	2.8048374
0.2	2.61000	2.61903	2.6187308
0.3	2.42900	2.44122	2.4408182
0.4	2.25610	2.27080	2.2703200
0.5	2.09049	2.10708	2.1065307
0.6	1.93144	1.94940	1.9488116
0.7	1.77830	1.79721	1.7965853
0.8	1.63047	1.64998	1.6493290
0.9	1.48742	1.50723	1.5065697
1.0	1.34868	1.36854	1.3678794

Computer algebra system. The implementation for `maple` appears below. Part of the interface is execution of a group, which is used here to divide the algorithm into three distinct parts. The code produces a list `L` which contains Euler (left panel) or Heun (right panel) approximations.

```
# Euler algorithm
# Group 1, initialize.
f:=(x,y)->-y+1-x:
x0:=0:y0:=3:h:=.1:L:=[x0,y0]:
# Group 2, loop count = 10
for i from 1 to 10 do
Y:=y0+h*f(x0,y0):
x0:=x0+h:y0:=Y:L:=L,[x0,y0];
end do;
# Group 3, plot.
plot([L]);
```

```
# Heun algorithm
# Group 1, initialize.
f:=(x,y)->-y+1-x:
x0:=0:y0:=3:h:=.1:L:=[x0,y0]
# Group 2, loop count = 10
for i from 1 to 10 do
Y:=y0+h*f(x0,y0):
Y:=y0+h*(f(x0,y0)+f(x0+h,Y))/2:
x0:=x0+h:y0:=Y:L:=L,[x0,y0];
end do;
# Group 3, plot.
plot([L]);
```

Numerical laboratory. The implementation of the Heun method for `matlab`, `octave` and `scilab` will be described. The code is written into files `f.m` and `heun.m`, which must reside in a default directory. Then `[X,Y]=heun(0,3,1,10)` produces the xy-table. The graphic is made with `plot(X,Y)`.

File `f.m`:

```
function yp = f(x,y)
yp= -y+1-x;
```

File `heun.m`:

```
function [X,Y] = heun(x0,y0,x1,n)
h=(x1-x0)/n;X=x0;Y=y0;
for i=1:n;
y1= y0+h*f(x0,y0);
y0= y0+h*(f(x0,y0)+f(x0+h,y1))/2;
x0=x0+h;
X=[X;x0];Y=[Y;y0];
end
```

7 Example (Euler, Heun and RK4 Methods) Solve the initial value problem $y' = -y + 1 - x$, $y(0) = 3$ by Euler's method, Heun's method and the

RK4 method for $x = 0$ to $x = 1$ in steps of $h = 0.1$. Produce a table of values which compares approximate and exact solutions.

Solution: **Table of xy-values**. The Euler and Heun methods were applied in Examples 5, 6. The Runge-Kutta method (RK4) will be illustrated here. The first pair is 0, 3. The second pair X, Y will be computed by hand calculator.

$$\begin{aligned}
X &= x_0 + h && \text{Definition of } X\text{-values.}\\
&= 0.1, && \text{Substitute } x_0 = 0 \text{ and } h = 0.1.\\
k_1 &= hf(x_0, y_0) && \text{First RK4 formula.}\\
&= 0.1(-y_0 + 1 - x_0) && \text{Use } f(x,y) = -y + 1 - x.\\
&= -0.2, && \text{Row 1 supplies } x_0 = 0,\ y_0 = 3.\\
k_2 &= hf(x_0 + h/2, y_0 + k_1/2) && \text{Second RK4 formula.}\\
&= 0.1f(0.05, 2.9) &&\\
&= -0.195, &&\\
k_3 &= hf(x_0 + h/2, y_0 + k_2/2) && \text{Third RK4 formula.}\\
&= 0.1f(0.05, 2.9025) &&\\
&= -0.19525, &&\\
k_4 &= hf(x_0 + h, y_0 + k_3) && \text{Fourth RK4 formula.}\\
&= 0.1f(0.1, 2.80475) &&\\
&= -0.190475, &&\\
Y &= y_0 + \tfrac{1}{6}(k_1 + 2k_2 + 2k_2 + k_4), && \text{Last RK4 formula.}\\
&= 3 + \tfrac{1}{6}(-1.170975) && \text{Use } x_0 = 0,\ y_0 = 3,\ Y_1 = 2.8.\\
&= 2.8048375. &&
\end{aligned}$$

Therefore, the second row is $X = 0.1$, $Y = 2.8048375$. Continuing, the third row is $X = 0.2$, $Y = 2.6187309$. The Euler and Heun steps were done in the previous example and recorded in Table 6. We have computed by hand calculator the first three rows of the computer-generated xy-table below, in which exact values $y = 2 - x + e^{-x}$ are also tabulated.

Table 7. Euler, Heun and RK4 methods applied with $h = 0.1$ on $0 \le x \le 1$ to the problem $y' = -y + 1 - x$, $y(0) = 3$.

x	y-Euler	y-Heun	y-RK4	y-Exact
0.0	3.00000	3.00000	3.0000000	3.0000000
0.1	2.80000	2.80500	2.8048375	2.8048374
0.2	2.61000	2.61903	2.6187309	2.6187308
0.3	2.42900	2.44122	2.4408184	2.4408182
0.4	2.25610	2.27080	2.2703203	2.2703200
0.5	2.09049	2.10708	2.1065309	2.1065307
0.6	1.93144	1.94940	1.9488119	1.9488116
0.7	1.77830	1.79721	1.7965856	1.7965853
0.8	1.63047	1.64998	1.6493293	1.6493290
0.9	1.48742	1.50723	1.5065700	1.5065697
1.0	1.34868	1.36854	1.3678798	1.3678794

Computer algebra system. The implementation of RK4 for `maple` appears below, as a modification of the code for Example 6.

```
# Group 2, loop count = 10
for i from 1 to 10 do
k1:=h*f(x0,y0):
k2:=h*f(x0+h/2,y0+k1/2):
k3:=h*f(x0+h/2,y0+k2/2):
k4:=h*f(x0+h,y0+k3):
Y:=y0+(k1+2*k2+2*k3+k4)/6:
x0:=x0+h:y0:=Y:L:=L,[x0,y0];
end do;
```

The reader is requested to verify that this code in the special case $f(x, y) = F(x)$ (independent of y) reduces to a poor implementation of Simpson's Rule for $\int_a^{a+h} F(x)dx$. The wasted effort is calculation of k_3, because k_2, k_3 are the same for $f(x, y) = F(x)$.

Numerical laboratory. The implementation of RK4 for `matlab`, `octave` and `scilab` appears below, to be added to the code for Example 6. The code is written into file `rk4.m`, which must reside in a default directory. Then `[X,Y]=rk4(0,3,1,10)` produces the xy-table.

```
function [X,Y] = rk4(x0,y0,x1,n)
h=(x1-x0)/n;X=x0;Y=y0;
for i=1:n;
 k1=h*f(x0,y0);
 k2=h*f(x0+h/2,y0+k1/2);
 k3=h*f(x0+h/2,y0+k2/2);
 k4=h*f(x0+h,y0+k3);
 y0=y0+(k1+2*k2+2*k3+k4)/6;
 x0=x0+h;
 X=[X;x0];Y=[Y;y0];
end
```

Motivation for the three methods. The entry point to the study is the equivalent integral equation

$$y(x) = y_0 + \int_{x_0}^{x} f(t, y(t))dt. \tag{5}$$

The ideas can be explained by replacement of the integral in (5) by the rectangular, trapezoidal or Simpson rule. Unknown values of y that appear are subsequently replaced by suitable approximations.

These approximations, originating with L. Euler, are known as **predictors** and **correctors**. They are defined as follows from the integral formula

$$y(b) = y(a) + \int_a^b f(x, y(x))dx, \tag{6}$$

by assuming the integrand is a constant C.

Predictor $Y = y(a) + (b-a)f(a,Y^*)$. Given an estimate or an exact value Y^* for $y(a)$, then variable Y predicts $y(b)$. The approximation assumes the integrand in (6) constantly $C = f(a,Y^*)$.

Corrector $Y = y(a) + (b-a)f(b,Y^{**})$. Given an estimate or an exact value Y^{**} for $y(b)$, then variable Y corrects $y(b)$. The approximation assumes the integrand in (6) constantly $C = f(b,Y^{**})$.

Euler's method. Replace in (5) $x = x_0 + h$ and apply the rectangular rule to the integral. The resulting approximation is known as **Euler's method**:

$$y(x_0+h) \approx Y = y_0 + hf(x_0,y_0). \tag{7}$$

Heun's method. Replace in (5) $x = x_0 + h$ and apply the trapezoidal rule to the integral, to get

$$y(x_0+h) \approx y_0 + \frac{h}{2}\left(f(x_0,y(x_0) + f(x_0+h,y(x_0+h))\right).$$

The troublesome expressions are $y(x_0)$ and $y(x_0+h)$. The first is y_0. The second can be estimated by the **predictor** $y_0 + hf(x_0,y_0)$. The resulting approximation is known as **Heun's method** or the **modified Euler method**:

$$\begin{aligned} &Y_1 = y_0 + hf(x_0,y_0), \\ &y(x_0+h) \approx Y = y_0 + \frac{h}{2}\left(f(x_0,y_0) + f(x_0+h,Y_1)\right). \end{aligned} \tag{8}$$

RK4 method. Replace in (5) $x = x_0 + h$ and apply Simpson's rule to the integral. This gives $y(x_0+h) \approx y_0 + S$ where the Simpson estimate S is given by

$$S = \frac{h}{6}\left(f(x_0,y(x_0) + 4f(M,y(M)) + f(x_0+h,y(x_0+h))\right) \tag{9}$$

and $M = x_0 + h/2$ is the midpoint of $[x_0, x_0+h]$. The troublesome expressions in S are $y(x_0)$, $y(M)$ and $y(x_0+h)$. The work of Runge and Kutta shows that

- Expression $y(x_0)$ is replaced by y_0.
- Expression $y(M)$ can be replaced by either Y_1 or Y_2, where $Y_1 = y_0 + 0.5hf(x_0,y_0)$ is a **predictor** and $Y_2 = y_0 + 0.5hf(M,Y_1)$ is a **corrector**.
- Expression $y(x_0+h)$ can be replaced by $Y_3 = y_0 + hf(M,Y_2)$. This replacement arises from the **predictor** $y(x_0+h) \approx y(M) + 0.5hf(M,y(M))$ by using **corrector** $y(M) \approx y_0 + 0.5hf(M,y(M))$ and then replacing $y(M)$ by Y_2.

The formulas of Runge-Kutta result by using the above replacements for $y(x_0)$, $y(M)$ and $y(x_0+h)$, with the caveat that $f(M, y(M))$ gets replaced by the **average** of $f(M, Y_1)$ and $f(M, Y_2)$. In detail,

$$\begin{aligned}6S &= hf(x_0, y(x_0) + 4hf(M, y(M)) + hf(x_0+h, y(x_0+h))\\ &\approx hf(x_0, y_0) + 4h\frac{f(M,Y_1)+f(M,Y_2)}{2} + hf(x_0+h, Y_3)\\ &= k_1 + 2k_2 + 2k_3 + k_4\end{aligned}$$

where the RK4 quantities k_1, k_2, k_3, k_4 are defined by (4), page 237. The resulting approximation is known as the **RK4 method**.

Exercises 4.2

Euler's Method. Apply Euler's method to make an xy-table for $y(x)$ with 11 rows and step size $h = 0.1$. Graph the approximate solution and the exact solution. Follow Example 5.

1. $y' = 2 + y$, $y(0) = 5$. Exact $y(x) = -2 + 7e^x$.

2. $y' = 3 + y$, $y(0) = 5$. Exact $y(x) = -3 + 8e^x$.

3. $y' = e^{-x} + y$, $y(0) = 4$. Exact $y(x) = -\frac{1}{2}e^{-x} + \frac{9}{2}e^x$.

4. $y' = 3e^{-2x} + y$, $y(0) = 4$. Exact $y(x) = -e^{-2x} + 5e^x$.

5. $y' = y\sin x$, $y(0) = 1$. Exact $y(x) = e^{1-\cos x}$.

6. $y' = 2y\sin 2x$, $y(0) = 1$. Exact $y(x) = e^{1-\cos 2x}$.

7. $y' = y/(1+x)$, $y(0) = 1$. Exact $y(x) = 1 + x$.

8. $y' = y(x)/(1+2x)$, $y(0) = 1$. Exact $y(x) = \sqrt{1+2x}$.

9. $y' = yxe^x$, $y(0) = 1$. Exact $y(x) = e^{u(x)}$, $u(x) = 1 + (x-1)e^x$.

10. $y' = 2y(x^2+x)e^{2x}$, $y(0) = 1$. Exact $y(x) = e^{u(x)}$, $u(x) = x^2e^{2x}$.

Heun's Method. Apply Heun's method to make an xy-table for $y(x)$ with 6 rows and step size $h = 0.2$. Graph the approximate solution and the exact solution. Follow Example 6.

11. $y' = 2 + y$, $y(0) = 5$. Exact $y(x) = -2 + 7e^x$.

12. $y' = 3 + y$, $y(0) = 5$. Exact $y(x) = -3 + 8e^x$.

13. $y' = e^{-x} + y$, $y(0) = 4$. Exact $y(x) = -\frac{1}{2}e^{-x} + \frac{9}{2}e^x$.

14. $y' = 3e^{-2x} + y$, $y(0) = 4$. Exact $y(x) = -e^{-2x} + 5e^x$.

15. $y' = y\sin x$, $y(0) = 1$. Exact $y(x) = e^{1-\cos x}$.

16. $y' = 2y\sin 2x$, $y(0) = 1$. Exact $y(x) = e^{1-\cos 2x}$.

17. $y' = y/(1+x)$, $y(0) = 1$. Exact $y(x) = 1 + x$.

18. $y' = y(x)/(1+2x)$, $y(0) = 1$. Exact $y(x) = \sqrt{1+2x}$.

19. $y' = yxe^x$, $y(0) = 1$. Exact $y(x) = e^{u(x)}$, $u(x) = 1 + (x-1)e^x$.

20. $y' = 2y(x^2+x)e^{2x}$, $y(0) = 1$. Exact $y(x) = e^{u(x)}$, $u(x) = x^2e^{2x}$.

RK4 Method. Apply the Runge-Kutta method (RK4) to make an xy-table for $y(x)$ with 6 rows and step size $h = 0.2$. Graph the approximate solution and the exact solution. Follow Example 7.

21. $y' = 2 + y$, $y(0) = 5$. Exact $y(x) = -2 + 7e^x$.

22. $y' = 3 + y$, $y(0) = 5$. Exact $y(x) = -3 + 8e^x$.

23. $y' = e^{-x} + y$, $y(0) = 4$. Exact $y(x) = -\frac{1}{2}e^{-x} + \frac{9}{2}e^x$.

24. $y' = 3e^{-2x} + y$, $y(0) = 4$. Exact $y(x) = -e^{-2x} + 5e^x$.

