Elementary Differential Equations

Introduction

After being lazy in my(Glanz) undergraduate year when learning Differential Equations, I have to suffer to learn from zero again when I want to learn at higher degree, thus this is the summarization to help me first, then others to understand Differential Equations. Hopefully the knowledge could be used to make breakthrough and innovations for the nature and animals as well (not only human).

An equation involving one dependent variable and its derivatives with respect to one or more independent variables is called a differential equation.

Many of the general laws of nature—in physics, chemistry, biology, and astronomy—find their most natural expression in the language of differential equations.

Applications also abound in mathematics itself, especially in geometry, and in engineering, economics, and many other fields of applied science.

If $y = f(x)$ is a given function, then its derivative dy/dx can be interpreted as the rate of change of y with respect to x.

According to Newton's second law of motion, the acceleration a of a body of mass m is proportional to the total force F acting on it, with $1/m$ as the constant of proportionality, so that $a = F/m$ or

$F = ma$

Suppose , for instance , that a body of mass m falls freely under the influence of gravity alone.

- In this case the only force acting on it is mg, where g is the acceleration due to gravity.
- If y is the distance down to the body from some fixed height, then its velocity $v = dy/dt$ is the rate of change of position
- Its acceleration $a = \frac{dv}{dt} = \frac{d^2y}{dt^2}$ is the rate of change of velocity

$$F = m\frac{d^2y}{dt^2}$$

$$mg = m\frac{d^2y}{dt^2}$$

$$g = \frac{d^2y}{dt^2} \qquad (*)$$

If we alter the situation by assuming that air exerts a resisting force proportional to the velocity , then the total force acting on the body is $mg - k\frac{dy}{dt}$, and becomes

$$m\frac{d^2y}{dt^2} = mg - k\frac{dy}{dt} \qquad (**)$$

Equations $(*)$ and $(**)$ are the differential equations that express the essential attributes of the physical processes under consideration.

For this differential equations:

$$m\frac{d^2y}{dt^2} = mg - k\frac{dy}{dt}\frac{dy}{dt} = -ky\frac{d^2y}{dt^2} = 0$$

- The dependent variable in each equations above is y, notice that $\frac{dy}{dt}$ is the derivative of y with respect to t.

Ordinary Differential Equation

An ordinary differential equation is one in which there is only one independent variable , so that all the derivatives occurring in it are ordinary derivatives.

The order of a differential equation is the order of the highest derivative present.

Partial Differential Equation

A partial differential equation is one involving more than one independent variable, so that the derivatives occurring in it are partial derivatives.

For example , if $w = f(x, y, z, t)$ is a function of time and the three rectangular coordinates of a point in space , then the following are partial differential equations of the second order:

$$\frac{\partial^2 w}{\partial x^2} + \frac{\partial^2 w}{\partial y^2} + \frac{\partial^2 w}{\partial z^2} = 0a^2\left(\frac{\partial^2 w}{\partial x^2} + \frac{\partial^2 w}{\partial y^2} + \frac{\partial^2 w}{\partial z^2}\right) = \frac{\partial w}{\partial t}a^2\left(\frac{\partial^2 w}{\partial x^2} + \frac{\partial^2 w}{\partial y^2} + \frac{\partial^2 w}{\partial z^2}\right) = \frac{\partial^2 w}{\partial t^2}$$

These equations above are also classical, and are called Laplace's equation, the heat equation, and the wave equation, respectively.

General Remarks and Solutions

The general ordinary differential equation of the nth order is

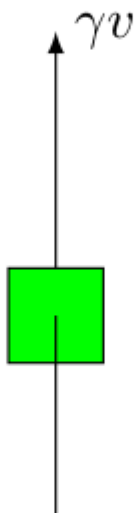
$$F\left(x, y, \frac{dy}{dx}, \frac{d^2y}{dx^2}, \ldots, \frac{d^ny}{dx^n},\right) = 0$$

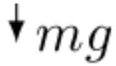
or, using the prime notation for derivatives,

$$F(x, y, y', y'', \ldots, y^{(n)}) = 0$$

Example: A Falling Object

Imagine a leaf falling from the tree with the free-body diagram of the forces on a falling object simplified below:





1. Let v represent the velocity of the falling object, velocity will presumably change with time, thus v can be assumed as a function of t.
2. The physical law that governs the motion of objects is Newton's second law, which can be expressed by the equation

$$F = ma$$

with m is the mass of the object, a is its acceleration, and F is the net force exerted on the object. We measure m in kilograms, a in meters/second ^{2}, and F in newtons.

a is the changing of speed per second, denoted by

$a = \frac{dv}{dt}$

thus we can rewrite Newton's second law in the form

$$F = m\frac{dv}{dt}$$

3. Gravity exert a force equal to the weight of the object (mg), where g is the acceleration due to gravity. g has been determined experimentally to be approximately equal to $9.8 \ m/s^2$ near the earth's surface.
4. The drag force has the magnitude γv, where γ is a constant called the drag coefficient.
5. When writing an expression for the net force F, the gravity always acts in the downward (positive) direction, whereas drag acts in the upward(negative) direction

$$F = mg - \gamma v$$
$$m\frac{dv}{dt} = mg - \gamma v$$

The equation above is a mathematical model of an object falling in the atmosphere near sea level.

6. The constants m and γ depend very much on the particular pbject that is falling, and they are different for different objects. They can be considered as parameters as they may take on a range of values.
7. To solve the equation above, we need to find a function $v = v(t)$ that satisfies the equation.

$$\frac{dv}{dt} = 9.8 - \frac{v}{5}$$

with an assumption that $m = 10$ kg and γ = 2 kg/s.

Further Analysis

1. If $v = 40$ then $\frac{dv}{dt} = 1.8$, this means that the slope of a solution $v = v(t)$ has the value 1.8 at any point where $v = 40$ (a positive slope).
2. If $v = 60$ then $\frac{dv}{dt} = -2.2$, this means that the slope of a solution $v = v(t)$ has the value -2.2 at any point where $v = 60$ (a negative slope).
3. The constant function $v(t) = 49$ does not change with time and is called an equilibrium solution that separate the negative slopes above it (the falling object slows down as it falls) and the positive slopes below it (the speed of the falling object increases as it falls)
4. All solutions seem to be converging to the equilibrium solution as t increases.

Direction Fields

Direction fields are valuable tools in studying the solutions of differential equations of the form

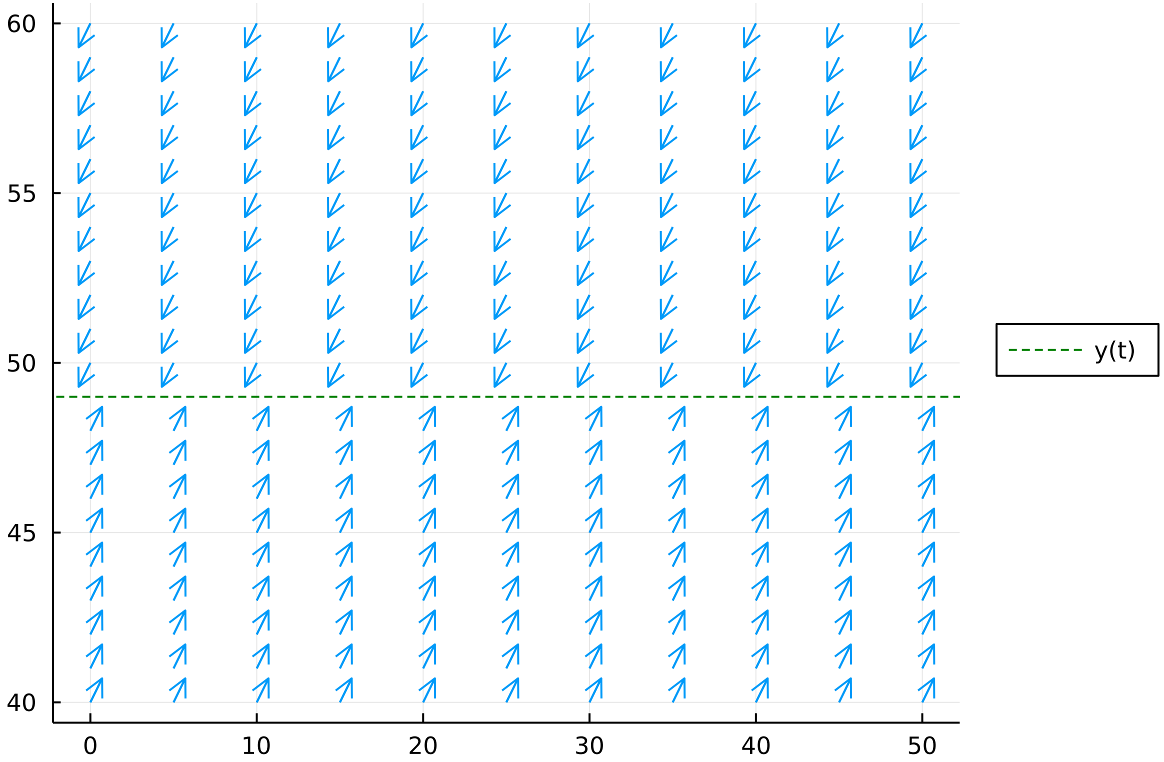
$$\frac{dy}{dt} = f(t, y)$$

- f is referred as a rate function
- Direction field can be used to find information about the long term behavior of the solution
- The construction of a direction field is often a useful first step in the investigation of a differential equation
- You should use a computer to draw a direction field

See the direction field below (slope field)

In [19]:
```
using LinearAlgebra, Plots

xs = 0:5:50
ys = 40:1:60

# dx/dy = f(y,x)
df(x, y) = normalize([9.8-(y/5), 9.8-(y/5)])

xxs = [x for x in xs for y in ys]
yys = [y for x in xs for y in ys]

Plots.quiver(xxs, yys, quiver=df)
plot!([49], seriestype="hline", linestyle=:dash, color=:green, label="y(t)", legend=:outerright)
```

Out[19]:



Example

A pond initially contains 1,000,000 gal of water and an unknown amount of an undesirable chemical. Water containing 0.01 g of this chemical per gallon flows into the pond at a rate of 300 gal/h. The mixture flows out at the same rate, so the amount of water in the pond remains constant. Assume that the chemical is uniformly distributed throughout the pond.

(a) Write the differential equation for the amount of chemical in the pond at any time

(b) How much of the chemical will be in the pond after a very long time? Does this limiting amount depend on the amount that was present initially

Solution

(a)

$$\frac{dq}{dt} = 300 \times (0.01) - 300 \times (1/1,000,000) \times q$$
$$\frac{dq}{dt} = 300\,((0.01) - 10^{-6}q)$$

with $q(t)$ represents the amount of water inside the pond at time t / the rate function of water inside the pond.

- q in g (grams)
- t in h (hour)

(b) The amount that was present initially does not affect the chemical amount in the pond, thus

$q \to 10^4$g

In [4]:
```
using LinearAlgebra, Plots

xs = 0:5:50
ys = -5*10^(-4):5*10^(-5):2*10^(-4)

# dx/dy = f(y,x)
df(x, y) = normalize([3-(3*10^(-4)y), 3-(3*10^(-4)y)])

xxs = [x for x in xs for y in ys]
yys = [y for x in xs for y in ys]

Plots.quiver(xxs, yys, quiver=df)
plot!([10^(-4)], seriestype="hline", linestyle=:dash, color=:green, label="y(t)", legend=:outerright)
```

```
WARNING: both Plots and CairoMakie export "plot!"; uses of it in module Main must be qualified

UndefVarError: plot! not defined

Stacktrace:
 [1] top-level scope
   @ In[4]:13
 [2] eval
   @ ./boot.jl:373 [inlined]
 [3] include_string(mapexpr::typeof(REPL.softscope), mod::Module, code::String, filename::String)
   @ Base ./loading.jl:1196
```

Example

Newton's law of cooling states that the temperature of an object changes at a rate proportional to the difference between the temperature of an object itself and the temperature of its surroundings (the ambient air temperature in most cases). Suppose that the ambient temperature is 75^0F and that the rate constant is 0.05 (min)$^{-1}$.

Write a differential equation for the temperature of the object at any time. Note that the differential equation is the same whether the temperature of the object is above or below the ambient temperature.

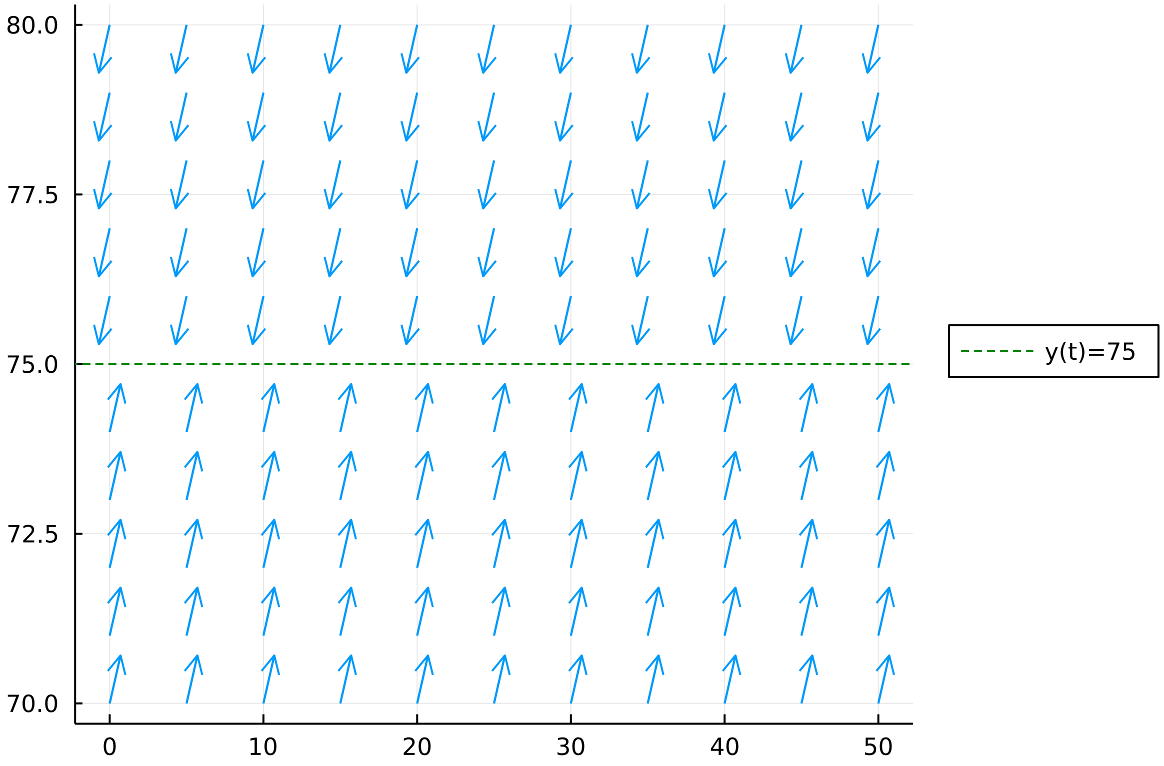
Solution

The differential equation that represent the temperature of the object at any time is

$$\frac{du}{dt} = -0.05(u - 75)$$

In [5]:
```
using LinearAlgebra, Plots

xs = 0:5:50
ys = 70:1:80

# dx/dy = f(y,x)
df(x, y) = normalize([-0.05*(y-75), -0.05*(y-75)])

xxs = [x for x in xs for y in ys]
yys = [y for x in xs for y in ys]

Plots.quiver(xxs, yys, quiver=df)
plot!([75], seriestype="hline", linestyle=:dash, color=:green, label="y(t)=75", legend=:outerright)
```

Out[5]:



Example

A certain drug is being administered intravenously to a hospital patient. Fluid containing 5 mg/cm^3 of the drug enters the patient's bloodstream at a rate of 100 cm^3/h. The drug is absorbed by body tissues or otherwise leaves the bloodstream at a rate proportional to the amount present, with a rate constant of $0.4(\mathrm{h})^{-1}$.

(a) Assuming that the drug is always uniformly distributed throughout the bloodstream, write a differentail equation for the amount of the drug that is present in the bloodstream at any time

(b) How much of the drug is present in the bloodstream after a long time

Solution

(a) $\frac{dq}{dt} = 500 - 0.4q$

with q in mg represents the amount of the drug that is present in the bloodstream at any time. t in h.

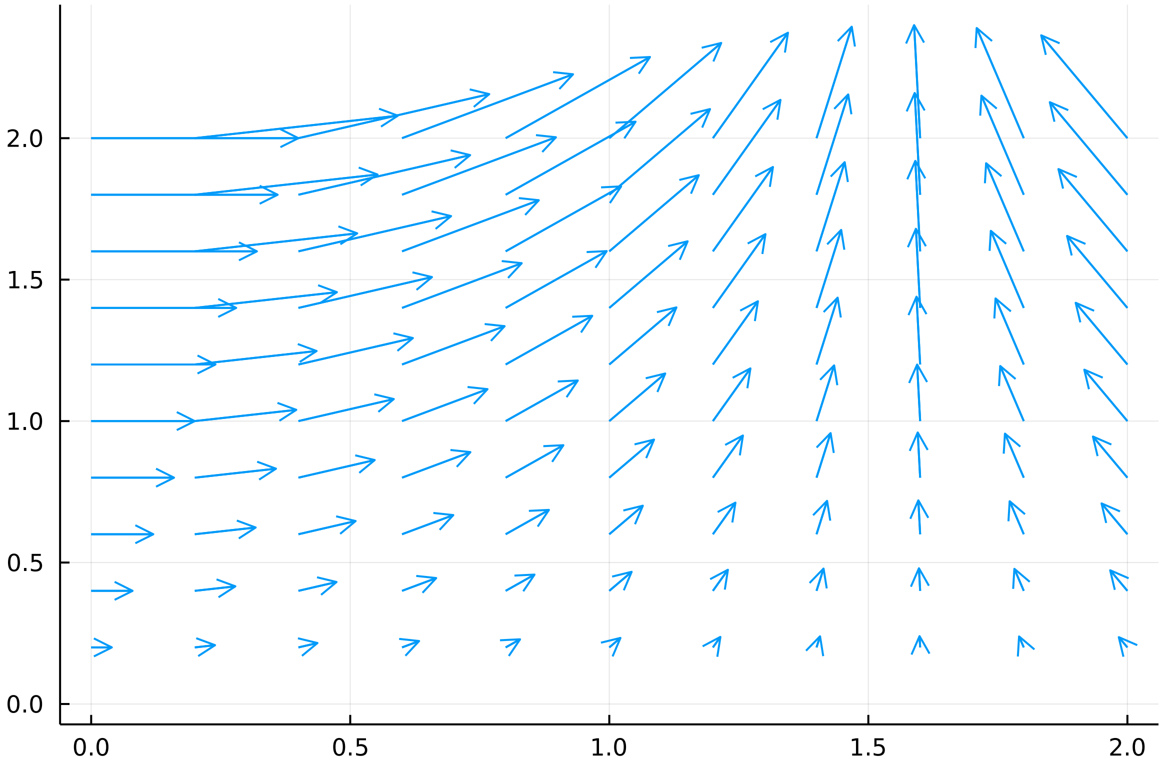
(b) After a long time then

$$\begin{aligned} \frac{dq}{dt} &= 500 - 0.4q \\ 0 &= 500 - 0.4q \\ q &= 1250 \end{aligned}$$

thus, $q \rightarrow 1250$ mg

In [15]:
```
using Plots
gr()

meshgrid(x, y) = (repeat(x, outer=length(y)), repeat(y, inner=length(x)))
x, y = meshgrid(0:0.2:2, 0:0.2:2)
u = @. cos(x) * y
v = @. sin(x) * y

# we can simply broadcast-multiply u and v by a scale constant.
scale = 0.2
u = @. scale * cos(x) * y
v = @. scale * sin(x) * y

quiver(x, y, quiver=(u, v))
```

Out[15]:



In [ ]:
```
using Plots
gr()
#gr(size=(600,400))

function example()
  X = range(-2, stop=2, length=100)
  Y = range(-2, stop=2, length=100)
  f(x, y) = x^3 - 3x + y^2
  contour(X, Y, f)

  x = range(-2, stop=2, length=11)
  y = range(-2, stop=2, length=11)
  df(x, y) = [3x^2 - 3; 2y] / 25
  quiver!(repeat(x,11), vec(repeat(y',11)), quiver=df, c=:blue)

  xlims!(-2, 2)
  ylims!(-2, 2)

  # to save as png uncomment the code below and gr(size=(600,400))
  #png("example")
end

example()
```

In [ ]:
```
using Plots, Zygote

f(x) =  -x^2 + 2x + 2
gradient_line(f, x₀) = (x -> f(x₀) + f'(x₀)*(x-x₀))
plot(f, -3:0.1:5, label = "f(x) = -x² + 2x + 2", xlabel = "x", ylabel = "f(x)");

@gif for i in 1:3
    plot(gradient_line(f, i), -3:0.1:0.5, label = "f'($i)", color = 2);
end
```

Constructing Mathematical Models

It is necessary first to formulate the appropriate differential equation that describes, or models, the problem being investigated.