25. $y' = y \sin x$, $y(0) = 1$. Exact $y(x) = e^{1-\cos x}$.

26. $y' = 2y \sin 2x$, $y(0) = 1$. Exact $y(x) = e^{1-\cos 2x}$.

27. $y' = y/(1 + x)$, $y(0) = 1$. Exact $y(x) = 1 + x$.

28. $y' = y(x)/(1 + 2x)$, $y(0) = 1$. Exact $y(x) = \sqrt{1 + 2x}$.

29. $y' = yxe^x$, $y(0) = 1$. Exact $y(x) = e^{u(x)}$, $u(x) = 1 + (x - 1)e^x$.

30. $y' = 2y(x^2 + x)e^{2x}$, $y(0) = 1$. Exact $y(x) = e^{u(x)}$, $u(x) = x^2e^{2x}$.

Euler and RK4 Methods. Apply the Euler method and the Runge-Kutta method (RK4) to make a table with 6 rows and step size $h = 0.1$. The table columns are x, y_1, y_2, y where y_1 is the Euler approximation, y_2 is the RK4 approximation and y is the exact solution. Graph y_1, y_2, y.

31. $y' = \frac{1}{2}(y - 2)^2$, $y(0) = 3$. Exact $y(x) = \frac{2x-6}{x-2}$.

32. $y' = \frac{1}{2}(y - 3)^2$, $y(0) = 4$. Exact $y(x) = \frac{3x-8}{x-2}$.

33. $y' = x^3/y^2$, $y(2) = 3$. Exact $y(x) = \frac{1}{2}\sqrt[3]{6x^4 + 120}$.

34. $y' = x^5/y^2$, $y(2) = 3$. Exact $y(x) = \frac{1}{2}\sqrt[3]{4x^6 - 40}$.

35. $y' = 2x(1 + y^2)$, $y(1) = 1$. Exact $y(x) = \tan(x^2 - 1 + \pi/4)$.

36. $y' = 3y^{2/3}$, $y(0) = 1$. Exact $y(x) = (x + 1)^3$.

37. $y' = 1 + y^2$, $y(0) = 0$. Exact $y(x) = \tan x$.

38. $y' = 1 + y^2$, $y(0) = 1$. Exact $y(x) = \tan(x + \pi/4)$.

4.3 Error in Numerical Methods

Numerical Errors

Studied here are *cumulative error*, *local error*, *roundoff error* and *truncation error*. The Landau order notation is introduced.

Cumulative Error. This error measurement is commonly used in displays like Table 8, in which approximate and exact solution columns already appear. In such applications, the cumulative error is the difference of the approximate and exact columns. The **exact solution** refers to $y(x)$ defined by $y' = f(x, y)$, $y(x_0) = y_0$ ($x_0 = 0$, $y_0 = 3$ from line 1 of Table 8). The **approximate solution** refers to the y-values computed by the algorithm (column 2 in Table 8). A precise definition of the **cumulative error** E is given in terms of the exact solution $y(x)$: *given table entry* X, Y, *then* $E = |y(X) - Y|$.

Table 8. Cumulative error.
A third column, cumulative error, is added to an existing xy-table of approximate and exact solutions. The cumulative error is computed by the formula $E = |y_2 - y_1|$, where y_1 is the approximation and y_2 is the exact value.

x	y-Approx	y-Exact	Error
0.0	3.00000	3.0000000	0.0000000
0.1	2.80000	2.8048374	0.0048374
0.2	2.61000	2.6187308	0.0087308
0.3	2.42900	2.4408182	0.0118182

Local Error. This error is made by one algorithm step in going from table entry x_1, y_1 to the next table entry x_2, y_2. It can be precisely defined in terms of the solution $u(x)$ to $u' = f(x, u)$, $u(x_1) = y_1$ by the formula

$$E_{\text{loc}} = |u(x_2) - y_2|.$$

Noteworthy is that $u(x) \neq y(x)$. To explain, the exact solution $y(x)$ solves $y' = f(x, y)$, $y(x_0) = y_0$ where x_0, y_0 is the *first table entry*, while $u(x)$ solves $u' = f(x, u)$ for a *different* set of initial conditions. In particular, an xy-table of approximate and exact solution values, like Table 8, does not contain enough information to determine the local error!

To illustrate the ideas, consider $y' = 2y$, $y(0) = 1$ with exact solution

$y = e^{2x}$. Using Euler's method with step size $h = 0.1$ gives the table

x	y-approx	y-exact
0	1	1
0.1	1.2	1.2214028
0.2	1.44	1.4918247

To find the local error for line 2 to line 3 requires solving $u' = 2u$, $u(0.1) = 1.2$, and then evaluating $E = |u(0.2) - 1.4918247|$. We find that $u(x) = 1.2e^{2(x-0.1)}$ and then $E = |1.2e^{0.2} - 1.4918247| = 0.026141390$.

Roundoff Error. Also called rounding error, the roundoff error is the difference between the calculated approximation of a number to finitely many digits and its exact value in terms of infinitely many digits. The technical error is made by computers due to the representation of floating point numbers, which limits the number of significant digits in any computation. *Integer arithmetic* will normally generate no errors, unless **integer overflow** occurs, i.e., $x+y$ or xy can result in an integer larger than the machine can represent. *Floating point arithmetic* usually generates errors because of results that must be rounded to give a machine representation. To illustrate, 8-digit precision requires $a = 1.00000005$ be represented as $\hat{a} = 1.0000001$ and $b = 1.00000004$ be represented as $\hat{b} = 1$. Then $2a + 2b = 4.00000018$, which rounds to 4.0000002, while $2\hat{a} + 2\hat{b} = 4.0000001$. The roundoff error in this example is 0.0000001.

For numerical methods, this translates into *fewer* roundoff errors for $h = 0.1$ than for $h = 0.001$, because the number of arithmetic operations increases 1000-fold for $h = 0.001$. The payoff in increased accuracy expected for a change in step size from $h = 0.1$ to $h = 0.001$ may be less than theoretically possible, because the roundoff errors accumulate to cancel the effects of decreased step size. Positive and negative roundoff errors tend to cancel, leading to situations where a thousand-fold step size change causes only a thirty-fold change in roundoff error.

Truncation Error. It is typical in numerical mathematics to use formulas like $\pi = 3.14159$ or $e = 2.718$. These formulas **truncate** the actual decimal expansion, causing an error. **Truncation** is the term used for reducing the number of digits to the right of the decimal point, by discarding all digits past a certain point, e.g., 0.123456789 truncated to 5 digits is 0.12345. Common truncation errors are caused by dropping higher order terms in a Taylor series, or by approximating a nonlinear term by its linearization. In general, a truncation error is made whenever a formula is replaced by an approximate formula, in which case the formula is wrong even if computed exactly.

Landau Symbol. Edmund Landau, a German mathematician, introduced a convenient notation to represent truncation errors. If f and g are defined near $h = 0$, then $f = \mathbf{O}(g)$ means that $|f(h)| \le K|g(h)|$ as $h \to 0$, for some constant K. The **Landau notation** $f = \mathbf{O}(g)$ is vocalized as "*f equals big owe of g*." The symbol $\mathbf{O}(h^n)$ therefore stands for terms or order h^n. Taylor series expansions can then be referenced succinctly, e.g., $\sin h = h + \mathbf{O}(h^3)$, $e^h = 1 + h + \mathbf{O}(h^2)$, and so on. Some simple rules for the Landau symbol:

$$\mathbf{O}(h^n) + \mathbf{O}(h^m) = \mathbf{O}(h^{\min(n,m)}), \quad \mathbf{O}(h^n)\mathbf{O}(h^m) = \mathbf{O}(h^{n+m}).$$

Finite Blowup of Solutions. The solution $y = (1-x)^{-1}$ for $y' = y^2$, $y(0) = 1$ exists on $0 \le x < 1$, but it becomes infinite at $x = 1$. The finite value $x = 1$ causes blowup of the y-value. This event is called **finite blowup**. Attempts to solve $y' = y^2$, $y(0) = 1$ numerically will fail near $x = 1$, and these errors will propagate past $x = 1$, if the numerical problem is allowed to be solved over an interval larger than $0 \le x < 1$.

Unfortunately, finite blowup cannot be detected in advance from smoothness of $f(x, y)$ or the fact that the problem is *applied.* For example, logistic population models $y' = y(a - by)$ typically have solutions with finite blowup, because the solution y is a fraction which can have a zero denominator at some instant x . On the positive side, there are three common conditions which guarantee no finite blowup:

- A linear equation $y' + p(x)y = q(x)$ does not exhibit finite blowup on the domain of continuity of $p(x)$ and $q(x)$.
- An equation $y' = f(x, y)$ does not exhibit finite blowup if f is continuous and $\max |f_y(x, y)| < \infty$.
- An equation $y' = f(x, y)$ does not exhibit finite blowup if f is continuous and f satisfies a Lipschitz condition $|f(x, y_1) - f(x, y_2)| \le M|y_1 - y_2|$ for some constant $M > 0$ and all x, y_1, y_2.

Numerical Instability. The equation $y' = y + 1 - x$ has solution $y = x + ce^x$. Attempts to solve for $y(0) = 1$ will meet with failure, because errors will cause the numerical solution to lock onto some solution with $c \ne 0$ and small, which causes the numerical solution to grow like e^x. In this case, the instability was caused by the problem itself.

Numerical instability can result even though the solution is physically stable. An example is $y' = -50(y - \sin x) + \cos x$, $y(0) = 0$. The general solution is $y = ce^{-50x} + \sin x$ and $y(0) = 0$ gives $c = 0$. The negative exponential term is *transient* and $\sin x$ is the unique periodic *steady-state* solution. The solution is insensitive to minor changes in the initial condition. For popular numerical methods, the value at $x = 1$ seems to depend greatly on the step size, as is shown by Table 9.

Table 9. Cumulative error at $x = 1$ for Euler, Heun and RK4 methods applied to $y' = -50(y - \sin x) + \cos x$, $y(0) = 0$, for various step sizes.

	$h = 0.1$	$h = 0.05$	$h = 0.02$	$h = 0.01$
Euler	40701.23	0.183e7	0.00008	0.00004
Heun	0.328e12	0.430e14	0.005	0.00004
RK4	0.318e20	0.219e18	0.00004	0.000001

The sensitivity to step size is due to the *algorithm* and not to instability of the problem.

Stiff Problems. The differential equation $y' = -50(y - \sin x) + \cos x$, which has solution $y = ce^{-50x} + \sin x$, is called **stiff**, a technical term defined precisely in advanced numerical analysis references, e.g., Burden-Faires [**?**]. Characteristically, it means that the equation has a solution $y(x)$ containing a transient term $y_1(x)$ with derivative $y_1'(x)$ tending slowly to zero. For instance, if $y(x)$ has a term like $y_1(x) = ce^{-50x}$, then the derivative $y_1'(x)$ is approximately 50 times larger ($y_1'/y_1 \approx -50$). Applications with transient terms of Landau order e^{-at} are stiff when a is large. Stiff problems occupy an active branch of research in applied numerical analysis. Researchers call a problem **stiff** provided certain numerical methods for it are unstable (e.g., inaccurate) unless the step size is taken to be extremely small.

Cumulative Error Estimates

It is possible to give theoretical but not practical estimates for the cumulative error in the case of Euler's method, Heun's method and the RK4 method. Applied literature and computer documentation often contain references to these facts, typically in the following succinct form.

- Euler's method has order 1.
- Heun's method has order 2.
- The Runge-Kutta method (RK4) has order 4.

The *exact meaning* of these statements is given below in the theorems. The phrase **order** n in this context refers to Edmund Landau's order notation $\mathbf{O}(h^n)$. In particular, *order* 2 means $\mathbf{O}(h^2)$.

In practical terms, the statements measure the quality and accuracy of the algorithms themselves, and hence establish an expectation of performance from each algorithm. They *do not mean* that step size $h = 0.001$

gives three digits of accuracy in the computed answer! The meaning is that repeated halving of the step size will result in three digits of accuracy, eventually. Most persons half the step size until the first three digits repeat, then they take this to be the optimal step size for three-digit accuracy. The theorems don't say that this practise is correct, only that for *some step size* it is correct.

Theorem 1 (Euler's Method Error)
Let the initial value problem $y' = f(x, y)$, $y(x_0) = y_0$ have a solution $y(x)$ in the region $x_0 \le x \le x_0 + H$, $|y - y_0| \le K$ and assume that f, f_x and f_y are continuous. Then the cumulative error $E(x_0 + nh)$ at step n, $nh \le H$, made by Euler's method using step size h satisfies $E(x_0 + nh) \le Ch$. The constant C depends only on x_0, y_0, H, K, f, f_x and f_y. See [?] and [?].

Theorem 2 (Heun Method Error)
Let the initial value problem $y' = f(x, y)$, $y(x_0) = y_0$ have a solution in the region $x_0 \le x \le x_0 + H$, $|y - y_0| \le K$. Assume f is continuous with continuous partials to order 3. Then the cumulative error $E(x_0 + nh)$ at step n, $nh \le H$, made by Heun's method using step size h, satisfies $E(x_0 + nh) \le Ch^2$. The constant C depends only on x_0, y_0, H, K, f and the partials of f to order 3.

Theorem 3 (RK4 Method Error)
Let the initial value problem $y' = f(x, y)$, $y(x_0) = y_0$ have a solution $y(x)$ in the region $x_0 \le x \le x_0 + H$, $|y - y_0| \le K$. Assume f is continuous with continuous partials to order 5. Then the cumulative error $E(x_0 + nh)$ at step n, $nh \le H$, made by the RK4 method using step size h, satisfies $E(x_0 + nh) \le Ch^4$. The constant C depends only on x_0, y_0, H, K, f, and the partials of f to order 5.

The last two results are implied by local truncation error estimates for Taylor's method of order n (section 5.3 in Burden-Faires [?]).

Exercises 4.3

Cumulative Error. Make a table of 6 lines which has four columns x, y_1, y, E. Symbols y_1 and y are the approximate and exact solutions while E is the cumulative error. Find y_1 using Euler's method in steps $h = 0.1$.

1. $y' = 2 + y$, $y(0) = 5$. Exact solution $y(x) = -2 + 7e^x$.

2. $y' = 3 + y$, $y(0) = 5$. Exact solution $y(x) = -3 + 8e^x$.

3. $y' = e^{-x} + y$, $y(0) = 4$. Exact solution $y(x) = -\frac{1}{2}e^{-x} + \frac{9}{2}e^x$.

4. $y' = 3e^{-2x} + y$, $y(0) = 4$. Exact solution $y(x) = -e^{-2x} + 5e^x$.

Local Error. Make a table of 4 lines which has four columns x, y_1, y, E. Symbols y_1 and y are the approximate and exact solutions while E is the local error. Find y_1 using Euler's method in steps $h = 0.1$. The general solution in each exercise is the solution for $y(0) = c$.

5. $y' = 2 + y$, $y(0) = 5$. General solution $y(x) = -2 + (2 + c)e^x$.