1) Identify the independent and dependent variables and assign letters to represent them. Often the independent variable is time

2) Choose the units of measurement for each variable.

3) Articulate the basic principle that underlies or governs the problem you are investigating. It could be a widely recognized physical law, such as Newton's law of motion, or it may be a more speculative assumption that may be based on your own experience or observations.

4) Express the principle or law in step 3 in terms of the variables you chose in step 1.

5) Make sure that each term in the equation has the same physical units

6) The result of step 4 is a single differential equation, in more complex problems the resulting mathematical model may be involving a system of several differential equations.

Solutions of Some Differential Equations

Consider this general form of a differential equation

$\frac{dy}{dt} = ay - b$

where a and b are given constants.

Example

Consider the equation

$\frac{dp}{dt} = 0.5p - 450$

which describes the interaction of certain populations of field mice and owls.

To solve the equation above, we need to find functions $p(t)$ that reduce it to an obvious identity.

$$\frac{dp}{dt} = \frac{p - 900}{2}$$

if $p \neq 900$

$$\begin{aligned} \frac{dp/dt}{p-900} &= \frac{1}{2} \\ \frac{d}{dt}\ln|p-900| &= \frac{1}{2} \\ \ln|p-900| &= \frac{t}{2} + C \end{aligned}$$

where C is an arbitrary constant of integration. Therefore, by taking the exponential of both sides, we obtain

$$\begin{aligned} |p-900| &= e^{(t/2)+C} = e^C e^{t/2} \\ p-900 &= \pm e^C e^{t/2} \\ p &= 900 + ce^{t/2} \end{aligned}$$

where $c = \pm e^C$ is also an arbitrary (nonzero) constant.

Initial Value Problem

Most often, we specify a point that must lie on the graph (from experiment or observation), such as value 850 at time $t = 0$. In other words, the graph of the solution must pass through the point $(0, 850)$.

Symbolically, we can express this condition as

$p(0) = 850$

Substituting into the equation of p above we obtain

$$\begin{aligned} 850 &= 900 + ce^{0/2} \\ 850 &= 900 + c \\ c &= -50 \end{aligned}$$

Then by inserting the value of $c = -50$ we obtain the desired solution

$$p = 500 - 50e^{t/2}$$

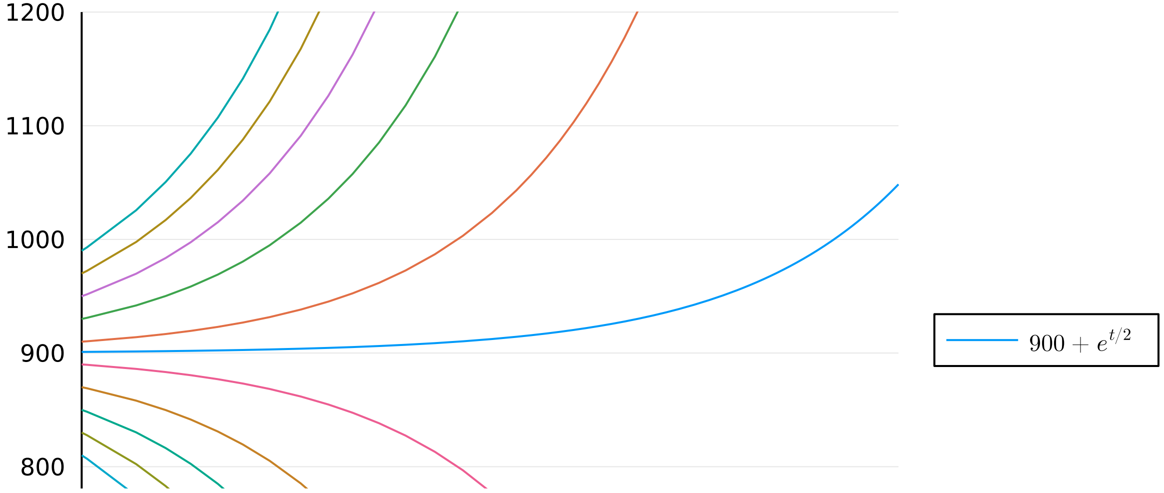
The additional condition at time $t = 0$ that we used to determine c is an example of an initial condition.

The initial value problem will generally consisting of these two (the differential equation and the initial condition):

$$\frac{dp}{dt} = Cp + D$$

$$p(0) = K$$

In [1]:
```
using Plots, LaTeXStrings, Plots.PlotMeasures
gr()

f(x) = 900 + exp(x / 2)

p = plot(f, 0, 10, xticks=false, xlims=(0, 10), ylims=(600, 1200),
    bottom_margin=10mm, label=L" 900 + e^{t/2}", framestyle=:zerolines,
    legend=:outerright)

for i = 10:20:100
    g(x) = 900 + i * exp(x / 2)
    plot!(p, g, xticks=false, xlims=(0, 10), ylims=(600, 1200),
        bottom_margin=10mm, label="", framestyle=:zerolines,
        legend=:outerright)
end

for i = 10:20:100
    g(x) = 900 - i * exp(x / 2)
    plot!(p, g, xticks=false, xlims=(0, 10), ylims=(600, 1200),
        bottom_margin=10mm, label="", framestyle=:zerolines,
        legend=:outerright)
end

display(p)
```



600 |

General Initial Value Problem

Consider the more general problem consisting of the differential equation:

$\frac{dy}{dt} = ay - b$

and the initial condition

$y(0) = y_0$

where y_0 is an arbitrary initial value. We can write the differential equation as

$$\begin{aligned} \frac{dy/dt}{y - (b/a)} &= a \\ \ln|y - (b/a)| &= at + C \\ y &= (b/a) + ce^{at} \end{aligned}$$

where C is arbitrary, and $c = \pm e^C$ is also arbitrary.

Observe that $c = 0$ corresponds to the equilibrium solution $y = b/a$.

The initial condition requires that $c = y_0 - (b/a)$, thus the solution of the initial value problem is

$$y = (b/a) + [y_0 - (b/a)]e^{at}$$

For $a \neq 0$, the equation $y = (b/a) + ce^{at}$ is called the general solution.

- The geometric representation of the general solution is an infinite family of curves called integral curves. Each integral curve is associated with a particular value of c and is the graph of the solution corresponding to that value of c.

Example: A Falling Object

Consider a falling airplane' wing (of a cheap airline) due to crash of mass $m = 100$ kg and drag coefficient $\gamma = 2$ kg/s. Then the equation of motion becomes

$\frac{dv}{dt} = 9.8 - \frac{v}{50}$

Suppose this object is dropped from a height of $11,000$ m. Find its velocity at any time t. How long will it take to fall to the ground, and how fast will it be moving at the time of impact?

Solution

The first step is to state an appropriate initial condition. We will use the initial condition

$v(0) = 0$

First rewrite the equation

$$\begin{aligned}
\frac{dv}{dt} &= 9.8 - \frac{v}{50} \\
\frac{dv}{dt} &= \frac{490 - v}{50} \\
\frac{dv/dt}{490 - v} &= \frac{1}{50} \\
\frac{dv/dt}{v - 490} &= -\frac{1}{50} \\
\ln|v - 490| &= -\frac{t}{50} + C \\
|v - 490| &= e^{(-t/50)+C} \\
|v - 490| &= e^C e^{-t/50} \\
v - 490 &= \pm e^C e^{-t/50} \\
v &= 490 + ce^{-t/50}
\end{aligned}$$

$v = 490 + ce^{-t/50}$ is the general solution, where c is arbitrary.

To determine c, we substitute $t = 0$ and $v = 0$ from the initial condition, with the result that $c = -490$.

Then the solution of the initial value problem is

$$v = 490(1 - e^{-t/50})$$

The equation above gives the velocity of the falling object at any positive time (before it hits the ground).

To find the velocity of the object when it hits the ground, we need to know the time at which impact occurs.

Note that the distance x the object has fallen is related to its velocity v by the equation $v = \frac{dx}{dt}$, or

$$\frac{dx}{dt} = 490(1 - e^{-t/50})$$
$$x = 490t + 2450e^{-t/50} + c$$

the object starts to fall when $t = 0$, so we know that $x = 0$ when $t = 0$ thus

$$x = 490t + 2450e^{-t/50} + c$$
$$0 = 490(0) + 2450e^{-(0)/50} + c$$
$$c = -2450$$

the distance the object fallen at time t is given by

$$x = 490t + 2450e^{-t/50} - 2450$$

Let T be the time at which the object hits the ground; then $x = 11,000$ when $t = T$. By substituting these values, we obtain the equation

$$490T + 2450e^{-T/50} - 13450 = 0$$

The value of T can be approximated by a numerical process (finding the root of an equation), thus

$T \cong 24.38$ s (the result is obtained from the code below to find the root).