6. $y' = 3 + y$, $y(0) = 5$. General solution $y(x) = -3 + (3 + c)e^x$.

7. $y' = 2e^{-x} + y$, $y(0) = 4$. General solution $y(x) = -e^{-x} + (1 + c)e^x$.

8. $y' = 3e^{-2x} + y$, $y(0) = 4$. General solution $y(x) = -e^{-2x} + (1 + c)e^x$.

Roundoff Error. Compute the roundoff error for $y = 5a + 4b$.

9. Assume 3-digit precision. Let $a = 0.0001$ and $b = 0.0003$.

10. Assume 3-digit precision. Let $a = 0.0002$ and $b = 0.0001$.

11. Assume 5-digit precision. Let $a = 0.000007$ and $b = 0.000003$.

12. Assume 5-digit precision. Let $a = 0.000005$ and $b = 0.000001$.

Truncation Error. Find the truncation error.

13. Truncate $x = 1.123456789$ to 3 digits right of the decimal point.

14. Truncate $x = 1.123456789$ to 4 digits right of the decimal point.

15. Truncate $x = 1.017171717$ to 7 digits right of the decimal point.

16. Truncate $x = 1.03939393939$ to 9 digits right of the decimal point.

Guessing the Step Size. Do a numerical experiment to estimate the step size needed for 7-digit accuracy of the solution. Using the given method, report the step size, which if halved repeatedly, generates a numerical solution with 7-digit accuracy.

17. $y' = 2 + y$, $y(0) = 5$. Exact solution $y(x) = -2 + 7e^x$. Euler's method.

18. $y' = 3 + y$, $y(0) = 5$. Exact solution $y(x) = -3 + 8e^x$. Euler's method

19. $y' = e^{-x} + y$, $y(0) = 4$. Exact solution $y(x) = -\frac{1}{2}e^{-x} + \frac{9}{2}e^x$. Euler's method

20. $y' = 3e^{-2x} + y$, $y(0) = 4$. Exact solution $y(x) = -e^{-2x} + 5e^x$. Euler's method.

21. $y' = y/(1 + x)$, $y(0) = 1$. Exact solution $y(x) = 1 + x$. Euler's method.

22. $y' = y(x)/(1 + 2x)$, $y(0) = 1$. Exact solution $y(x) = \sqrt{1 + 2x}$. Euler's method.

23. $y' = 2 + y$, $y(0) = 5$. Exact solution $y(x) = -2 + 7e^x$. Heun's method.

24. $y' = 3 + y$, $y(0) = 5$. Exact solution $y(x) = -3 + 8e^x$. Heun's method

25. $y' = e^{-x} + y$, $y(0) = 4$. Exact solution $y(x) = -\frac{1}{2}e^{-x} + \frac{9}{2}e^x$. Heun's method

26. $y' = 3e^{-2x} + y$, $y(0) = 4$. Exact solution $y(x) = -e^{-2x} + 5e^x$. Heun's method.

27. $y' = y/(1 + x)$, $y(0) = 1$. Exact solution $y(x) = 1 + x$. Heun's method.

28. $y' = y(x)/(1 + 2x)$, $y(0) = 1$. Exact solution $y(x) = \sqrt{1 + 2x}$. Heun's method.

29. $y' = 2 + y$, $y(0) = 5$. Exact solution $y(x) = -2 + 7e^x$. RK4 method.

30. $y' = 3 + y$, $y(0) = 5$. Exact solution $y(x) = -3 + 8e^x$. RK4 method

31. $y' = e^{-x} + y$, $y(0) = 4$. Exact solution $y(x) = -\frac{1}{2}e^{-x} + \frac{9}{2}e^x$. RK4 method

32. $y' = 3e^{-2x} + y$, $y(0) = 4$. Exact solution $y(x) = -e^{-2x} + 5e^x$. RK4 method.

33. $y' = y/(1+x)$, $y(0) = 1$. Exact solution $y(x) = 1 + x$. RK4 method.

34. $y' = y(x)/(1 + 2x)$, $y(0) = 1$. Exact solution $y(x) = \sqrt{1 + 2x}$. RK4 method.

4.4 Computing π, $\ln 2$ and e

The approximations $\pi \approx 3.1415927$, $\ln 2 \approx 0.69314718$, $e \approx 2.7182818$ can be obtained by numerical methods applied to the following initial value problems:

$$y' = \frac{4}{1+x^2}, \quad y(0) = 0, \quad \pi = y(1), \tag{1}$$

$$y' = \frac{1}{1+x}, \quad y(0) = 0, \quad \ln 2 = y(1), \tag{2}$$

$$y' = y, \quad y(0) = 1, \quad e = y(1). \tag{3}$$

Equations (1)–(3) *define* the constants π, $\ln 2$ and e through the corresponding initial value problems.

The third problem (3) requires a numerical method like RK4, while the other two can be solved using Simpson's quadrature rule. It is a fact that RK4 reduces to Simpson's rule for $y' = F(x)$, therefore, for simplicity, RK4 can be used for all three problems, ignoring speed issues. It will be seen that the choice of the DE-solver algorithm (e.g., RK4) affects computational accuracy.

Computing $\pi = \int_0^1 4(1+x^2)^{-1}dx$

The easiest method is Simpson's rule. It can be implemented in virtually every computing environment. The code below works in popular `matlab`-compatible numerical laboratories. It modifies easily to other computing platforms, such as `maple` and `mathematica`. To obtain the answer for $\pi = 3.14159265358979323385$ correct to 12 digits, execute the code on the right in Table 10, below the definition of f.

Table 10. Numerical integration of $\int_0^1 4(1+x^2)^{-1}dx$.
Simpson's rule is applied, using `matlab`-compatible code. About 50 subdivisions are required.

```
function ans = simp(x0,x1,n,f)
h=(x1-x0)/n; ans=0;
for i=1:n;
ans1=f(x0)+4*f(x0+h/2)+f(x0+h);
ans=ans+(h/6)*ans1;
x0=x0+h;
end
```

```
function y = f(x)
y = 4/(1+x*x);





ans=simp(0,1,50,f)
```

It is convenient in some laboratories to display answers with `printf` or `fprintf`, in order to show 12 digits. For example, `scilab` prints 3.1415927 by default, but 3.141592653589800 using `printf`.

The results checked in `maple` give $\pi \approx 3.14159265358979323385$, accurate to 20 digits, regardless of the actual `maple` numerical integration algorithm chosen (three were possible). The checks are invoked by `evalf(X,20)` where `X` is replaced by `int(4/(1+x*x),x=0..1)`.

The results for an approximation to π using numerical solvers for differential equations varied considerably from one algorithm to another, although all were accurate to 5 rounded digits. A summary for `odepack` routines appears in Table 11, obtained from the `scilab` interface. A selection of routines supported by `maple` appear in Table 12. Default settings were used with no special attempt to increase accuracy.

The `Gear` routines refer to those in the 1971 textbook [?]. The Livermore stiff solver `lsode` can be found in reference [?]. The Runge-Kutta routine of order 7-8 called `dverk78` appears in the 1991 reference of Enright [?]. The multistep routines of Adams-Moulton and Adams-Bashforth are described in standard numerical analysis texts, such as [?]. Taylor series methods are described in [?]. The Fehlberg variant of RK4 is given in [?].

Table 11. Differential equation numeric solver results for `odepack` routines, applied to the problem $y' = 4/(1+x^2)$, $y(0)=0$.

Exact value of π	3.1415926535897932385	20 digits
Runge-Kutta 4	3.1415926535910	10 digits
Adams-Moulton lsode	3.1415932355842	6 digits
Stiff Solver lsode	3.1415931587318	5 digits
Runge-Kutta-Fehlberg 45	3.1416249508084	4 digits

Table 12. Differential equation numeric solver results for some `maple`-supported routines, applied to the problem $y' = 4/(1+x^2)$, $y(0)=0$.

Exact value of π	3.1415926535897932385	20 digits
Classical RK4	3.141592653589790	15 digits
Gear	3.141592653688446	11 digits
Dverk78	3.141592653607044	11 digits
Taylor Series	3.141592654	10 digits
Runge-Kutta-Fehlberg 45	3.141592674191119	8 digits
Multistep Gear	3.141591703761340	7 digits
Lsode stiff solver	3.141591733742521	6 digits

Computing $\ln 2 = \int_0^1 dx/(1+x)$

Like the problem of computing π, the formula for $\ln 2$ arises from the method of quadrature applied to $y' = 1/(1+x)$, $y(0)=0$. The solution is $y(x) = \int_0^x dt/(1+t)$. Application of Simpson's rule with 150 points gives $\ln 2 \approx 0.693147180563800$, which agrees with the exact value $\ln 2 = 0.69314718055994530942$ through 12 digits.

More robust numerical integration algorithms produce the exact answer for $\ln 2$, within the limitations of machine representation of numbers.

Differential equation methods, as in the case of computing π, have results accurate to at least 5 digits, as is shown in Tables 13 and 14. Lower order methods such as classical Euler will produce results accurate to three digits or less.

Table 13. Differential equation numeric solver results for `odepack` routines, applied to the problem $y' = 1/(1+x)$, $y(0) = 0$.

Exact value of $\ln 2$	0.69314718055994530942	20 digits
Adams-Moulton lsode	0.69314720834637	7 digits
Stiff Solver lsode	0.69314702723982	6 digits
Runge-Kutta 4	0.69314718056011	11 digits
Runge-Kutta-Fehlberg 45	0.69314973055488	5 digits

Table 14. Differential equation numeric solver results for `maple`-supported routines, applied to the problem $y' = 1/(1+x)$, $y(0) = 0$.

Exact value of $\ln 2$	0.69314718055994530942	20 digits
Classical Euler	0.6943987430550621	2 digits
Classical Heun	0.6931487430550620	5 digits
Classical RK4	0.6931471805611659	11 digits
Gear	0.6931471805646605	11 digits
Gear Poly-extr	0.6931471805664855	11 digits
Dverk78	0.6931471805696615	11 digits
Adams-Bashforth	0.6931471793736268	8 digits
Adams-Bashforth-Moulton	0.6931471806484283	10 digits
Taylor Series	0.6931471806	10 digits
Runge-Kutta-Fehlberg 45	0.6931481489496502	5 digits
Lsode stiff solver	0.6931470754312113	7 digits
Rosenbrock stiff solver	0.6931473787603164	6 digits

Computing e from $y' = y$, $y(0) = 1$

The initial attack on the problem uses classical RK4 with $f(x, y) = y$. After 300 steps, classical RK4 finds the correct answer for e to 12 digits: $e \approx 2.71828182846$. In Table 15, the details appear of how to accomplish the calculation using `matlab`-compatible code. Corresponding `maple` code appears in Table 16 and in Table 17. Additional code for `octave` and `scilab` appear in Tables 18 and 19.

Table 15. Numerical solution of $y' = y$, $y(0) = 1$.
Classical RK4 with 300 subdivisions using matlab-compatible code.

```
function [x,y]=rk4(x0,y0,x1,n,f)
x=x0;y=y0;h=(x1-x0)/n;
for i=1:n;
 k1=h*f(x,y);
 k2=h*f(x+h/2,y+k1/2);
 k3=h*f(x+h/2,y+k2/2);
 k4=h*f(x+h,y+k3);
 y=y+(k1+2*k2+2*k3+k4)/6;
 x=x+h;
end
```

```
function yp = ff(x,y)
 yp= y;


[x,y]=rk4(0,1,1,300,ff)
```

Table 16. Numerical solution of $y' = y$, $y(0) = 1$ by maple internal classical RK4 code.

```
de:=diff(y(x),x)=y(x):
ic:=y(0)=1:
Y:=dsolve({de,ic},y(x),
           type=numeric,method=classical[rk4]):
Y(1);
```

Table 17. Numerical solution of $y' = y$, $y(0) = 1$ by classical RK4 with 300 subdivisions using maple-compatible code.

```
rk4 := proc(x0,y0,x1,n,f)
local x,y,k1,k2,k3,k4,h,i:
x=x0:  y=y0:  h=(x1-x0)/n:
for i from 1 to n do
 k1:=h*f(x,y):k2:=h*f(x+h/2,y+k1/2):
 k3:=h*f(x+h/2,y+k2/2):k4:=h*f(x+h,y+k3):
 y:=evalf(y+(k1+2*k2+2*k3+k4)/6,Digits+4):
 x:=x+h:
od:
RETURN(y):
end:

f:=(x,y)->y;
rk4(0,1,1,300,f);
```

A `matlab` *m*-file `"rk4.m"` is loaded into `scilab`-4.0 by `getf("rk4.m")`. Most `scilab` code is loaded by using default file extension `.sci`, e.g., `rk4scilab.sci` is a `scilab` file name. This code must obey `scilab` rules. An example appears below in Table 18.

Table 18. Numerical solution of $y' = y$, $y(0) = 1$ by classical RK4 with 300 subdivisions, using `scilab`-4.0 code.

```
function
[x,y]=rk4sci(x0,y0,x1,n,f)
x=x0,y=y0,h=(x1-x0)/n
 for i=1:n
 k1=h*f(x,y)
 k2=h*f(x+h/2,y+k1/2)
 k3=h*f(x+h/2,y+k2/2)
 k4=h*f(x+h,y+k3)
 y=y+(k1+2*k2+2*k3+k4)/6
 x=x+h
 end
endfunction
```

```
function yp = ff(x,y)
 yp= y
endfunction

[x,y]=rk4sci(0,1,1,300,ff)
```

The popularity of `octave` as a free alternative to `matlab` has kept it alive for a number of years. Writing code for `octave` is similar to `matlab` and `scilab`, however readers are advised to look at sample code supplied with `octave` before trying complicated projects. In Table 19 can be seen some essential agreements and differences between the languages. Versions of `scilab` after 4.0 have a `matlab` to `scilab` code translator.

Table 19. Numerical solution of $y' = y$, $y(0) = 1$ by classical RK4 with 300 subdivisions using `octave`-2.1.

```
function
[x,y]=rk4oct(x0,y0,x1,n,f)
x=x0;y=y0;h=(x1-x0)/n;
 for i=1:n
 k1=h*feval(f,x,y);
 k2=h*feval(f,x+h/2,y+k1/2);
 k3=h*feval(f,x+h/2,y+k2/2);
 k4=h*feval(f,x+h,y+k3);
 y=y+(k1+2*k2+2*k3+k4)/6;
 x=x+h;
 endfor
endfunction
```

```
function yp = ff(x,y)
yp= y;
end

[x,y]=rk4oct(0,1,1,300,'ff')
```

Exercises 4.4

Computing π. Compute $\pi = y(1)$ from the initial value problem $y' = 4/(1 + x^2)$, $y(0) = 0$, using the given method.

1. Use the Rectangular integration rule. Determine the number of steps for 5-digit precision.

2. Use the Rectangular integration rule. Determine the number of steps for 8-digit precision.

3. Use the Trapezoidal integration rule. Determine the number of steps for 5-digit precision.

4. Use the Trapezoidal integration

rule. Determine the number of steps for 8-digit precision.

5. Use classical RK4. Determine the number of steps for 5-digit precision.

6. Use classical RK4. Determine the number of steps for 10-digit precision.

7. Use computer algebra system assist for RK4. Report the number of digits of precision using system defaults.

8. Use numerical workbench assist for RK4. Report the number of digits of precision using system defaults.

Computing $\ln(2)$. Compute $\ln(2) = y(1)$ from the initial value problem $y' = 1/(1+x)$, $y(0) = 0$, using the given method.

9. Use the Rectangular integration rule. Determine the number of steps for 5-digit precision.

10. Use the Rectangular integration rule. Determine the number of steps for 8-digit precision.

11. Use the Trapezoidal integration rule. Determine the number of steps for 5-digit precision.

12. Use the Trapezoidal integration rule. Determine the number of steps for 8-digit precision.

13. Use classical RK4. Determine the number of steps for 5-digit precision.

14. Use classical RK4. Determine the number of steps for 10-digit precision.

15. Use computer algebra system assist for RK4. Report the number of digits of precision using system defaults.

16. Use numerical workbench assist for RK4. Report the number of digits of precision using system defaults.

Computing e. Compute $e = y(1)$ from the initial value problem $y' = y$, $y(0) = 1$, using the given computer assist. Report the number of digits of precision using system defaults.

17. Improved Euler method, also known as Heun's method.

18. RK4 method.

19. RKF45 method.

20. Adams-Moulton method.

Stiff Differential Equation. The flame propagation equation $y' = y^2(1-y)$ is known to be **stiff** for initial conditions $y(0) = y_0$ with $y_0 > 0$ and small. Use classical RK4 and then a stiff solver to compute and plot the solution $y(t)$ in each case. Expect 3000 steps with RK4 versus 100 with a stiff solver.

The exact solution of this equation can be expressed in terms of the **Lambert function** $w(u)$, defined by $u = w(x)$ if and only if $ue^u = x$. For example, $y(0) = 0.01$ gives

$$y(t) = \frac{1}{w\left(99e^{99-t}\right) + 1}.$$

See R.M. Corless, G.H. Gonnet, D.E.G. Hare, D.J. Jeffrey, and D.E. Knuth. "On The Lambert W Function," Advances in Computational Mathematics 5 (1996): 329-359.

21. $y(0) = 0.01$

22. $y(0) = 0.005$

23. $y(0) = 0.001$

24. $y(0) = 0.0001$

4.5 Earth to the Moon

A projectile launched from the surface of the earth is attracted both by the earth and the moon. The altitude $r(t)$ of the projectile above the earth is known to satisfy the initial value problem (see *Technical Details* page 261)

$$\begin{aligned} &r''(t) = -\frac{Gm_1}{(R_1 + r(t))^2} + \frac{Gm_2}{(R_2 - R_1 - r(t))^2}, \\ &r(0) = 0, \quad r'(0) = v_0. \end{aligned} \tag{1}$$

The unknown initial velocity v_0 of the projectile is given in meters per second. The constants in (1) are defined as follows.

$G = 6.6726 \times 10^{-11}$ N-m^2/kg^2	Universal gravitation constant,
$m_1 = 5.975 \times 10^{24}$ kilograms	Mass of the earth,
$m_2 = 7.36 \times 10^{22}$ kilograms	Mass of the moon,
$R_1 = 6,378,000$ meters	Radius of the earth,
$R_2 = 384,400,000$ meters	Distance from the earth's center to the moon's center.

The Jules Verne Problem

In his 1865 novel *From the Earth to the Moon*, Jules Verne asked what initial velocity must be given to the projectile in order to reach the moon. The question in terms of equation (1) becomes:

> What minimal value of v_0 causes the projectile to have zero net acceleration at some point between the earth and the moon?

The projectile only has to travel a distance R equal to the surface-to-surface distance between the earth and the moon. The altitude $r(t)$ of the projectile must satisfy $0 \le r \le R$. Given v_0 for which the net acceleration is zero, $r''(t) = 0$ in (1), then the projectile has reached a critical altitude r^*, where gravitational effects of the moon take over and the projectile will fall to the surface of the moon.

Let $r''(t) = 0$ in (1) and substitute r^* for $r(t)$ in the resulting equation. Then

$$\begin{aligned} &-\frac{Gm_1}{(R_1 + r^*)^2} + \frac{Gm_2}{(R_2 - R_1 - r^*)^2} = 0, \\ &r^* = \frac{R_2}{1 + \sqrt{m_2/m_1}} - R_1 \approx 339,260,779 \text{ meters.} \end{aligned} \tag{2}$$

Using energy methods (see *Technical details*, page 262), it is possible to calculate exactly the *minimal* earth-to-moon velocity v_0^* required for the projectile to *just reach* critical altitude r^*:

$$v_0^* \approx 11067.19091 \quad \text{meters per second.} \tag{3}$$

A Numerical Experiment

The value $v_0^* \approx 11067.19091$ in (3) will be verified experimentally. As part of this experiment, the flight time is estimated.

Such a numerical experiment must adjust the initial velocity v_0 in initial value problem (1) so that $r(t)$ increases from 0 to R. Graphical analysis of a solution $r(t)$ for low velocities v_0 gives insight into the problem; see Figure 7.

The choice of numerical software solver makes for significant differences in this problem. Initial work used the Livermore Laboratory numerical stiff solver for ordinary differential equations (acronym `lsode`).

Computer algebra system `maple` documents and implements algorithm `lsode` with `dsolve` options of `method=lsode` or `stiff=true`. Other stiff solvers of equal quality can be used for nearly identical results. Experiments are necessary to determine if the required accuracy has been attained.

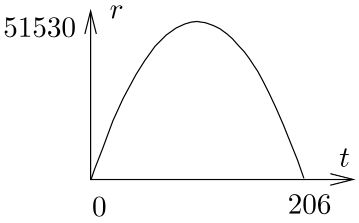


Figure 7. Jules Verne Problem. The solution $r(t)$ of (1) for $v_0 = 1000$. The projectile rises to a maximum height of about $51,530$ meters, then it falls back to earth. The trip time is 206 seconds.

The numerical experiment solves (1) using `lsode`, then the solution is graphed, to see if the projectile falls back to earth (as in Figure 7) or if it reaches an altitude near r^* and then falls to the moon. Suitable starting values for the initial velocity v_0 and the trip time T are $v_0 = 1000$ and $T = 210$ (see Figure 7), in the case when the projectile falls back to earth. The projectile travels to the moon when the r-axis of the graphic has maximum greater than $r^* \approx 339,260,779$ meters. The logic is that this condition causes the gravitation effects of the moon to be strong enough to force the projectile to fall to the moon.

In Table 20 appears `maple` initialization code. In Table 21, group 2 is executed a number of times, to refine estimates for the initial velocity v_0 and the trip time T. A summary of some estimates appear in Table 22. The graphics produced along the way resemble Figure 7 or Figure 8. A successful trip to the moon is represented in Figure 8, which uses $v_0 = 11068$ meters per second and $T = 515250$ seconds.

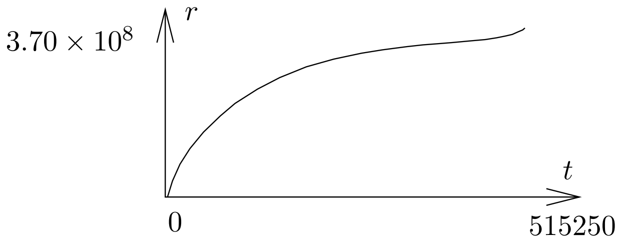


Figure 8. Experimental trip to the moon.
The initial velocity is $v_0 = 24,764$ miles per hour and the trip time is 143 hours. See Table 22 for details about how these values were obtained.

Table 20. Initialization code in `maple` for the numerical experiment.
Group 1 defines seven constants G, m_1, m_2, R_1, R_2, R_3, R and computes values $r^* \approx 339,260,779$ and $v_0^* \approx 11067.19091$.

```
# Group 1:  Constants plus rstar and v0star
G:=6.6726e-11:  m1:=5.975e24:  m2:=7.36e22:
R1:=6.378e6:  R2:=3.84e8:  R3:=1.74e6:
R:=R2-R1-R3:
ans:=[solve(-G*m1/(r+R1)^2 + G*m2/(R2-R1-r)^2=0,r)]:
rstar:=ans[1];
FF:=r->G*m1/(R1+r)+G*m2/(R2-R1-r):
v0star:=sqrt(2*(FF(0)-FF(rstar)));
```

Table 21. Iteration code in `maple` for the numerical experiment.
Group 2 plots a graphic for given v_0 and T. A successful trip to the moon must use velocity $v_0 > v_0^* \approx 11067.19091$. The relation $\max_{0 \le t \le T} Y(t) > r^* \approx 339,260,779$ must be valid. Finally, $Y(T) \ge R$ must hold.

```
# Group 2:  Iteration code
v0:=1000:  # v0<v0star.  Projectile falls to earth.
de:=diff(r(t),t,t)=-G*m1/(r(t)+R1)^2+G*m2/(R2-R1-r(t))^2:
ic:=r(0)=0,D(r)(0)=v0:
p:=dsolve({de,ic},r(t),
type=numeric,method=lsode,startinit=true);
Y:=t->rhs(p(t)[2]):
T:=200:  # Guess the trip time T
plot('Y(t)',t=0..T);
# Plot done.  Change v0, T and re-execute group 2.
```

Table 22. Experimental results with the `lsode` solver to obtain estimates for the initial velocity v_0 and the trip time T.

v_0	T	Results	
11000	38500	$r(T/2) = 1.872 \times 10^8$,	$r(T) = 0$
12000	80000	$r(T) > r^* \approx 3.39 \times 10^8$	
11125	200000	$r(T) > r^*$	
11060	780000	$r(T/2) = 2.918 \times 10^8$,	$r(T) = 0$
11070	377500	$r(T) > r^*$	
11068	515250	$r(T) \approx R$	

Exact trip time. The time T for a trip with velocity $v_0 = 11068$ can be computed once an approximate value for the trip time is known. For instance, if $T = 515250$ gives a successful plot, but $T = 515150$ does not, then the exact value of T is between 515250 and 515150. The computer algebra system can be used to determine the more precise value $T = 515206.1757$, as follows.

```
# Group 2
v0:=11068:  # Projectile reaches the moon.
de:=diff(r(t),t,t)=-G*m1/(r(t)+R1)^2
+G*m2/(R2-R1-r(t))^2:
ic:=r(0)=0,D(r)(0)=v0:
p:=dsolve({de,ic},r(t),
type=numeric,method=lsode,startinit=true);
Y:=t->rhs(p(t)[2]):
fsolve(’Y(t)’=R,t,515150..515250);
# T==515206.1757
```

Technical details for (1): To derive (1), it suffices to write down a competition between the Newton's second law force relation $mr''(t)$ and the sum of two forces due to gravitational attraction for the earth and the moon. Here, m stands for the mass of the projectile.

Gravitational force for the earth. This force, by Newton's universal gravitation law, has magnitude

$$F_1 = \frac{Gm_1m}{\mathcal{R}_3^2}$$

where m_1 is the mass of the earth, G is the universal gravitation constant and $\mathcal{R}_3$ is the distance from the projectile to the center of the earth: $\mathcal{R}_3 = R_1 + r(t)$.

Gravitational force for the moon. Similarly, this force has magnitude

$$F_2 = \frac{Gm_2m}{\mathcal{R}_4^2}$$

where m_2 is the mass of the moon and $\mathcal{R}_4$ is the distance from the projectile to the center of the moon: $\mathcal{R}_4 = R_2 - R_1 - r(t)$.

Competition between forces. The force equation is

$$mr''(t) = -F_1 + F_2$$

due to the directions of the force vectors. Simplifying the relations and cancelling m gives equation (1).

Technical details for (3): To justify the value for v_0, multiply equation (1) by r' and integrate the new equation from $t = 0$ to $t = t_0$ to get

$$\begin{aligned} &\frac{1}{2}\,(r'(t_0))^2 = F(r(t_0)) - F(0) + \frac{1}{2}\,v_0^2, \quad \text{where} \\ &F(r) = \frac{Gm_1}{R_1 + r} + \frac{Gm_2}{R_2 - R_1 - r}. \end{aligned} \tag{4}$$

The expression $F(r)$ is minimized when $F'(r) = 0$ or else at $r = 0$ or $r = R$. The right side of (1) is $F'(r)$, hence $F(r)$ has unique critical point $r = r^*$. Compute $F(0) = 62522859.35$, $F(r^*) = 1281502.032$ and $F(R) = 3865408.696$. Then the minimum of $F(r)$ is at $r = r^*$ and $F(r^*) \le F(r(t_0))$.

The left side of (4) is nonnegative, therefore also the right side is nonnegative, giving $\frac{1}{2}\,v_0^2 \ge F(0) - F(r(t_0))$. If the projectile ever reaches altitude r^*, then $r(t_0) = r^*$ is allowed and $v_0 \ge \sqrt{2F(0) - 2F(r^*)} \approx 11067.19091$. Restated, $v_0 < 11067.19091$ implies the projectile *never reaches altitude* r^*, hence it falls back to earth. On the other hand, if $v_0 > 11067.19092$, then by (4) and $F(r^*) \le F(r)$ it follows that $r'(t) > 0$ and therefore the projectile cannot return to earth. That is, $r(t) = 0$ for some $t > 0$ can't happen.

In summary, the least launch velocity v_0^* which allows $r(t) = r^*$ for some $t > 0$ is given by the formulas

$$v_0^* = \sqrt{2F(0) - 2F(r^*)}, \quad F(r) = \frac{Gm_1}{R_1 + r} + \frac{Gm_2}{R_2 - R_1 - r}.$$

This completes the proof of equation (3).

Exercises 4.5

Critical Altitude r^*. The symbol r^* is the altitude $r(t)$ at which gravitational effects of the moon take over, causing the projectile to fall to the moon.

1. Justify from the differential equation that $r''(t) = 0$ at $r^* = r(t)$ implies the first relation in (2):

$$\frac{Gm_2}{(R_2 - R_1 - r^*)^2} - \frac{Gm_1}{(R_1 + r^*)^2} = 0.$$

2. Solve symbolically the relation of the previous exercise for r^*, to obtain the second equation of (2):

$$r^* = \frac{R_2}{1 + \sqrt{m_2/m_1}} - R_1.$$

3. Use the previous exercise and values for the constants R_1, R_2, m_1, m_2 to obtain the approximation

$$r^* = 339,260,779 \text{ meters}.$$

4. Determine the effect on r^* for a one percent error in measurement m_2. Replace m_2 by $0.99m_2$ and $1.01m_2$ in the formula for r^* and report the two estimated critical altitudes.

Escape Velocity v_0^*. The symbol v_0^* is the velocity $r'(0)$ such that $\lim_{t\to\infty} r(t) = \infty$, but smaller launch velocities will cause the projectile to fall back to the earth. Throughout, define

$$F(r) = \frac{Gm_1}{R_1 + r} + \frac{Gm_2}{R_2 - R_1 - r}.$$

5. Let $v_0 = r'(0)$, $r^* = r(t_0)$. Derive the formula

$$\frac{1}{2}\,(r'(t_0))^2 = F(r^*) - F(0) + \frac{1}{2}\,v_0^2$$

which appears in the proof details.