At time $T \cong 24.38$ s the corresponding velocity is

$v_T = 490(1 - e^{-T/50})$

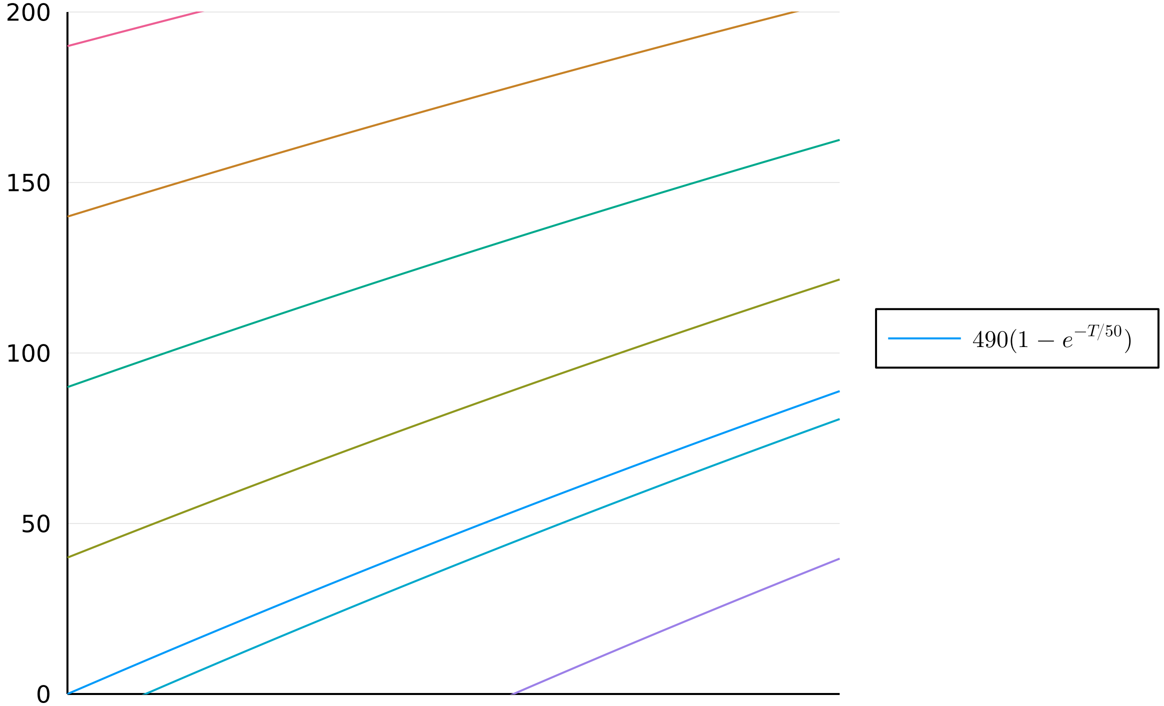
In [20]:
```
# The find_zeros function can be used to search for all zeros in a specified interval.
# The basic algorithm essentially splits the interval into many subintervals.

using Roots

f(x) = 490x + 2450*(exp(-x/50)) - 13450
find_zeros(f, -1000,1000)
```

Out[20]:
```
2-element Vector{Float64}:
 -188.21393471896695
   24.378388299422703
```

In [21]:
```
490*(1-exp(-24.38/50))
```

Out[21]:
```
189.09175297966192
```

In [15]:
```
using Plots, LaTeXStrings, Plots.PlotMeasures
gr()

f(x) = 490*(1-exp(-x/50))

p = plot(f, 0, 10, xticks=false, xlims=(0, 10), ylims=(0, 100),
    bottom_margin=10mm, label=L"490(1 - e^{-T/50})", framestyle=:zerolines,
    legend=:outerright)

for i = 50:50:1000
    g(x) = 490 - i * exp(-x / 50)
    plot!(p, g, xticks=false, xlims=(0, 10), ylims=(0, 200),
        bottom_margin=10mm, label="", framestyle=:zerolines,
        legend=:outerright)
end

display(p)
```



Example

Consider the falling object of mass 10 kg, but assume now that the drag force is proportional to the square of the velocity.

(a) If the limiting velocity is 49 m/s, show that the equation of motion can be written as

$dv/dt = [(49)^2 - v^2]/245$

(b) If $v(0) = 0$, find an expression for $v(t)$ at any time

(c) Plot your solution from part (b)

(d) Find the distance $x(t)$ that the object falls in time t

(e) Find the time T it takes the object to fall 300 m

Solution

(a)

$$
\begin{aligned}
m\frac{dv}{dt} &= mg - \gamma v \\
m\frac{dv}{dt} &= mg - v^2 \cdot v \\
\frac{dv}{dt} &= g - \frac{v^2 \cdot v}{m} \\
\frac{dv}{dt} &= 9.8 - \frac{49 \cdot v^2}{10} \\
\frac{dv}{dt} &= \frac{98 - 49v^2}{10} \\
\frac{dv}{dt} &= \frac{(49)^2 - v^2}{245}
\end{aligned}
$$

(b)

$$\frac{dv}{dt} = \frac{49^2 - v^2}{245}$$

Example: Decay Rate

A radioactive material, such as isotope thorium-234, disintegrates at a rate proportional to the amount currently present. If $Q(t)$ is the amount present at time t, then $dQ/dt = -rQ$, where $r > 0$ is the decay rate.

(a) If 100 mg of thorium-234 decays to 82.04 mg in 1 week, determine the decay rate r.

(b) Find an expression for the amount of thorium-234 present at any time t.

(c) Find the time required for the thorium-234 to decay to one-half its original amount.

Solution

(a) We will denote the variable t as time in day, thus

$$\begin{aligned}
\frac{dQ}{dt} &= -rQ \\
\frac{dQ/dt}{Q} &= -r \\
ln|Q| &= -rt + C \\
Q(t) &= ce^{-rt}
\end{aligned}$$

with c is an arbitrary constant. We know the initial value condition is that the thorium-234 at the beginning has a mass of 100 mg then after 1 week(7 days) the mass becomes 82.04 mg, thus

$$\begin{aligned}
Q(0) &= ce^{-rt} \\
100 &= ce^{-r \cdot 0} \\
c &= 100
\end{aligned}$$

we obtain the constant $c = 100$. Now we can calculate the decay rate

$$
\begin{aligned}
Q(7) &= ce^{-rt} \\
82.04 &= 100e^{-r \cdot 7} \\
\frac{82.04}{100} &= e^{-r \cdot 7} \\
0.8204 &= e^{-7r} \\
ln|0.8204| &= ln|e^{-7r}| \\
-0.197963253 &= -7r \\
r &\cong 0.02828
\end{aligned}
$$

the decay rate is $r \cong 0.02828$.

(b) The amount of thorium-234 present at any time t can be represented with $Q(t) = e^{-0.02828t}$.

(c) To find the time required for 100 mg of thorium-234 to become one-half its original amount

$$
\begin{aligned}
Q(t) &= ce^{-rt} \\
50 &= 100e^{-0.02828t} \\
\frac{50}{100} &= e^{-0.02828t} \\
0.5 &= e^{-0.2828t} \\
ln|0.5| &= ln|e^{-0.02828t}|
\end{aligned}
$$

Example: Electric Circuit

Consider an electric circuit containing a capacitor, resistor, and battery. The charge $Q(t)$ on the capacitor satisfies the equation

$$R\frac{dQ}{dt} + \frac{Q}{C} = V$$

where R is the resistance, C is the capacitance, and V is the constant voltage supplied by the battery.

(a) If $Q(0) = 0$, find $Q(t)$ at any time t, and sketch the graph of Q versus t.

(b) Find the limiting value Q_t that $Q(t)$ approaches after a long time

(c) Suppose that $Q(t_1) = Q_t$ and that at time $t = t_1$ the battery is removed and the circuit closed again. Find $Q(t)$ for $t > t_1$ and sketch its graph.

Solution

(a)

$$\begin{aligned} R\frac{dQ}{dt} + \frac{Q}{C} &= V \\ R\frac{dQ}{dt} &= V - \frac{Q}{C} \\ R\frac{dQ}{dt} &= \frac{CV - Q}{C} \\ \frac{1}{CV - Q}\frac{dQ}{dt} &= \frac{1}{RC} \\ \frac{dQ/dt}{CV - Q} &= \frac{1}{RC} \\ \frac{dQ}{CV - Q} &= \frac{1}{RC}\ dt \\ ln|CV - Q| &= \frac{t}{RC} + K \\ CV - Q &= \frac{-t}{RC} + K \\ CV - Q &= e^{-t/RC + K} \\ CV - Q &= Ke^{-t/RC} \\ Q(t) &= CV - Ke^{-t/RC} \end{aligned}$$

with K is an arbitrary constant (at this point we do not chose a negative sign, The inner absolute value gets resolved to fit the signs at the initial point), thus we can find the value since we know that the initial value of $Q(0) = 0$, hence

$$\begin{aligned} Q(0) &= CV - Ke^{-t/RC} \\ 0 &= CV - Ke^{-0/RC} \\ Ke^0 &= CV \\ K &= CV \end{aligned}$$

After we know that the value of constant $K = CV$, thus the equation that represent the amount of charge at time t is $Q(t) = CV(1 - e^{-t/RC})$.