6. Verify using the previous exercise that $r'(t_0) = 0$ implies

$$v_0^* = \sqrt{2(F(0) - F(r^*))}.$$

7. Verify by hand calculation that $v_0^* \approx 11067.19091$ meters per second.

8. Argue by mathematical proof that $F(r)$ is not minimized at the endpoints of the interval $0 \le r \le R$.

Numerical Experiments. Assume values given in the text for physical constants. Perform the given experiment, using numerical software, on initial value problem (1), page 258. The cases when $v_0 > v_0^*$ escape the earth, while the others fall back to earth.

9. RK4 solver, $v_0 = 11068$, $T = 515000$. Plot the solution on $0 \le t \le T$.

10. Stiff solver, $v_0 = 11068$, $T = 515000$. Plot the solution on $0 \le t \le T$.

11. RK4 solver, $v_0 = 11067.2$, $T = 800000$. Plot the solution on $0 \le t \le T$.

12. Stiff solver, $v_0 = 11067.2$, $T = 800000$. Plot the solution on $0 \le t \le T$.

13. RK4 solver, $v_0 = 11067$, $T = 1000000$. Plot the solution on $0 \le t \le T$.

14. Stiff solver, $v_0 = 11067$, $T = 1000000$. Plot the solution on $0 \le t \le T$.

15. RK4 solver, $v_0 = 11066$, $T = 800000$. Plot the solution on $0 \le t \le T$.

16. Stiff solver, $v_0 = 11066$, $T = 800000$. Plot the solution on $0 \le t \le T$.

17. RK4 solver, $v_0 = 11065$. Find a suitable value T which shows that the projectile falls back to earth, then plot the solution on $0 \le t \le T$.

18. Stiff solver, $v_0 = 11065$. Find a suitable value T which shows that the projectile falls back to earth, then plot the solution on $0 \le t \le T$.

19. RK4 solver, $v_0 = 11070$. Find a suitable value T which shows that the projectile falls to the moon, then plot the solution on $0 \le t \le T$.

20. Stiff solver, $v_0 = 11070$. Find a suitable value T which shows that the projectile falls to the moon, then plot the solution on $0 \le t \le T$.

4.6 Skydiving

A skydiver of 160 pounds jumps from a hovercraft at $15,000$ feet. The fall is mostly vertical from zero initial velocity, but there are significant effects from air resistance until the parachute opens at $5,000$ feet. The resistance effects are determined by the skydiver's clothing and body shape.

Velocity Model. Assume the skydiver's air resistance is modeled in terms of velocity v by a force equation

$$F(v) = av + bv^2 + cv^3.$$

The constants a, b, c are given by the formulas

$$a = 0.009, \quad b = 0.0008, \quad c = 0.0001.$$

In particular, the force $F(v)$ is positive for v positive. According to Newton's second law, the velocity $v(t)$ of the skydiver satisfies $mv'(t) = mg - F(v)$. We assume $mg = 160$ pounds and $g \approx 32$ feet per second per second. The **velocity model** is

$$v'(t) = 32 - \frac{32}{160}\left(0.009v(t) + 0.0008v^2(t) + 0.0001v^3(t)\right), \quad v(0) = 0.$$

Distance Model. The distance $x(t)$ traveled by the skydiver, measured from the hovercraft, is given by the **distance model**

$$x'(t) = v(t), \quad x(0) = 0.$$

The velocity is expected to be positive throughout the flight. Because the parachute opens at 5000 feet, at which time the velocity model must be replaced the open parachute model (not discussed here), the distance $x(t)$ increases with time from 0 feet to its limiting value of 10000 feet. Values of $x(t)$ from 10000 to 15000 feet make sense only for the open parachute model.

Terminal Velocity. The **terminal velocity** is an equilibrium solution $v(t) = v_\infty$ of the velocity model, therefore constant v_∞ satisfies

$$32 - \frac{32}{160}\left(0.009v_\infty + 0.0008v_\infty^2 + 0.0001v_\infty^3\right) = 0.$$

A numerical solver is applied to find the value $v_\infty = 114.1$ feet per second, which is about 77.8 miles per hour. For the solver, we define $f(v) = 32 - F(v)$ and solve $f(v) = 0$ for v. Some `maple` details:

```
f:=v->32 - (32/160)*(0.009*v+0.0008*v^2+0.0001*v^3);
fsolve(f(v)=0,v);             # 114.1032777 ft/sec
60*60*fsolve(f(v)=0,v)/5280;  # 77.79768934 mi/hr
```

A Numerical Experiment. The Runge-Kutta method will be applied to produce a table which contains the elapsed time t, the skydiver velocity $v(t)$ and the distance traveled $x(t)$, up until the distance reaches nearly 10000 feet, whereupon the parachute opens.

The objective here is to illustrate practical methods of table production in a computer algebra system or numerical laboratory. It is efficient in these computational systems to phrase the problem as a system of two differential equations with two initial conditions.

System Conversion. The velocity substitution $v(t) = x'(t)$ used in the velocity model gives us two differential equations in the unknowns $x(t)$, $v(t)$:

$$x'(t) = v(t),\ v'(t) = g - \frac{1}{m}F(v(t)).$$

Define $f(v) = g - (1/m)F(v)$. The path we follow is to execute the `maple` code below, which produces the table that follows using the default Runge-Kutta-Fehlberg algorithm.

```
eq:=32 - (32/160)*(0.009*v+0.0008*v^2+0.0001*v^3:
f:=unapply(eq,v);
de1:=diff(x(t),t)=v(t); de2:=diff(v(t),t)=f(v(t));
ic:=x(0)=0,v(0)=0;opts:=numeric,output=listprocedure:
p:=dsolve({de1,de2,ic},[x(t),v(t)],opts);
X:=eval(x(t),p); V:=eval(v(t),p);
fmt:="%10.2f  %10.2f  %10.2f\n";
seq(printf(fmt,5*t,X(5*t),V(5*t)),t=0..18);
```

t	$x(t)$	$v(t)$	t	$x(t)$	$v(t)$
5.00	331.26	106.84	50.00	5456.76	114.10
10.00	892.79	113.97	55.00	6027.28	114.10
15.00	1463.15	114.10	60.00	6597.80	114.10
20.00	2033.67	114.10	65.00	7168.31	114.10
25.00	2604.18	114.10	70.00	7738.83	114.10
30.00	3174.70	114.10	75.00	8309.35	114.10
35.00	3745.21	114.10	80.00	8879.86	114.10
40.00	4315.73	114.10	85.00	9450.38	114.10
45.00	4886.25	114.10	90.00	10020.90	114.10

The table says that the flight time to parachute open at 10,000 feet is about 90 seconds and the terminal velocity 114.10 feet/sec is reached in about 15 seconds.

More accurate values for the flight time 89.82 to 10,000 feet and time 14.47 to terminal velocity can be determined as follows.

```
fsolve(X(t)=10000,t,80..95);
fsolve(V(t)=114.10,t,2..20);
```

Alternate Method. Another way produce the table is to solve the velocity model numerically, then determine $x(t) = \int_0^t v(r)dr$ by numerical integration. Due to accuracy considerations, a variant of Simpson's rule is used, called the **Newton-cotes rule**. The `maple` implementation of this idea follows.

The first method of conversion into two differential equations is preferred, even though the alternate method reproduces the table using only the textbook material presented in this chapter.

```
f:=unapply(32-(32/160)*(0.009*v+0.0008*v^2+0.0001*v^3),v);
de:=diff(v(t),t)=f(v(t)); ic:=v(0)=0;
q:=dsolve({de,ic},v(t),numeric);
V:=t->rhs(q(t)[2]);
X:=u->evalf(Int(V,0..u,continuous,_NCrule));
fmt:="%10.2f  %10.2f  %10.2f\n";
seq(printf(fmt,5*t,X(5*t),V(5*t)),t=0..18);
```

Ejected Baggage. Much of what has been done here applies as well to an ejected parcel, instead of a skydiver. What changes is the force equation $F(v)$, which depends upon the parcel exterior and shape. The distance model remains the same, but the restraint $0 \le x \le 10000$ no longer applies, since no parachute opens. We expect the parcel to reach terminal velocity in 5 to 10 seconds and hit the ground at that speed.

Variable Mass. The mass of a skydiver can be time-varying. For instance, the skydiver lets water leak from a reservoir. This kind of problem assumes mass $m(t)$, position $x(t)$ and velocity $v(t)$ for the diver. Then Newton's second law gives a position-velocity model

$$\begin{aligned} &x'(t) = v(t), \\ &(m(t)v(t))' = G(t, x(t), v(t)). \end{aligned}$$

The problem is similar to rocket propulsion, in which expended fuel decreases the in-flight mass of the rocket. Simplifying assumptions make it possible to present formulas for $m(t)$ and $G(t, x, v)$, which can be used by the differential equation solver.

Exercises 4.6

Terminal Velocity. Assume force $F(v) = av+bv^2+cv^3$ and $g = 32$, $m = 160/g$. Using computer assist, find the terminal velocity v_∞ from the velocity model $v' = g - \frac{1}{m}F(v)$, $v(0) = 0$.

1. $a = 0$, $b = 0$ and $c = 0.0002$.

2. $a = 0$, $b = 0$ and $c = 0.00015$.

3. $a = 0$, $b = 0.0007$ and $c = 0.00009$.

4. $a = 0$, $b = 0.0007$ and $c = 0.000095$.

5. $a = 0.009$, $b = 0.0008$ and $c = 0.00015$.

6. $a = 0.009$, $b = 0.00075$ and $c = 0.00015$.

7. $a = 0.009$, $b = 0.0007$ and $c = 0.00009$.

8. $a = 0.009$, $b = 0.00077$ and $c = 0.00009$.

9. $a = 0.009$, $b = 0.0007$ and $c = 0$.

10. $a = 0.009$, $b = 0.00077$ and $c = 0$.

Numerical Experiment. Assume the skydiver problem (**??**) with $g = 32$ and constants m, a, b, c supplied below. Using computer assist, apply a numerical method to produce a table for the elapsed time t, the velocity $v(t)$ and the distance $x(t)$. The table must end at $x(t) \approx 10000$ feet, which determines the flight time.

11. $m = 160/g$, $a = 0$, $b = 0$ and $c = 0.0002$.

12. $m = 160/g$, $a = 0$, $b = 0$ and $c = 0.00015$.

13. $m = 130/g$, $a = 0$, $b = 0.0007$ and $c = 0.00009$.

14. $m = 130/g$, $a = 0$, $b = 0.0007$ and $c = 0.000095$.

15. $m = 180/g$, $a = 0.009$, $b = 0.0008$ and $c = 0.00015$.

16. $m = 180/g$, $a = 0.009$, $b = 0.00075$ and $c = 0.00015$.

17. $m = 170/g$, $a = 0.009$, $b = 0.0007$ and $c = 0.00009$.

18. $m = 170/g$, $a = 0.009$, $b = 0.00077$ and $c = 0.00009$.

19. $m = 200/g$, $a = 0.009$, $b = 0.0007$ and $c = 0$.

20. $m = 200/g$, $a = 0.009$, $b = 0.00077$ and $c = 0$.

Flight Time. Assume the skydiver problem (**??**) with $g = 32$ and constants m, a, b, c supplied below. Using computer assist, apply a numerical method to find accurate values for the flight time to 10,000 feet and the time required to reach terminal velocity.

21. $mg = 160$, $a = 0.0095$, $b = 0.0007$ and $c = 0.000092$.

22. $mg = 160$, $a = 0.0097$, $b = 0.00075$ and $c = 0.000095$.

23. $mg = 240$, $a = 0.0092$, $b = 0.0007$ and $c = 0$.

24. $mg = 240$, $a = 0.0095$, $b = 0.00075$ and $c = 0$.

Ejected Baggage. Baggage of 45 pounds is dropped from a hovercraft at $15,000$ feet. Assume air resistance force $F(v) = av + bv^2 + cv^3$, $g = 32$ and $mg = 45$. Using computer assist, find accurate values for the flight time to the ground and the terminal velocity. Estimate the time required to reach 99.95% of terminal velocity.

25. $a = 0.0095$, $b = 0.0007$, $c = 0.00009$

26. $a = 0.0097$, $b = 0.00075$, $c = 0.00009$

27. $a = 0.0099$, $b = 0.0007$, $c = 0.00009$

28. $a = 0.0099$, $b = 0.00075$, $c = 0.00009$

4.7 Lunar Lander

A lunar lander goes through free fall to the surface of the moon, its descent controlled by retrorockets that provide a constant deceleration to counter the effect of the moon's gravitational field.

The retrorocket control is supposed to produce a **soft touchdown**, which means that the velocity $v(t)$ of the lander is zero when the lander touches the moon's surface. To be determined:

$H =$ height above the moon's surface for retrorocket activation,

$T =$ flight time from retrorocket activation to soft touchdown.

Investigated here are two models for the lunar lander problem. In both cases, it is assumed that the lander has mass m and falls in the direction of the moon's gravity vector. The initial speed of the lander is assumed to be v_0. The retrorockets supply a constant thrust deceleration g_1. Either the fps or mks unit system will be used. Expended fuel ejected from the lander during thrust will be ignored, keeping the lander mass constantly m.

The distance $x(t)$ traveled by the lander t time units after retrorocket activation is given by

$$x(t) = \int_0^t v(r)dr, \quad 0 \le t \le T.$$

Therefore, H and T are related by the formulas

$$v(T) = 0, \quad x(T) = H.$$

Constant Gravitational Field

Let g_0 denote the constant acceleration due to the moon's gravitational field. Assume given initial velocity v_0 and the retrorocket thrust deceleration g_1. Define $A = g_1 - g_0$, the effective thrust. Set the origin of coordinates at the center of mass of the lunar lander. Let vector $\vec{\imath}$ have tail at the origin and direction towards the center of the moon. The force on the lander is $mv'(t)\vec{\imath}$ by Newton's second law. The forces $mg_0\vec{\imath}$ and $-mg_1\vec{\imath}$ add to $-mA\vec{\imath}$. Force competition $mv'(t)\vec{\imath} = -mA\vec{\imath}$ gives the velocity model

$$mv'(t) = -mA, \quad v(0) = v_0.$$

This quadrature-type equation is solved routinely to give

$$v(t) = -At + v_0, \quad x(t) = -A\frac{t^2}{2} + v_0 t.$$

The equation $v(T) = 0$ gives $T = v_0/A$ and $H = x(T) = v_0^2/(2A)$.