(b) With basic calculus and limit function as $t \to \infty$ thus the $\lim_{t \to infty} Q(t)$:

$Q(t) \to CV = Q_L$

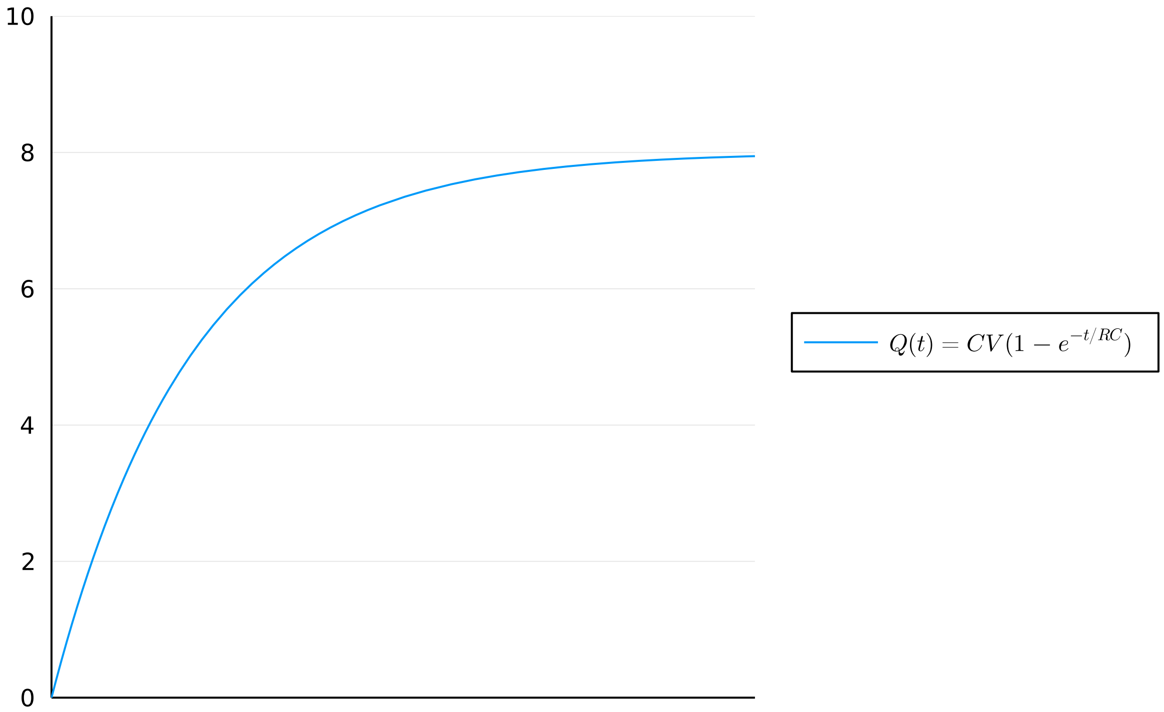
There are two variables that are of concern: the amount of the voltage supplied by the battery to the circuit and the capacitance, those two influence the amount of charge after a very long time.

(c)

In [6]:
```
# graph of Q(t)=CV(1-e^{-t/RC}), Q versus t.

using Plots, LaTeXStrings, Plots.PlotMeasures
gr()

#Constants and setup
const C,V,R = 1,8,2

f(x) = C*V*(1-exp(-x/(R*C)))

p = plot(f, 0, 10, xticks=false, xlims=(0, 10), ylims=(0, 10),
    bottom_margin=10mm, label=L"Q(t) = CV(1 - e^{-t/RC})", framestyle=:zerolines,
    legend=:outerright)
```

```
WARNING: redefinition of constant V. This may fail, cause incorrect answers, or produce other errors.
```

Out[6]:



Example: Chemical' Flow Rate in a Pond

A pond containing 1,000,000 gal of water is initially free of a certain undesirable chemical. Water containing 0.01 g/gal of the chemical flows into the pond at a rate of 300 gal/h, and water also flows out of the pond at the same rate. Assume that the chemical us uniformly distributed throughout the pond.

(a) Let $Q(t)$ be the amount of the chemical in the pond at time t. Write down an initial value problem for $Q(t)$

(b) Solve the problem in part (a) for $Q(t)$. How much chemical is in the pond after 1 year?

(c) At the end of 1 year the source of the chemical in the pond is removed; thereafter pure water flows into the pond, and the mixture flows out at the same rate as before. Write down the initial value problem that describes this new situation

(d) Solve the initial value problem in part (c). How much chemical remains in the pond after 1 additional year (2 years from the beginnning of the problem)?

(e) How long does it take for $Q(t)$ to be reduced to 10 g?

(f) Plot $Q(t)$ versus t for 3 years

Solution

Remark

Always remember that the ultimate test of any mathematical model is whether its predictions agree with observations or experimental results.

First Order Differential Equations

Method of Integrating Factors

1) We must have a general first order linear equation of this form

$$\frac{dy}{dt} + p(t)y = g(t)$$

$\frac{dy}{dt}$ can be written as y'

2) We will calculate the integrating factor $\mu(t)$

$$\mu(t) = \exp\int p(t)\, dt$$

3) The general solution can be obtained by

$$y(t) = \frac{1}{\mu(t)}\left[\int^t \mu(s)\, g(s)\, ds + c\right]$$

Example

Solve the initial value problem

$$\begin{aligned} ty' + 2y &= 4t^2 \\ y(1) &= 2 \end{aligned}$$

1) First arrange the the standard form:

$y' + (2/t)y = 4t$

2) Compute the integrating factor $\mu(t)$:

$\mu(t) = \exp\int \frac{2}{t}\, dt = 2^{2\ln|t|} = t^2$

3) Find the general solution:

$$
\begin{aligned}
y(t) &= \frac{1}{\mu(t)}\left[\int^t \mu(s)\, g(s)\, ds + c\right] \\
&= \frac{1}{t^2}\left[\int^t s^2 \cdot 4(s)\, ds + c\right] \\
&= \frac{1}{t^2}\left[\int^t 4s^3\, ds + c\right] \\
&= \frac{1}{t^2}\left[t^4 + c\right] \\
y(t) &= t^2 + \frac{c}{t^2}
\end{aligned}
$$

To satisfy the initial condition substitue $t = 1$

$2 = 1^2 + \frac{c}{\ }$

In [13]:
```
#It should be plotted using REPL to show all curves

using Plots, LaTeXStrings, Plots.PlotMeasures
gr()

f(x) = x^2 + 1/x^2

p = plot(f, -2,2, xticks=false, xlims=(-2, 2), ylims=(-2, 4),
    bottom_margin=10mm, label=L"t^{2} + \frac{c}{t^{2}}", framestyle=:zerolines,
    legend=:outerright)

    for i = 1:0.5:3
        g(x) =  x^(2) + (i / x^(2))
        plot!(g, xticks=false, xlims=(-2, 2), ylims=(-2, 4),
            bottom_margin=10mm, label="", framestyle=:zerolines,
            legend=:outerright)
    end

    for i = 1:0.5:3
        g(x) =  x^(2) - (1 / x^(2))
        plot!(g, xticks=false, xlims=(-2, 2), ylims=(-2, 4),
            bottom_margin=10mm, label="", framestyle=:zerolines,
            legend=:outerright)
    end
plot!()
```

```
cannot define function g; it already has a value

Stacktrace:
 [1] top-level scope
   @ none:0
 [2] top-level scope
   @ In[13]:12
 [3] eval
   @ ./boot.jl:373 [inlined]
 [4] include_string(mapexpr::typeof(REPL.softscope), mod::Module, code::String, filename::String)
   @ Base ./loading.jl:1196
```

Separable Equations

1) We use a process of direct integration to solve first order linear equation of the form

$$\frac{dy}{dx} = ay + b = f(x, y)$$

where a and b are constants.

2) Rewrite the differential equation into:

$$M(x, y) + N(x, y)\frac{dy}{dx} = 0$$

by setting $M(x, y) = -f(x, y)$ and $N(x, y) = 1$ the equation above will become

$$M(x) + N(y)\frac{dy}{dx} = 0M(x)\,dx + N(y)\,dy = 0$$

A separable equation can be solved by integrating the functions M and N.

Example

Show that the equation

$\frac{dy}{dx} = \frac{x^2}{1-y^2}$

is separable, and then find an equation for its integral curves.

Solution

1) First rewrite the equation into the separable form

$-x^2 + (1 - y^2)\frac{dy}{dx} = 0$

2) Recall from calculus if y is a function of x, then by the chain rule

$\frac{d}{dx}f(y) = \frac{d}{dy}f(y)\frac{dy}{dx} = f^{'}(y)\frac{dy}{dx}$

We have found the similarity with the equation 1) for the function $(1 - y^2)\frac{dy}{dx}$ at the right side of the chain rule above.

Now, we have to integrate the function $(1 - y^2)\frac{dy}{dx}$ by using the chain rule formula above.

$$\frac{d}{dx}f(y) = (1 - y^2)\frac{dy}{dx}$$
$$\frac{d}{dx}(y - y^3/3) = (1 - y^2)\frac{dy}{dx}$$

If you are confused, $f(y) = y - y^3/3$ is the function which when it is differentiated it becomes $f^{'}(y) = 1 - y^2$. Thus, we can substitute it to solve the differential equation with separable equation.

3) Now, rewrite the equation from 1) to become

$$-x^2 + (1 - y^2)\frac{dy}{dx} = 0$$

$$\frac{d}{dx}\left(- \frac{x^3}{3} \right) + \frac{d}{dx}\left(y - \frac{y^3}{3} \right) = 0$$

$$\frac{d}{dx}\left(- \frac{x^3}{3} + y - \frac{y^3}{3} \right) = 0$$

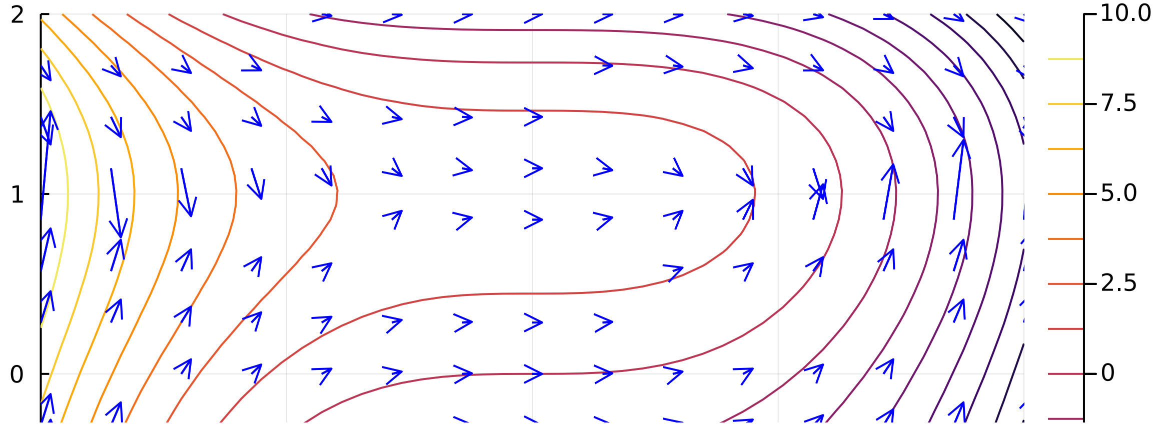
$$\text{(by integrating)}$$
$$-x^3 + 3y - y^3 = c$$