Numerical illustration. Let $v_0 = 1200$ miles per hour and $A = 30000$ miles per hour per hour. We compute values $T = 1/25$ hours $= 2.4$ minutes and $H = x(T) = 24$ miles. A `maple` answer check appears below.

```
v0:=1200; A:=30000;
X:=t->-A*t^2/2+v0*t;
T:=(v0/A): (T*60.0).’min’,X(T).’miles’; # 2.4 min, 24 miles
A1:=A*2.54*12*5280/100/3600/3600; # mks units 3.725333334
v1:=v0*12*2.54*5280/100/3600;     # mks units 536.448
evalf(convert(X(T),units,miles,meters)); # 38624.256
```

The constant field model predicts that the retrorockets should be turned on 24 miles above the moon's surface with soft landing descent time of 2.4 minutes. It turns out that a different model predicts that 24 miles is too high, but only by a small amount. We investigate now this alternative model, based upon replacing the constant gravitational field by a variable field.

Variable Gravitational Field

The system of units will be the *mks* system. Assume the lunar lander is located at position P above the moon's surface. Define symbols:

$m =$ mass of the lander in kilograms,

$M = 7.35 \times 10^{22}$ kilograms is the mass of the moon,

$R = 1.74 \times 10^{6}$ meters is the mean radius of the moon,

$G = 6.6726 \times 10^{-11}$ is the universal gravitation constant, in *mks* units,

$H =$ height in meters of position P above the moon's surface,

$v_0 =$ lander velocity at P in meters per second,

$g_0 = GM/R^2 =$ constant acceleration due to the moon's gravity in meters per second per second,

$g_1 =$ constant retrorocket thrust deceleration in meters per second per second,

$A = g_1 - g_0 =$ effective retrorocket thrust deceleration in meters per second per second, constant field model,

$t =$ time in seconds,

$x(t) =$ distance in meters from the lander to position P,

$v(t) = x'(t) =$ velocity of the lander in meters per second.

The project is to find the height H above the moon's surface and the descent time T for a soft landing, using fixed retrorockets at time $t = 0$. The origin of coordinates will be P and $\vec{\imath}$ is directed from the lander to the moon. Then $x(t)\vec{\imath}$ is the lander position at time t. The initial conditions are $x(0) = 0$, $v(0) = v_0$. Let $g_0(t)$ denote the variable acceleration of the lander due to the moon's gravitational field. Newton's universal gravitation law applied to point masses representing the lander and the moon gives the expression

$$\text{Force} = mg_0(t)\vec{\imath} = \frac{GmM}{(R + H - x(t))^2}\vec{\imath}.$$

The force on the lander is $mx''(t)\vec{\imath}$ by Newton's second law. The force is also $mg_0(t)\vec{\imath} - mg_1\vec{\imath}$. Force competition gives the second order distance model

$$mx''(t) = -mg_1 + \frac{mMG}{(R + H - x(t))^2}, \quad x(0) = 0, \quad x'(0) = v_0.$$

The technique from the Jules Verne problem applies: multiply the differential equation by $x'(t)$ and integrate from $t = 0$ to the soft landing time $t = T$. The result:

$$\left.\frac{(x'(t))^2}{2}\right|_{t=0}^{t=T} = -g_1(x(T) - x(0)) + \left.\frac{GM}{R + H - x(t)}\right|_{t=0}^{t=T}.$$

Using the relations $x(0) = 0$, $x'(0) = v_0$, $x'(T) = 0$ and $x(T) = H$ gives a simplified implicit equation for H:

$$-\frac{v_0^2}{2} = -g_1 H + \frac{GM}{R} - \frac{GM}{R + H}.$$

Numerical illustration. Use $v_0 = 536.448$, $g_1 = 5.3452174$ to mimic the constant field example of initial velocity 1200 miles per hour and effective retrorocket thrust 30000 miles per hour per hour. A soft landing is possible from height $H = 23.7775$ miles with a descent time of $T = 2.385$ minutes. These results compare well with the constant field model, which had results of $H = 24$ miles and $T = 2.4$ minutes. Some `maple` details follow.

```
M:=7.35* 10^(22);R:=1.74* 10^6;G:=6.6726* 10^(-11);
v0_CFM:=1200: A_CFM:=30000: # Constant field model values
cf:=1*5280*12*2.54/100/3600: # miles/hour to meters/second
v0:=v0_CFM*cf; g0:=G*M/R^2: g1:=A_CFM*cf/3600+g0;
```

```
eq:= -(v0^2/2) + g1*H + G*M/(R+H) - G*M/R=0:
HH:=[solve(eq,H)][1];  # HH := 38266 meters
de:=diff(x(t),t,t) = -g1 + M*G/(R+HH-x(t))^2;
ic:= x(0)=0, D(x)(0)=v0;
with(DEtools):
DEplot(de,x(t),t=0..290,[[ic]]); # See the plot below
p:=dsolve({de,ic},x(t),numeric):
X:=t->evalf(rhs(p(t)[2])):
V:=t-> evalf(rhs(p(t)[3])):
TT1:=fsolve('V(t)'=0,t,100..800): TT:=TT1/60:
TT1.'seconds', TT.'minutes';
X(TT1).'meters', ((X(TT1)*100/2.54)/12/5280).'miles';
```

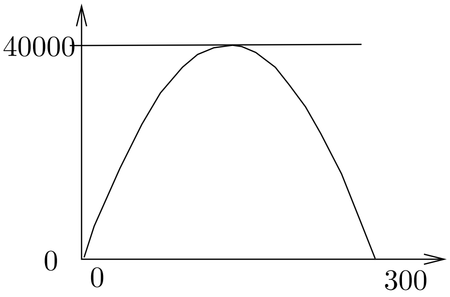


Figure 9. A `maple` plot used to determine the descent time $T = 2.385$ minutes.

Modeling

The field of the earth has been ignored in both models, which is largely justified because the universal gravitation law term for the lander and the earth is essentially zero for lander locations near the moon.

The field for the lander and the moon is not constant, and therefore it can be argued that conditions exist when assuming it is constant will produce invalid and obviously incorrect results.

Are there cases when the answers for the two models differ greatly? Yes, but the height H of retrorocket activation has to be large. This question is re-visited in the exercises.

Control problems. The descent problem for a lunar lander is a control problem in which the **controller** is the retrorocket plus the duration of time in which it is active. All we have done here is to decide that the descent should be controlled by retrorockets well in advance of 24 miles above the moon's surface. The methods used here can be applied to gain insight into the **bang-bang control problem** of turning on the retrorockets for n intervals of time of durations $\Delta t_1, \ldots, \Delta t_n$ to make an *almost* soft landing.

Primitive numerical methods. The predictions made here using the computer algebra system `maple` can be replaced by primitive RK4 methods and graphing. No practising scientist or engineer would do *only* that, however, because they want to be confident of the calculations and the results. The best idea is to use a **black box** of numerical and graphical methods which have little chance of failure, e.g., a computer algebra system or a numerical laboratory.

Exercises 4.7

Lunar Lander Constant Field. Find the retrorocket activation time T and the activation height $x(T)$. Assume the constant gravitational field model. Units are miles/hour and miles/hour per hour.

1. $v_0 = 1210$, $A = 30020$.

2. $v_0 = 1200$, $A = 30100$.

3. $v_0 = 1300$, $A = 32000$.

4. $v_0 = 1350$, $A = 32000$.

5. $v_0 = 1500$, $A = 45000$.

6. $v_0 = 1550$, $A = 45000$.

7. $v_0 = 1600$, $A = 53000$.

8. $v_0 = 1650$, $A = 53000$.

9. $v_0 = 1400$, $A = 40000$.

10. $v_0 = 1450$, $A = 40000$.

Lunar Lander Variable Field. Find the retrorocket activation time T and the activation height $x(T)$. Assume the variable gravitational field model and *mks* units.

11. $v_0 = 540.92$, $g_1 = 5.277$.

12. $v_0 = 536.45$, $g_1 = 5.288$.

13. $v_0 = 581.15$, $g_1 = 5.517$.

14. $v_0 = 603.504$, $g_1 = 5.5115$.

15. $v_0 = 625.86$, $g_1 = 5.59$.

16. $v_0 = 603.504$, $g_1 = 5.59$.

17. $v_0 = 581.15$, $g_1 = 5.59$.

18. $v_0 = 670.56$, $g_1 = 6.59$.

19. $v_0 = 670.56$, $g_1 = 6.83$.

20. $v_0 = 715.26$, $g_1 = 7.83$.

Distinguishing Models. The constant field model (**1**) (page 268) and the variable field model (**2**) (page 269) are verified here, by example, to be distinct. Find the retrorocket activation times T_1, T_2 and the activation heights $x_1(T_1)$, $x_2(T_2)$ for the two models (**1**), (**2**). Relations $A = g_1 - g_0$ and $g_0 = GM/R^2$ apply to compute g_1 for the variable field model.

21. $v_0 = 1200$ mph, $A = 10000$ mph/h. Answer: 72, 66.91 miles.

22. $v_0 = 1200$ mph, $A = 12000$ mph/h. Answer: 60, 56.9 miles.

23. $v_0 = 1300$ mph, $A = 10000$ mph/h. Answer: 84.5, 77.7 miles.

24. $v_0 = 1300$ mph, $A = 12000$ mph/h. Answer: 70.42, 66.26 miles.

4.8 Comets

Planet Mercury. Its elliptical orbit has major semi-axis $a = 0.3871$ AU (astronomical units) and eccentricity $e = 0.2056$. The ellipse can be described by the equations

$$\begin{aligned} x(t) &= a\cos(E(t)), \\ y(t) &= a\sqrt{1-e^2}\sin(E(t)), \end{aligned}$$

where t is the mean anomaly ($0 \le t \le 2\pi$) and $E(t)$ is the eccentric anomaly determined from Kepler's equation $E = t + e\sin(E)$.

The path of mercury is an ellipse, yes. Like the earth, the path is essentially circular, due to eccentricity near zero.

Halley's Comet. The Kepler theory for mercury applies to Halley's comet, which has a highly elliptical orbit of eccentricity $e = 0.967$. The major semi-axis is $a = 17.8$ astronomical units (AU), the minor semi-axis is $b = a\sqrt{1-e^2} = 4.535019431$ AU, with period about 76 earth-years.

Our project is to determine $E(t)$ numerically for Halley's comet and plot an animation of the elliptical path of the comet.

History

Kepler's laws of planetary motion were published in 1609 and 1618. The laws are named after Johannes Kepler (1571-1630), a German mathematician and astronomer, who formulated the laws after years of calculation based upon excellent observational data of the Danish astronomer Tycho Brahe (1546-1601). The three laws:

I. The orbit of each planet is an ellipse with the sun at one focus.

II. The line joining the sun to a planet sweeps out equal areas in equal time.

III. The square of the planet's period of revolution is proportional to the cube of the major semi-axis of its elliptical orbit.

These laws apply not only to planets, but to satellites and comets. A proof of Kepler's first two laws, assuming Newton's laws and a vector analysis background, can be found in this text, page 532, *infra.*

The elliptical orbit can be written as

$$\begin{aligned} x(M) &= a\cos(E(M)), \\ y(M) &= b\sin(E(M)), \end{aligned}$$

where a and b are the semi-axis lengths of the ellipse. Astronomers call function E the planet's **eccentric anomaly** and M the planet's **mean anomaly**.

The minor semi-axis of the ellipse is given by

$$b = a\sqrt{1-e^2},$$

where e is the **eccentricity** of the elliptical orbit. The mean anomaly satisfies $M = 2\pi t/T$, where t=time and T is the period of the planet.

It is known that the first two laws of Kepler imply **Kepler's equation**

$$E = M + e\sin(E).$$

Kepler's Initial Value Problem

The equation $E = M + e\sin E$, called Kepler's equation, is the unique implicit solution of the separable differential equation

$$(1)\qquad \begin{cases} \dfrac{dE}{dM} &= \dfrac{1}{1-e\cos(E)}, \\ E(0) &= 0. \end{cases}$$

The initial value problem (1) *defines* the eccentric anomaly $E(M)$. We are able to compute values of E by suitable first order numerical methods, especially RK4.

The reader should pause and compute dE/dM by implicit differentiation of Kepler's equation. The idea works on many implicit equations: find an initial value problem by implicit differentiation, which replaces the implicit equation.

Eccentric Anomaly and Elliptical Orbit

The solution for comet Halley uses `maple` in a direct manner, basing the solution on Kepler's equation. Details:

```
# Kepler's equation E = M + e sin(E)
 e:=0.967:EE := unapply(RootOf(_Z-M-e*sin(_Z)),M);
 Ex:=cos(EE(M)):Ey:=sqrt(1-e^2)*sin(EE(M)):
 plot(EE(M),M=0..2*Pi);
 plot([Ex,Ey,M=0..2*Pi]);
```

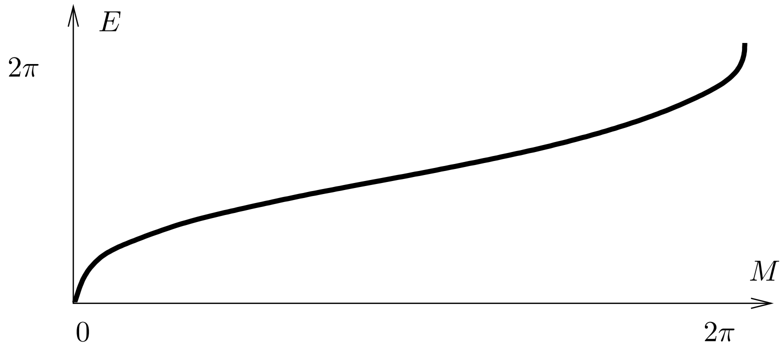


Figure 10. Eccentric anomaly plot for Halley's comet.

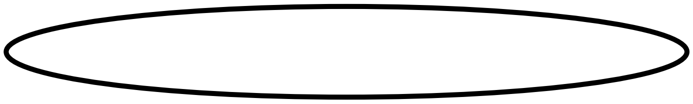


Figure 11. Elliptic trace plot of Halley's comet.

Comet Halley's Positions each Year

The elliptic trace plot can be modified to display a circle for each comet position from year 0 to year 75; see Figure 12. Implemented here is an approach to evaluation of the eccentric anomaly $E(M)$ by numerical differential equation methods. This method is orders of magnitude faster than the `RootOf` method of the previous illustration.

The lack of circles near the focus on the right is explained by the increased speed of the comet near the sun, which is at this focus.

```
# Comet positions each year
 e:=0.967:de:=diff(y(x),x)=1/(1-e*cos(y(x))); ic:=y(0)=0;
 desolved:=dsolve({de,ic},numeric,output=listprocedure);
 EE := eval(y(x),desolved):
 Ex:=unapply(cos(EE(M)),M):
 Ey:=unapply(sqrt(1-e^2)*sin(EE(M)),M):
 snapshots:=seq([Ex(2*n*Pi/76),Ey(2*n*Pi/76)],n=0..76):
 opts:=scaling=constrained,axes=boxed,style=point,
       symbolsize=12,symbol=circle,thickness=2:
 plot([snapshots],opts);
```

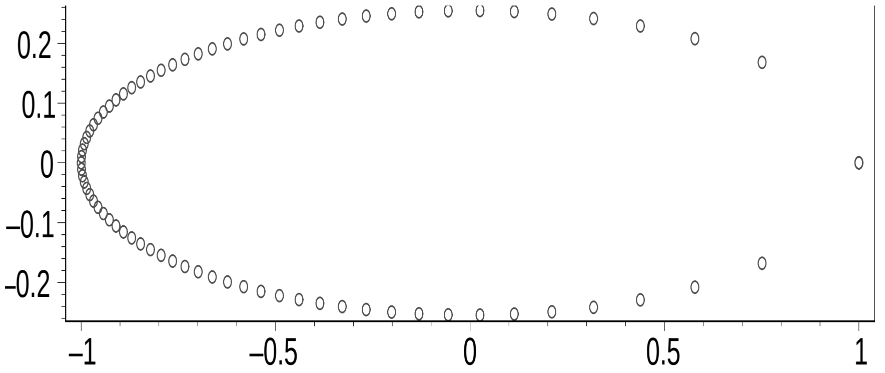


Figure 12. Halley's comet positions each earth-year. On the axes, one unit equals 17.8 **AU.**

Halley's Comet Animation

The computer algebra system `maple` will be used to produce a simple animation of Halley's comet as it traverses its 76-year orbit around the sun. The idea is to solve Kepler's initial value problem in order to find the value of the eccentric anomaly $E(M)$, then divide the orbit into 76 frames and display each in succession to obtain the animation. The obvious method of defining E by Kepler's equation $E = M + e\sin E$ is too slow for most machines, hence the differential equations method is used.

While each comet position in Figure 13 represents an equal block of time, about one earth-year, the amount of path traveled varies. This is because the speed along the path is not constant, the comet traveling fastest near the sun. The most detail is shown for an animation at 2 frames per second. The orbit graph uses one unit equal to about 17.8 astronomical units, to simplify the display.

```
# Simple Halley's comet animation
 e:=0.967:de:=diff(y(x),x)=1/(1-e*cos(y(x))); ic:=y(0)=0;
 desolved:=dsolve({de,ic},numeric,output=listprocedure);
 EE := eval(y(x),desolved):
 xt:=cos(EE(M)):yt:=sqrt(1-e^2)*sin(EE(M)):
 opts:=view=[-1..1,-0.28..0.28],frames=76,
       scaling=constrained,axes=boxed,style=point,
       symbolsize=12,symbol=circle,thickness=2:
 plots[animatecurve]([xt,yt,M=0..2*Pi],opts);
```

Improved Animation. To display the ellipse constantly and animate the comet along the ellipse requires more plot steps. The method is illustrated in this block of `maple` code. The comet position for $t = 2.4516$ earth-years ($M \approx 2\pi t/76$) is shown in Figure 14.

```
# Improved animation of Halley's comet
 e:=0.967:de:=diff(y(x),x)=1/(1-e*cos(y(x))); ic:=y(0)=0;
 desolved:=dsolve({de,ic},numeric,output=listprocedure);
 EE := eval(y(x),desolved):
 comet:=unapply([cos(EE(M)),sqrt(1-e^2)*sin(EE(M))],M):
 options1:=view=[-1..1,-0.28..0.28]:
 options2:=scaling=constrained,axes=none,thickness=2:
 options3:=style=point,symbolsize=24,symbol=circle:
 opts1:=options1,options2,color=blue:
 opts:=options1,options2,options3:
 COMET:=[[comet(2*Pi*t/(76))],opts]:
 ellipse:=plot([cos(x),sqrt(1-e^2)*sin(x),x=0..2*Pi],opts1):
 with(plots):
 F:=animate( plot,COMET,t=0..4,frames=32,background=ellipse):
 G:=animate( plot,COMET,t=5..75,frames=71,background=ellipse):
 H:=animate( plot,COMET,t=75..76,frames=16,background=ellipse):
 display([F,G,H],insequence=true);
```

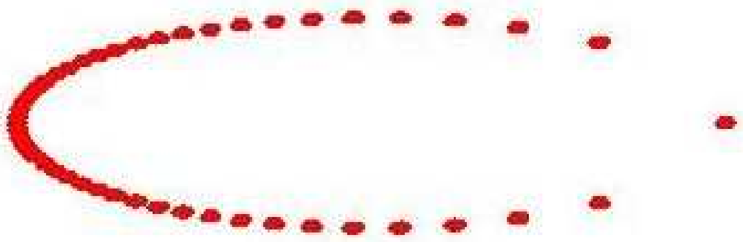


Figure 13. A simple Halley's comet animation.

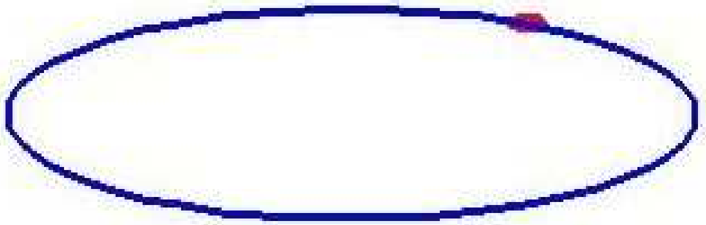


Figure 14. Improved Halley's comet animation. The frame shown is for mean anomaly $M = 2.4516$.

Exercises 4.8

Eccentric Anomaly for the Planets. Make a plot of the eccentric anomaly $E(M)$ on $0 \le M \le 2\pi$.

1. Mercury, $e = 0.2056$

2. Venus, $e = 0.0068$

3. Earth, $e = 0.0167$

4. Mars, $e = 0.0934$

5. Jupiter, $e = 0.0483$

6. Saturn, $e = 0.0560$

7. Uranus, $e = 0.0461$

8. Neptune, $e = 0.0097$

Elliptic Path of the Planets. Make a plot of the elliptic path of each planet, using constrained scaling with the given major semi-axis A (in astronomical units AU).

9. Mercury, $e = 0.2056$, $A = 0.39$

10. Venus, $e = 0.0068$, $A = 0.72$

11. Earth, $e = 0.0167$, $A = 1$

12. Mars, $e = 0.0934$, $A = 1.52$

13. Jupiter, $e = 0.0483$, $A = 5.20$

14. Saturn, $e = 0.0560$, $A = 9.54$

15. Uranus, $e = 0.0461$, $A = 19.18$

16. Neptune $e = 0.0097$, $A = 30.06$

Planet Positions. Make a plot with at least 8 planet positions displayed. Use constrained scaling with major semi-axis 1 in the plot. Display the given major semi-axis A and period T in the legend.

17. Mercury, $e = 0.2056$, $A = 0.39$ AU, $T = 0.24$ earth-years

18. Venus, $e = 0.0068$, $A = 0.72$ AU, $T = 0.62$ earth-years

19. Earth, $e = 0.0167$, $A = 1$ AU, $T = 1$ earth-years

20. Mars, $e = 0.0934$, $A = 1.52$ AU, $T = 1.88$ earth-years

21. Jupiter, $e = 0.0483$, $A = 5.20$ AU, $T = 11.86$ earth-years

22. Saturn, $e = 0.0560$, $A = 9.54$ AU, $T = 29.46$ earth-years

23. Uranus, $e = 0.0461$, $A = 19.18$ AU, $T = 84.01$ earth-years

24. Neptune $e = 0.0097$, $A = 30.06$ AU, $T = 164.8$ earth-years

Comet Positions. Make a plot with at least 8 comet positions displayed. Use constrained scaling with major-semiaxis 1 in the plot. Display the given eccentricity e and period T in the legend.

25. Churyumov-Gerasimenko orbits the sun every 6.57 earth-years. Discovered in 1969. Eccentricity $e = 0.632$.

26. Comet Wirtanen was the original target of the Rosetta space mission. This comet was discovered in 1948. The comet orbits the sun once every 5.46 earth-years. Eccentricity $e = 0.652$.

27. Comet Wild 2 was discovered in 1978. The comet orbits the sun once every 6.39 earth-years. Eccentricity $e = 0.540$.

28. Comet Biela was discovered in 1772. It orbits the sun every 6.62 earth-years. Eccentricity $e = 0.756$.

29. Comet Encke was discovered in 1786. It orbits the sun each 3.31 earth-years. Eccentricity $e = 0.846$.

30. Comet Giacobini-Zinner, discovered in 1900, orbits the sun each 6.59 earth-years. Eccentricity $e = 0.708$.

31. Comet Schwassmann-Wachmann, discovered in 1930, orbits the sun every 5.36 earth-years. Eccentricity $e = 0.694$.

32. Comet Swift-Tuttle was discovered in 1862. It orbits the sun each 120 earth-years. Eccentricity $e = 0.960$.

Comet Animations. Make an animation plot of comet positions. Use constrained scaling with major-semiaxis 1 in the plot. Display the given period T and eccentricity e in the legend.

33. Comet Churyumov-Gerasimenko
$T = 6.57$, $e = 0.632$.

34. Comet Wirtanen
$T = 5.46$, $e = 0.652$.

35. Comet Wild 2
$T = 6.39$, $e = 0.540$.

36. Comet Biela
$T = 6.62$, $e = 0.756$.

37. Comet Encke
$T = 3.31$, $e = 0.846$.

38. Comet Giacobini-Zinner
$T = 6.59$, $e = 0.708$.

39. Comet Schwassmann-Wachmann
$T = 5.36$, $e = 0.694$.

40. Comet Swift-Tuttle
$T = 120$, $e = 0.960$.

4.9 Fish Farming

Discussed are logistic models for population dynamics in fish farms. The models are suitable for Pangasius and Tilapia populations. The focus will be on species *tilapia*.

Pangasius. In America, both USA-produced and imported fresh-water catfish can be sold with the labels **Swai**, **Basa** or the subgenus label **Pangasius**, which is the predominant generic label in Europe, with more than 20 varieties. Basa and Swai are different catfish, with different texture and flavor. USA production of farmed catfish increased after 2002, when Vietnam Basa imports were stopped by labeling laws and tariffs. USA channel catfish (four barbels) are harvested after 18 months, at 10 pounds weight. Pangasius varieties are harvested after 4–6 months, at about 2 pounds or less, to produce fillets of 3–12 ounces.

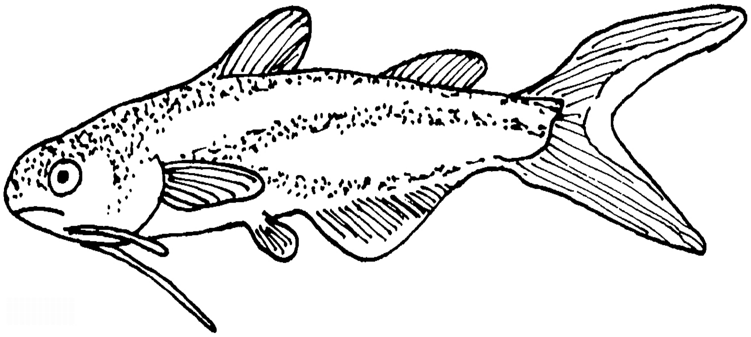


Figure 15. Pangasius, a fresh water catfish with two barbels.

Tilapia. This fresh-water fish originated in Africa 2500 years ago. The popular varieties sold in the USA are marketed under the label **Tilapia** (both dark and light flesh). They are produced in the USA at fish farms in Arizona, California and Florida. Imported Tilapia at 600-900 grams market weight (30% fillets) make up the bulk of USA-consumed Tilapia.

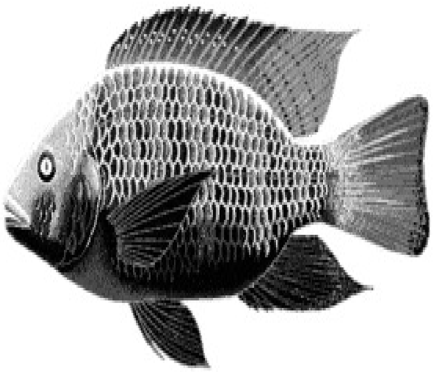


Figure 16. Tilapia.
A fresh water fish from the river Nile. Tilapia are farmed around the world in temperate climates.

Population Dynamics of Fisheries

Fisheries can be wild or farmed. One example is a fish hatchery using concrete tanks. Tilapia freshwater farms can use earthen ponds, canvas tanks, concrete tanks, river cages, pens and old mining quarries.

Tilapia Farming

Detailed life history data for Tilapia is as follows:

- Age at sexual maturity: 5–6 months
- Size at sexual maturity: 28–350 grams
- Stocking ratio for spawning: 7–10 broods/year using 2–5 females per male
- Spawning success: 20–30% spawns per week
- Eggs per female fish: 1–4 eggs per gram of fish
- Survival of egg to fry: 70–90% (fry less than 5 grams)
- Survival of fry to fingerling: 60–90% (fingerling 5–30 grams)
- Survival of fingerling to market: 70–98% (market is 30 to 680 grams)

Tilapia fry might be produced from an initial stock of 1000 female ND-2 and 250 male ND-1. Hatched ND-21 fry will be all male, which have higher market weight. Egg production per female averages from 300 to 500 fry per month, with about 10% lost before reaching 5 gram weight. The marketed Tilapia are about 900 grams in Central America plants (Belize, El Salvador). In Arizona, California and Florida plants, Tilapia market weights vary from 600 to 800 grams, or 1.5–1.75 pounds.

In commercial secondary tanks, fingerlings grow in water temperatures 76–84 degrees Fahrenheit with a death rate of about 0.05%. One fingerling grows to market size on less than 3 pounds of food.

Logistic Harvesting on a Time Interval

The Logistic equation for a **constant harvesting** rate $h \geq 0$ is

$$\frac{dx}{dt} = kx(t)(M - x(t)) - h.$$

The Logistic equation for a non-constant harvesting rate $h(t) \geq 0$ is

$$\frac{dx}{dt} = kx(t)(M - x(t)) - h(t).$$

A simplified situation is constant harvesting $h(t) = c > 0$ on a given time interval $a \le t \le b$, but zero otherwise.

In a more sophisticated setting, $h(t)$ is a positive constant c_i on given time interval $a_i \le t \le b_i$, $i = 1, \ldots, n$, but zero otherwise. Harvesting can also depend on the population size, which replaces $h(t)$ by $h(t)x(t)$ in the differential equation. Modelling need not be for an individual tank or pond, but the aggregate of all tanks, ponds and cages of an enterprise, viewed from the prospect of so many fish grown to market weight.

Logistic Periodic Harvesting

The periodic harvest Logistic equation is

$$\frac{dx}{dt} = kx(t)(M - x(t)) - h(t)$$

where $h(t) \ge 0$ is the rate of harvest, usually a positive constant c_i on a given time interval $a_i \le t \le b_i$, $i = 1, \ldots, n$, but zero otherwise. The equation $h(t + T) = h(t)$ might hold for some value of T, in which case $h(t)$ is a classical periodic function.

Tank harvests can be periodic, in order to reduce the density of fish per volume of water, or to remove fingerlings. Harvested fish can be assumed to be live, and sent either to slaughter or else to another tank, to grow bigger. This model fits Tilapia fry production in ponds, for which it is typical that ND-2 females produce more and more eggs as they mature (then $c_1 < c_2 < c_3 < \cdots$). The time intervals for Tilapia are about a month apart.