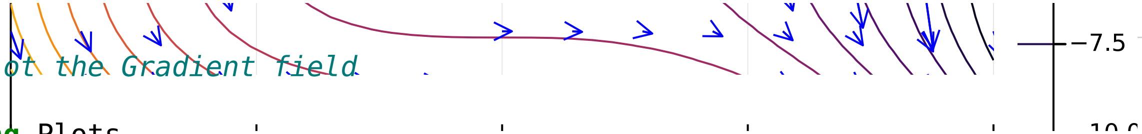
If the steps above are too hard to conceived, then back at equation 1) and just integrate

$(- x^2)dx + (1 - y^2)\frac{dy}{dx} = 0$

In [1]:
```
# Plot the slope field / direction field

using Plots
gr()
#gr(size=(600,400))

function example()
  X = range(-2, stop=2, length=50)
  Y = range(-2, stop=2, length=50)
  f(x, y) = -x^3 + 3y - y^3
  contour(X, Y, f)

  x = range(-2, stop=2, length=15)
  y = range(-2, stop=2, length=15)

  dydx_norm(x, y) = [1; x^2/(1 - y^2)] / 25

  quiver!(repeat(x,11), vec(repeat(y',11)), quiver=dydx_norm, c=:blue)

  xlims!(-2, 2)
  ylims!(-2, 2)

  # to save as png uncomment the code below and gr(size=(600,400))
  #png("example")
end

example()
```

Out[1]:



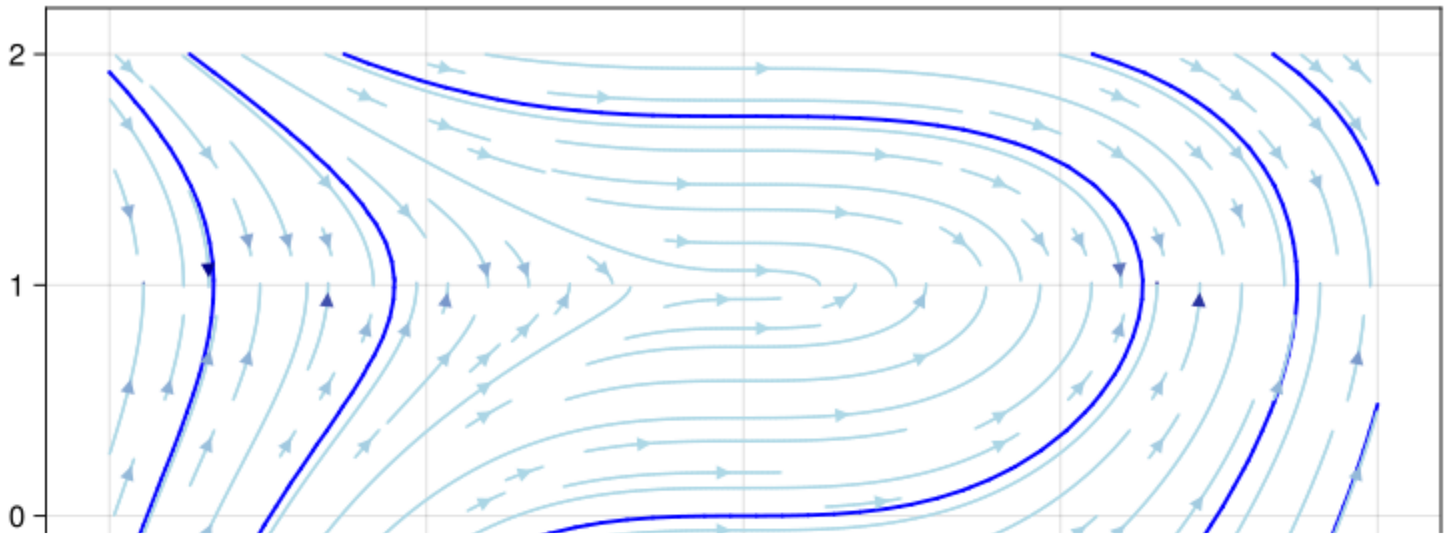


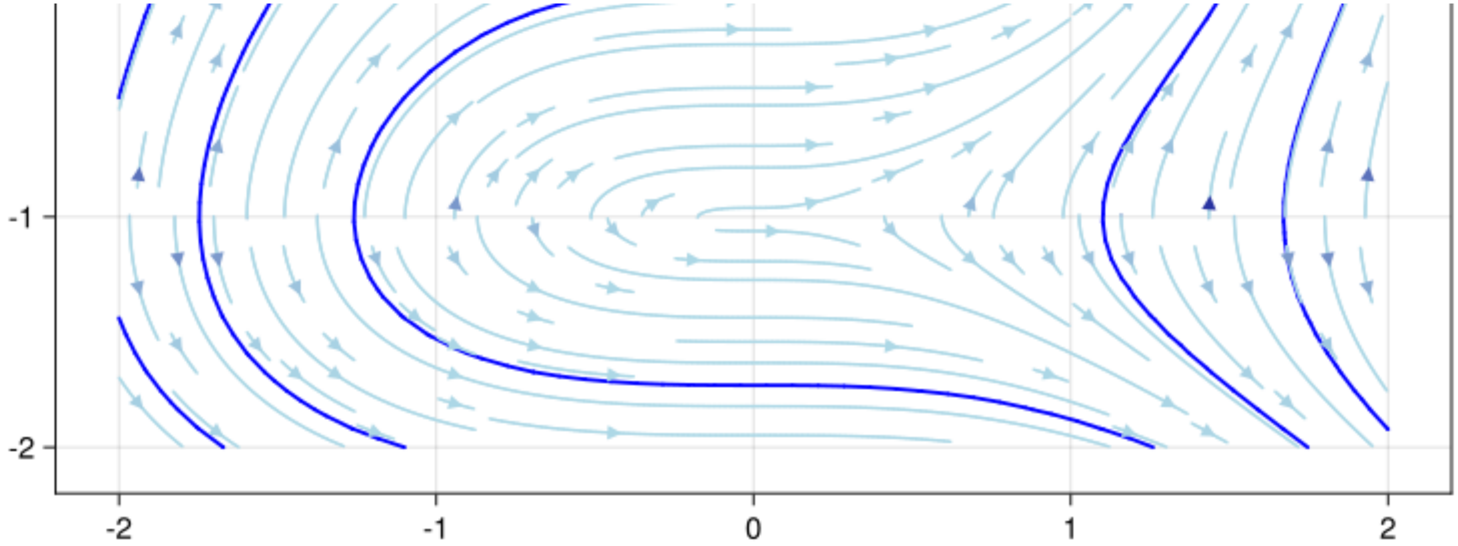
In [ ]:

```
# Plot the Gradient field

using Plots
gr()
#gr(size=(600,400))

function example()
  X = range(-2, stop=2, length=50)
  Y = range(-2, stop=2, length=50)
  f(x, y) = -x^3 + 3y - y^3
  contour(X, Y, f)

  x = range(-2, stop=2, length=15)
  y = range(-2, stop=2, length=15)
  # Calculate the partial derivative for each variable manually:
  df(x, y) = [-3x^2; 3 - 3y^2] / 25
  quiver!(repeat(x,11), vec(repeat(y',11)), quiver=df, c=:blue)

  xlims!(-2, 2)
  ylims!(-2, 2)

  # to save as png uncomment the code below and gr(size=(600,400))
  #png("example")
end

example()
```

In [ ]:
```
# Use CairoMakie in Julia REPL and the result is saved as svg
# Adjust with your own destination path

using CairoMakie

fig = Figure()
ax = Axis(fig[1,1])

X = range(-2, stop=2, length=50)
Y = range(-2, stop=2, length=50)
f(x, y) = -x^3 + 3y - y^3

contour!(X, Y, f, color = :blue, linewidth=2)

dydx(x,y) = Point2f(1, x^2/(1 - y^2))

streamplot!(dydx, -2.0..2.0, -2.0..2.0, colormap=:blues)

fig

save("/home/browni/LasthrimProjection/JupyterLab/CairoMakie/example.svg",fig)
```

The result from CairoMakie is indeed beautiful:





Example

Solve the equation

$\frac{dy}{dx} = \frac{4x - x^3}{4 + y^3}$

and draw graphs of several integral curves. Also find the solution passing through the point $(0, 1)$ and determine its interval of validity.

Solution

1) Rewrite the equation and then rearringing the terms thus we are able to integrate and solve it

$$\frac{dy}{dx} = \frac{4x - x^3}{4 + y^3}$$

$$(4 + y^3)dy = (4x - x^3)dx$$

$$\text{(integrate it)}$$

$$\frac{y^4}{4} + 4y = 2x^2 - \frac{x^4}{4}$$

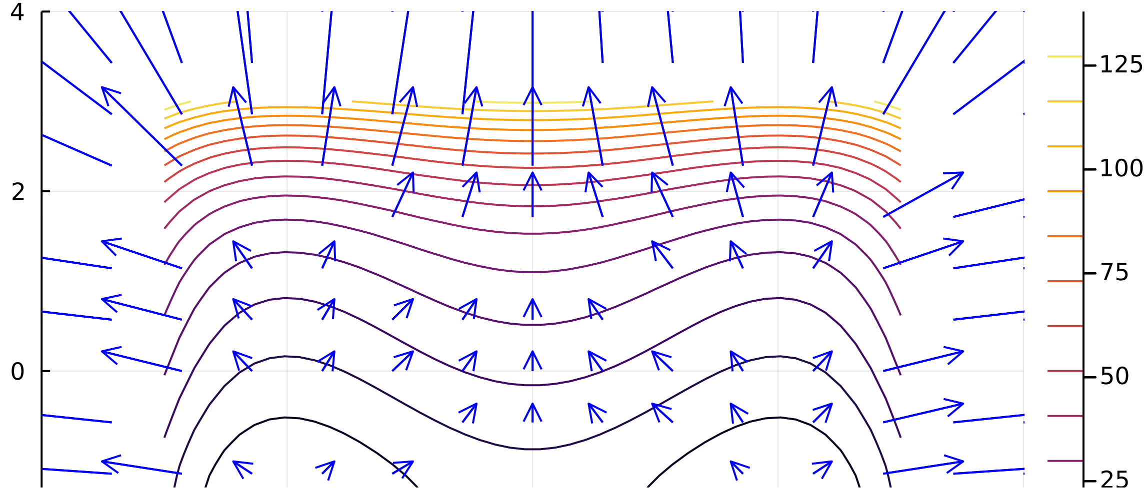
$$y^4 + 16y + x^4 - 8x^2 = c$$

with c an arbitrary constant, we can get the value of c by substituting the point $(0, 1)$ to the equation above, we set $x = 0$, and $y = 1$, hence

$$y^4 + 16y + x^4 - 8x^2 = 17$$

In [11]:
```
# Direction field and integral curves of y' = (4x - x^{3})/(4+y^{3})
using Plots
gr()
#gr(size=(600,400))

function example()
  X = range(-3, stop=3, length=50)
  Y = range(-3, stop=3, length=50)
  f(x, y) = y^4 + 16y + x^4 -8x^2
  contour(X, Y, f)

  x = range(-4, stop=4, length=15)
  y = range(-4, stop=4, length=15)
  # Calculate the partial derivative for each variable manually:
  df(x, y) = [4x^3 - 16x; 4y^3 + 16] /73
  quiver!(repeat(x,11), vec(repeat(y',11)), quiver=df, c=:blue)

  xlims!(-4, 4)
  ylims!(-4, 4)

  # to save as png uncomment the code below and gr(size=(600,400))
  #png("example")
end

example()
```

Out[11]:



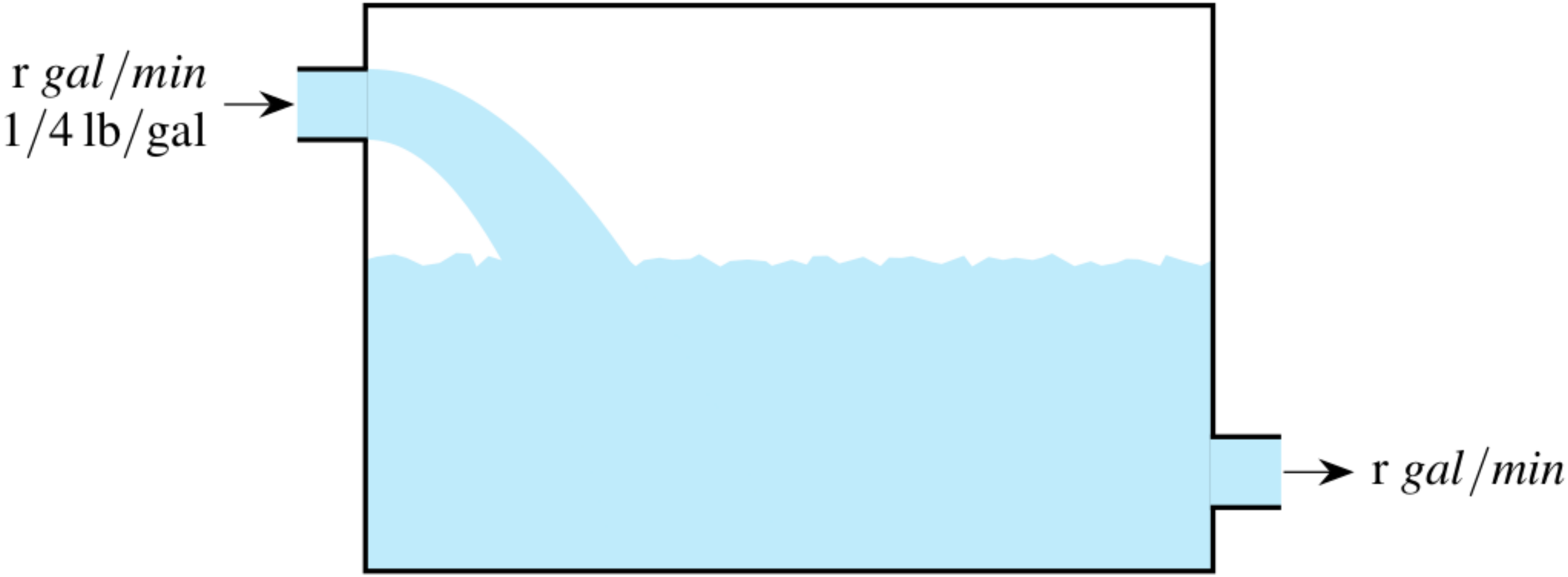
Modeling with First Order Equations

Differential equations are of interest to nonmathematicians primarily because of the possibility of using them to investigate a wide variety of problems in the physical, biological, and social sciences.

Mathematical models and their solutions lead to equations relating the variables and parameters in the problem. These equations often enable you to make predicitons about how the natural process will behave in various circumstances.

Example: Mixing

At time $t = 0$ a tank contains Q_0 lb of salt disolved in 100 gal of water.



Assume that water containing $\frac{1}{4}$ lb of salt/gal is entering the tank at a rate of r gal/min and that the well-stirred mixture is draining from the tank at the same rate. Set up the initial value problem that describes this flow process. Find the amount of salt $Q(t)$ in the tank at any time, and also find the limiting amount Q_L that is present after a very long time. If $r = 3$ and $Q_0 = 2Q_L$, find the time T after which the salt level is withing 2% of Q_L.

Also find the flow rate that is required if the value of T is not to exceed 45 min.

Solution

1) We assume that salt is neither created nor destroyed in the tank. Therefore variations in the amount of salt are due solely to the flows in and out of the tank. More precisely, the rate of change of salt in the tank, dQ/dt, is equal to the rate at which salt is flowing in minus the rate at which it is flowing out.

$\frac{dQ}{dt} = \text{rate in} - \text{rate out}$

- rate in = $\frac{r}{4}$ lb/min
- To find the rate at which salt leaves the tank, we need to multiply the concentration of salt in the tank by the rate of outflow, r gal/min.
- Since the rate in = rate out, the volume of water in the tank remains constant at 100 gal.
- Since the mixture is "well-stirred", the concentration throughout the tank is the same, $[Q(t)/100]$ lb/gal. The mixture originally in the tank will be replaced by the mixture flowing in (with concentration $\frac{1}{4}$ lb/gal.

2) The differential equation governing the process can be represented with

$$\frac{dQ}{dt} = \frac{r}{4} - \frac{rQ}{100}$$

The initial condition is

$$Q(0) = Q_0$$

To solve the differential equation:

$$
\begin{aligned}
\frac{dQ}{dt} &= \frac{r}{4} - \frac{rQ}{100} \\
&= \frac{25r - rQ}{100} \\
\frac{dQ}{25 - Q} &= \frac{r}{100} dt \\
\text{(integrate)} \\
\ln|25 - Q| &= \frac{rt}{100} + c \\
\text{(eliminate the absolute value)} \\
\ln(25 - Q) &= \pm \frac{rt}{100} + c \\
\text{(we use the minus sign)} \\
\ln(25 - Q) &= -\frac{rt}{100} + c \\
25 - Q &= e^c e^{\frac{-rt}{100}} \\
Q(t) &= 25 - ce^{\frac{-rt}{100}}
\end{aligned}
$$

where c is an arbitrary constant.

3) To satisfy the initial condition we must choose $c = Q_0 - 25$, therefore the solution of the initial value problem is

$$Q(t) = 25 + (Q_0 - 25)e^{\frac{-rt}{100}}$$

It can also be written with

$$Q(t) = 25(1 - e^{\frac{-rt}{100}}) + Q_0 e^{\frac{-rt}{100}}$$

From the equation above, we can see that if $t \to \infty$, then $Q(t) \to 25$.

- $Q(t)$ approaches the limit of 25 rapidly as r increases.
- $Q_(0)e^{\frac{-rt}{100}}$ is the portion of the original salt that remains at time t.

- $25(1 - e^{\frac{-rt}{100}})$ is the amount of salt in the tank due to the action of the flow processes.

4) If $r = 3$ and $Q_0 = 2Q_L$, to find the time T after which the salt level is within 2% of Q_L, then the differential equation becomes

$$Q(t) = 25 + (Q_0 - 25)e^{\frac{-rt}{100}}$$
$$Q(t) = 25 + (2Q_L - 25)e^{\frac{-rt}{100}}$$
$$Q(t) = 25 + (50 - 25)e^{\frac{-rt}{100}}$$
$$Q(t) = 25 + 25e^{\frac{-rt}{100}}$$

Since 2% of 25 is 0.5, to find the time T at which $Q(t)$ has the value 25.5, we will substitute $t = T$ and $Q(t) = 25.5$ thus

$$\begin{aligned} Q(T) &= 25 + 25e^{\frac{-rT}{100}} \\ 25.5 &= 25 + 25e^{\frac{-rT}{100}} \\ 0.5 &= 25e^{\frac{-rT}{100}} \\ ln(0.5/25) &= \frac{-3T}{100} \\ \frac{-3T}{100} &= \ln(0.02) \\ -3T &= 100(-3.912) \\ T &\approx 130.4 \end{aligned}$$

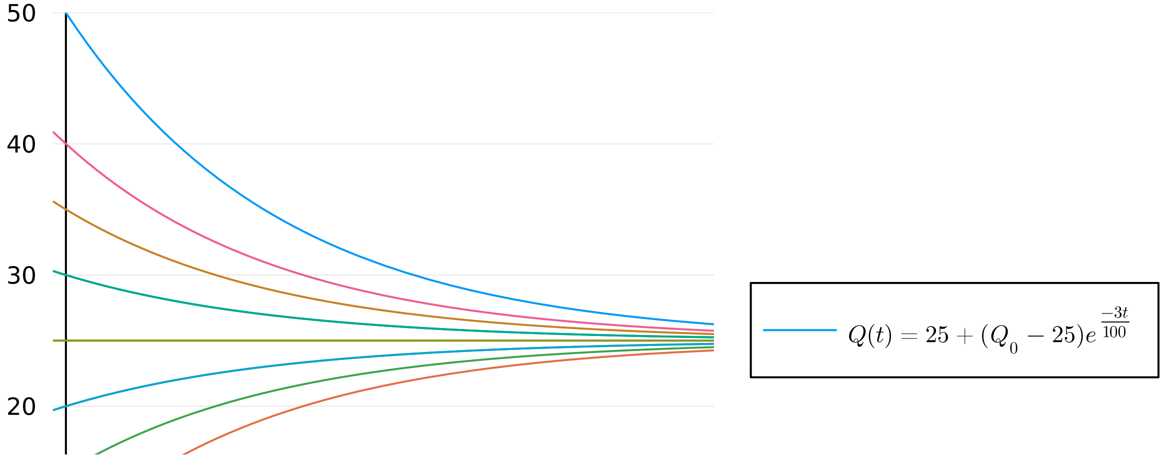
the time T after the salt level is within 2% of Q_L is 130.4 minutes.