Malaysian Tilapia Example

Described here is the 2012 work of M. F. Laham, et al, [**?**], in which a logistic model is used to study harvesting strategies for tilapia fish farming. This work is elementary, in the sense that it treats an ideal example, with no intentional application to management of a Tilapia farm. It illustrates general expectations for fish production, based on gross estimates of a pond scenario.

The data was obtained from the Department of Fisheries of Malaysia and from the Malaysian fish owner of selected ponds situated at Gombak, Selangor. The fisheries department claims (2008) that a fish pond can sustain 5 tilapia fish for every square meter of surface area.[3] The selected pond has an area of 15.61 Hectors, which is equivalent to 156100 square meters, 38 acres or 25000 square feet. The pond carrying capacity is

[3] Normal stocking is 1.6 fish per square meter, from which reproduction allows fish population growth to carrying capacity (a theoretical number).

$M = 780500$ fish. Tilapia mature in 6 months and at least 80 percent will survive to maturity (Thomas and Michael 1999 [**?**]).

The Logistic Growth Model, in the absence of harvesting, can be written in the form

$$\frac{dx}{dt} = rx(t)(1 - x(t)/M), \quad r = 0.8, \quad M = 780500.$$

In terms of the alternate model $P' = kP(M - P)$, the constant k equals $rM = 624400$. The work of Laham et al focuses on harvesting strategies, considering the constant harvesting model

$$\text{(1)} \qquad \frac{dx}{dt} = rx(t)(1 - x(t)/M) - H_0$$

and the periodic harvesting model

$$\text{(2)} \quad \frac{dy}{dt} = ry(t)(1 - y(t)/M) - H(t), \quad H(t) = \begin{cases} H_0 & 0 \le t \le 6, \\ 0 & 6 < t \le 12. \end{cases}$$

The constant $H_0 = 156100$ is explained below.The discontinuous harvesting function $H(t)$ is extended to be 12-month periodic: $H(t+12) = H(t)$.

Constant Harvesting. The parameters in the model are $r = 0.8$, an estimate of the fraction of fish that will survive to market age, and the pond carrying capacity $M = 780500$. The periodic harvesting value $H_0 = 156100$ arises from the constant harvesting model, by maximizing population size at the equilibrium point for the constant harvesting model. Briefly, the value H_0 is found by requiring $\frac{dx}{dt} = 0$ in the constant harvesting model, replacing $x(t)$ by constant P. This implies

$$\text{(3)} \qquad rP\left(1 - \frac{P}{M}\right) - H_0 = 0.$$

The mysterious value H_0 is the one that makes the discriminant zero in the quadratic formula for P. Then $H_0 = \frac{rM}{4} = 156100$ and $P = 389482$. This **bifurcation point** separates the global behavior of the constant harvesting model as in Table 23. We use the notation P_1, P_2 for the two real equilibrium roots of the quadratic equation (3), assuming $H_0 < 156100$ and $P_1 < P_2$.

Harvest Constant	Initial Population	Behavior
$H_0 = 156100$	$x(0) \ge 389482$	$x(t) \to 389482$,
$H_0 = 156100$	$x(0) < 389482$	$x(t) \to 0$, extinction,
$H_0 > 156100$	any $x(0)$	$x(t) \to 0$, extinction,
$H_0 < 156100$	$x(0) < P_1$	$x(t) \to 0$, extinction,
$H_0 < 156100$	$P_1 < x(0) < P_2$	$x(t) \to P_2$, sustainable,
$H_0 < 156100$	$x(0) \ge P_2$	$x(t) \to P_2$, sustainable.

Table 23. Constant Harvesting Model

Periodic Harvesting. The model is an initial value problem (2) with initial population $y(0)$ equal to the number of Tilapia present, where $t = 0$ is an artificial time representing the current time after some months of growth. The plan is to harvest H_0 fish in the first 6 months.

Direct inspection of the two models shows that $x(t) = y(t)$ for the first six months, regardless of the choice of H_0. Because the constant harvesting model shows that harvesting rates larger than 156100 lead to extinction, then it is clear that the harvesting rate can be $H_0 = 156100$.

The harvesting constant H_0 can be larger than 156100, because the population of fish is allowed to recover for six months after the harvest. Assuming $H_0 > 156100$, then the solution $y(t)$ decreases for 6 months to value $y(6)$, which if positive, allows recovery of the population in the following 6 non-harvest months. There is a catch: the population could fail to grow to harvest size in the following 6 months, causing a reduced production in subsequent years.

To understand the problem more clearly, we present an example where $H_0 > 156100$ and the harvest is sustainable for 3 years, then another example where $H_0 > 156100$ and the harvest fails in the second year.

8 Example (Sustainable Harvest $H_0 > 156100$) Choose $H_0 = 190000$ and $y(0) = 390250 = M/2$. Computer assist gives 6-month population size decreasing to $y(6) = 16028.6$. Then for $6 < t < 12$ the population increases to $y(12) = 560497.2$, enough for a second harvest. The population continues to rise and fall, $y(18) = 320546.6$, $y(24) = 771390.7$, $y(30) = 391554.0$, $y(36) = 774167.6$, a sustainable harvest for the first three years.

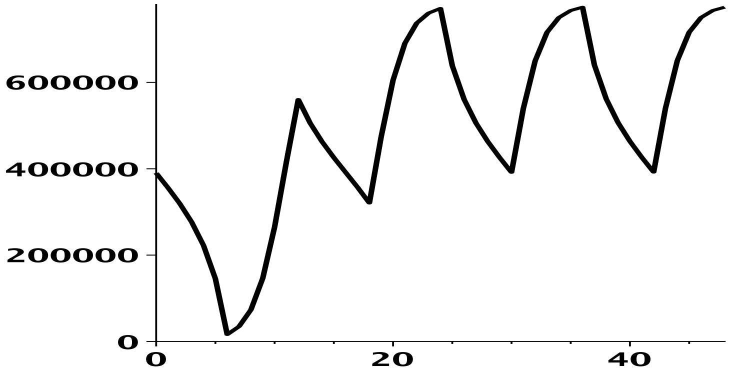


Figure 17. Sustainable harvest for 3 years, $H_0 = 190000$, $y(0) = M/2$.

9 Example (Unsustainable Harvest $H_0 > 156100$) Choose $H_0 = 190500$ and $y(0) = 390250 = M/2$. Computer assist gives 6-month population size decreasing to $y(6) = 5263.1$. Then for $6 < t < 12$ the population increases to $y(12) = 352814$, enough for a second harvest. But then the model $y(t)$ decreases to zero (extinction) at $t = 16.95$, meaning the harvest fails in the second year.

The same example with $y(0) = (M/2)(1.02) = 398055$ (2 percent larger) happens to be sustainable for three years. Sustainable harvest is sensitive to both harvesting constant and initial population.

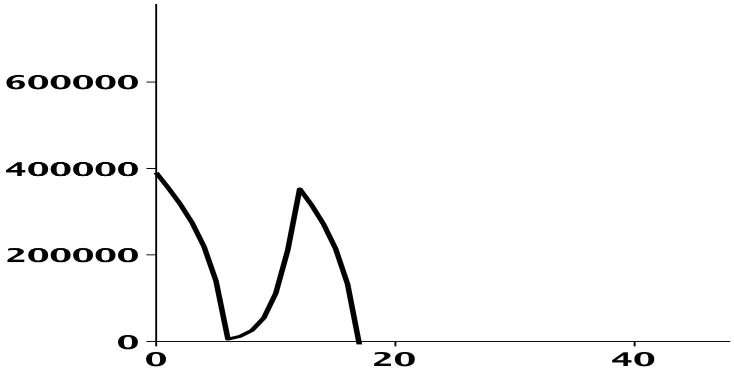


Figure 18. Unsustainable harvest, failure in year two, $H_0 = 190500$, $y(0) = M/2$.

Logistic Systems

The Lotka-Volterra equations, also known as the predator-prey equations, are a pair of first order nonlinear differential equations frequently used to describe the dynamics of biological systems in which two species interact, one a predator and one its prey (e.g., foxes and rabbits). They evolve in time according to the pair of equations:

$$\begin{aligned} \frac{dx}{dt} &= x(\alpha - \beta y), \\ \frac{dy}{dt} &= -y(\gamma - \delta x), \end{aligned}$$

where,

x is the number of prey,

y is the number of some predator,

t is time,

$\frac{dy}{dt}$ and $\frac{dx}{dt}$ are population growth rates,

Parameter α is a growth rate for the prey while parameter γ is a death rate for the predator.

Parameters β and δ describe species interaction, with $-\beta xy$ decreasing prey population and δxy increasing predator population.

A. J. Lotka (1910, 1920) used the predator-prey model to study autocatalytic chemical reactions and organic systems such as plants and grazing animals. In 1926, V. Volterra made a statistical analysis of fish catches in the Adriatic Sea, publishing at age 22 the same equations, an independent effort.

Walleye on Lake Erie

The one-dimensional theory of the logistic equation can be applied to fish populations in which there is a predator fish and a prey fish. This problem was studied by A. L. Jensen in 1988. Using the model of P. A. P. Larkin 1966, Jensen invented a mathematical model for walleye populations in the western basin of Lake Erie. The examples for **prey** are Rainbow Smelt (*Osmerus mordax*) in Lake Superior and Yellow Perch (*Perca flavescens*) from Minnesota lakes. The **predator** is Walleye (*Sander vitreus*).

Figure 19. Yellow Perch.
The prey, from Shagawa Lake in Northeast Minnesota.

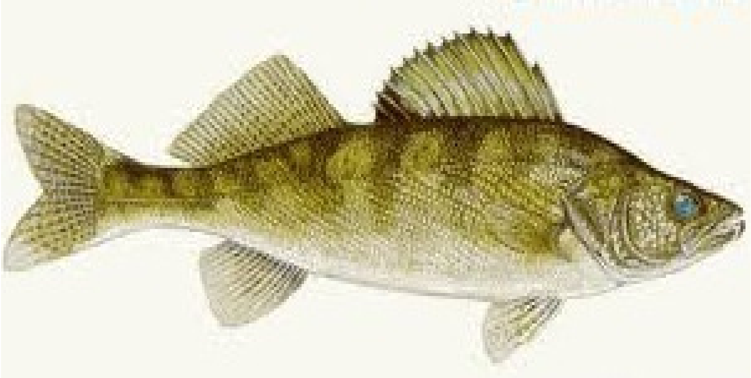


Figure 20. Walleye.
The predator, also called Yellow Pike, or Pickerel.

The basis for the simulation model is the Lotka-Volterra predator-prey model. The following assumptions were made.

- A decrease in abundance results in an increase in food concentration.
- An increase in food concentration results in an increase in growth and size.
- An increase in growth and size results in a decrease in mortality because mortality is a function of size.

The relation between prey abundance N_1 and predator abundance N_2 is given by the equations

$$\begin{aligned}\frac{dN_1}{dt} &= r_l N_1(1 - N_1/K_1) - b_1 N_1 N_2,\\ \frac{dN2}{dt} &= r_2 N_2(1 - N_2/K_2) - b_2 N_1 N_2.\end{aligned}$$

If $b_1 = b_2 = 0$, then there is no interaction of predator and prey, and the two populations N_1, N_2 grow and decay independently of one another. The carrying capacities are K_1, K_2, respectively, because each population N satisfies a logistic equation

$$\frac{dN}{dt} = rN(1 - N/K).$$

Interested readers are referred to the literature below, for further details. Solution methods for systems like (20) are largely numeric. Qualitative methods involving equilibrium points and phase diagrams have an important role in the analysis.

Jensen, A. L.: *Simulation of the potential for life history components to regulate Walleye population size*, Ecological Modelling 45(1), pp 27-41, 1989.

Larkin, P.A.P., 1966. *Exploitation in a type of predator-prey relationship. J. Fish. Res. Board Can.*, 23, pp 349-356, 1966.

Maple Code for Figures 17 and 18

The following sample `maple` code plots the solution on $0 < t < 24$ months with data $H_0 = 190000$, $P_0 = 390250$.

```
de:=diff(P(t),t)=r*(1-P(t)/M)*P(t)-H(t);
r:=0.8:M:=780500:H0:=190000:P0:=M/2:
H:=t->H0*piecewise(t<6,1,t<12,0,t<18,1,0);
DEtools[DEplot](de,P(t),t=0..24,P=0..M,[[P(0)=P0]]);
```

Exercises 4.9

Constant Logistic Harvesting. The model

$$x'(t) = kx(t)(M - x(t)) - h$$

can be converted to the logistic model

$$y'(t) = (a - by(t))y(t)$$

by a change of variables. Find the change of variables $y = x + c$ for the following pairs of equations.

1. $x' = -3x^2 + 8x - 5$,
$y' = (2 - 3y)y$

2. $x' = -2x^2 + 11x - 14$,
$y' = (3 - 2y)y$

3. $x' = -5x^2 - 19x - 18$,
$y' = (1 - 5y)y$

4. $x' = -x^2 + 3x + 4$,
$y' = (5 - y)y$

Periodic Logistic Harvesting. The periodic harvesting model

$$x'(t) = 0.8x(t)\left(1 - \frac{x(t)}{780500}\right) - H(t)$$

is considered with H defined by

$$H(t) = \begin{cases} 0 & 0 < t < 5, \\ H_0 & 5 < t < 6, \\ 0 & 6 < t < 17, \\ H_0 & 17 < t < 18, \\ 0 & 18 < t < 24. \end{cases}$$

The project is to make a computer graph of the solution on $0 < t < 24$ for various values of H_0 and $x(0)$. See Figures 17 and 18 and the corresponding examples.

5. $H_0 = 156100$, $P(0) = 300000$

6. $H_0 = 156100$, $P(0) = 800000$

7. $H_0 = 800100$, $P(0) = 90000$

8. $H_0 = 800100$, $P(0) = 100000$

von Bertalanffy Equation. Karl Ludwig von Bertalanffy (1901-1972) derived the equation $\frac{dL}{dt} = r_B(L_\infty - L(t))$ in 1938 from simple physiological arguments. It is a widely used growth curve, especially important in fisheries studies. The symbols:

t	time,
$L(t)$	length,
r_B	growth rate,
L_∞	expected length for zero growth.

9. Solve $\frac{dL}{dt} = 2(10 - L)$, $L(0) = 0$. The answer is the length in inches of a fish over time, with final adult size 10 inches.

10. Solve von Bertalanffy's equation to obtain the model

$$L(t) = L_\infty\left(1 - e^{-r_B(t-t_0)}\right).$$

11. Assume von Bertalanffy's model. Suppose field data $L(0) = 0$, $L(1) = 5$, $L(2) = 7$. Display the nonlinear regression details which determine $t_0 = 0$, $L_\infty = 25/3$ and $r_B = \ln(5/2)$.

12. Assume von Bertalanffy's model with field data $L(0) = 0$, $L(1) = 10$, $L(2) = 13$. Find the expected length L_∞ of the fish.