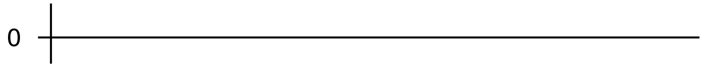
5) To determine the rate r that will make the amount of salt is within 2% of Q_L before T reach 45 minutes we can use the differential equation again, then set the same condition $Q_0 = 50,\ Q(t) = 25.5$

$$Q(T) = 25 + 25e^{\frac{-rT}{100}}$$
$$25.5 = 25 + 25e^{\frac{-45r}{}}$$

In [20]:
```
# Solutions of the initial value problem above
# with r=3 and several values of $Q_{0}$

using Plots, LaTeXStrings, Plots.PlotMeasures
gr()

f(x) = 25+25*(exp.(-3x/100))

p = plot(f, -2,100, xticks=false, xlims=(-2, 100), ylims=(0, 50),
    bottom_margin=10mm, label=L"Q(t)=25 + (Q_{0}-25) e^{\frac{-3t}{100}}", framestyle=:zerolines,
    legend=:outerright)

    for i = 10:5:30
        z1(x) =  25+(i-25)*(exp.(-3x/100))
        plot!(z1, xticks=false, xlims=(-2, 100), ylims=(-2, 50),
            bottom_margin=10mm, label="", framestyle=:zerolines,
            legend=:outerright)
    end

    for i = 10:5:30
        z2(x) =  25-(i-25)*(exp.(-3x/100))
        plot!(z2, xticks=false, xlims=(-2, 100), ylims=(-2, 50),
            bottom_margin=10mm, label="", framestyle=:zerolines,
            legend=:outerright)
    end
plot!()
```

Out[20]:





Example: Chemicals in a Pond

Consider a pond that initially contains 10 million gal of fresh water. Water containing an undesirable chemical flows into the pond at the rate of 5 million gal/yr, and the mixture in the pond flows out at the same rate. The concentration $\gamma(t)$ of chemical in the incoming water varies periodically with time according to the expression $\gamma(t) = 2 + \sin 2t$ g/gal. Construct a mathematical model of this flow process and determine the amount of chemical in the pond at any time.

Plot the solution and describe in words the effect of the variation in the incoming concentration.

Solution

1) Since the incoming and outgoing flows of water are the same, the amount of water in the pond remains constant at 10^7 gal. Let us denote time by t, measured in years, and the chemical by $Q(t)$, measured in grams. Thus

$$\frac{dQ}{dt} = \text{rate in} - \text{rate out}$$

where "rate in" and "rate out" refer to the rates at which the chemical flows into and out of the pond, respectively. The rate at which the chemical flows in is given by

$$\text{rate in} = (5 \times 10^6)\text{gal/yr}(2 + \sin 2t)\text{g/gal}$$

2) The concentration of chemical in the pond is $Q(t)/10^7$ g/gal, so the rate of flow out is

$$\text{rate out} = (5 \times 10^6)\text{gal/yr}[\frac{Q(t)}{10^7}]\text{g/gal} = \frac{Q(t)}{2}\text{ g/yr}$$

Thus we obtain the differential equation

$$\frac{dQ}{dt} = (5 \times 10^6)(2 + \sin 2t) - \frac{Q(t)}{2}$$

where each term has the units of g/yr.

3) To make the coefficients more manageable, it is convenient to introduce a new dependent variable defined by $q(t) = \frac{Q(t)}{10^6}$ or $Q(t) = 10^6 q(t)$.

This means that $q(t)$ is measured in millions of grams, or megagrams (metric tons). Substitute into the differential equation we will obtain

$$\frac{dQ}{dt} = (5 \times 10^6)(2 + \sin 2t) - \frac{Q(t)}{2}$$

$$\frac{d10^6 q}{dt} = (5 \times 10^6)(2 + \sin 2t) - \frac{10^6 q}{2}$$

$$\text{(we can eliminate } 10^6\text{)}$$

$$\frac{dq}{dt} = 10 + 5\sin 2t - \frac{q}{2}$$

$$\frac{dq}{dt} + \frac{1}{2}q = 10 + 5\sin 2t$$

Originally, there is no chemical in the pond, so the initial condition is

$q(0) = 0$

4) We will solve the differential equation by using the integrating factor

Remember that an equation of the form

$\frac{dy}{dt} + p(t)y = g(t)$

has the integrating factor $\mu(t)$:

$\mu(t) = exp\int p(t)\, dt$

Now to obtain the integrating factor :

$$\frac{dq}{dt} + \frac{1}{2}q = 10 + 5\sin 2t$$

$$\mu(t) = exp\int \frac{1}{2}\, dt$$

$$\mu(t) = e^{\frac{t}{2}}$$

After we obtain the integrating factor $\mu(t)$ we are now able to find the general solution with formula:

$$q(t) = \frac{1}{\mu(t)}\left[\int_{t_0}^{t} \mu(s)g(s)\, ds + c\right]$$

hence

$$q(t) = \frac{1}{e^{\frac{t}{2}}}\left[\int_{t_0}^{t} e^{\frac{s}{2}}(10 + 5\, sin(2s))\, ds + c\right]$$

$$q(t) = 20 - \frac{40}{17}\cos(2t) + \frac{10}{17}\sin(2t) + ce^{-t/2}$$

$$\text{(substitute the initial condition } q(0) = 0\text{)}$$

$$q(t) = 20 - \frac{40}{17}\cos(2t) + \frac{10}{17}\sin(2t) - \frac{300}{17}e^{-t/2}$$

- The exponential term in the solution is important for small t, but it diminishes rapidly as t increases.

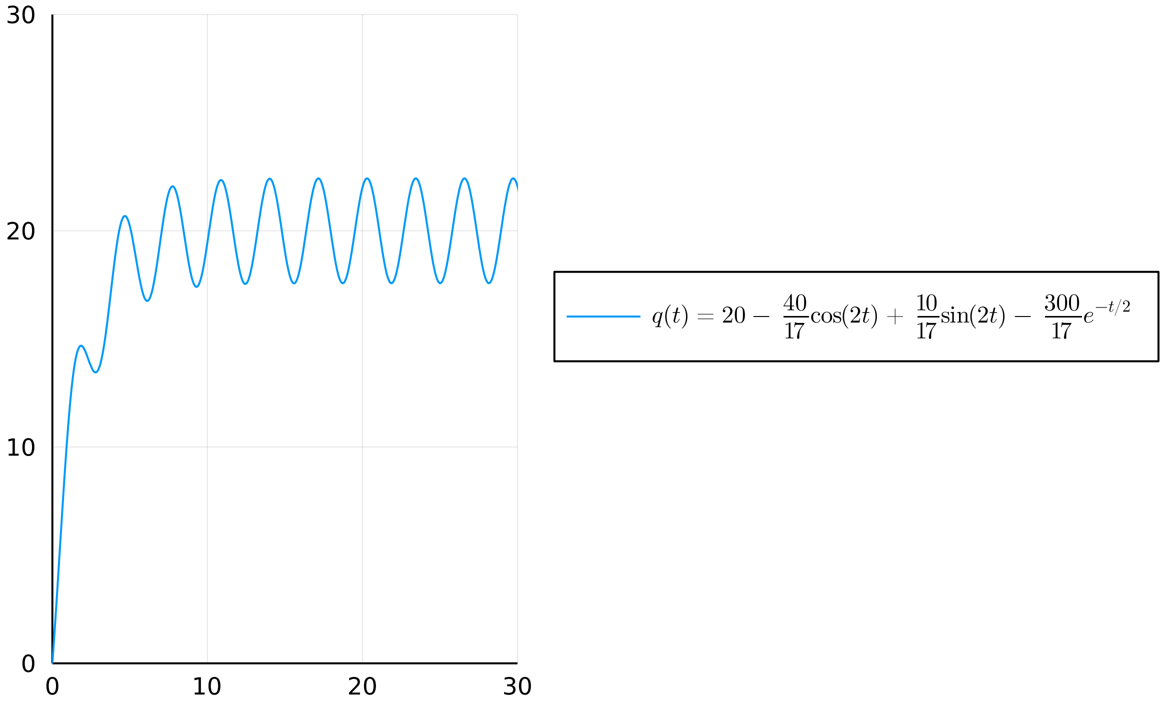
In [20]:
```julia
# To compute an indefinite integral
using SymPy

# we cal also use @vars x y z
s = symbols("s")

integrate(exp.(s/2)*(10+5*(sin.(2s))))
```

Out[20]:

$$\frac{10e^{\frac{s}{2}}\sin(2s)}{17} - \frac{40e^{\frac{s}{2}}\cos(2s)}{17} + 20e^{\frac{s}{2}}$$

In [28]:
```
# Solution of the initial value problem above

using Plots, LaTeXStrings, Plots.PlotMeasures
gr()

f(x) = 20 - (40/17)*(cos.(2x))+(10/17)*(sin.(2x))-(300/17)*(exp.(-x/2))

p = plot(f, 0,50, xlims=(0, 30), ylims=(0, 30),
    bottom_margin=10mm, label=L"q(t)=20 - \frac{40}{17} \cos(2t) + \frac{10}{17} \sin(2t) - \frac{300}{17}e
    legend=:outerright)
```

Out[28]:



Julia Codes for Calculating Differentiation and Integral with SymPy

In [11]:
```
# To compute an indefinite integral
using SymPy

# we cal also use @vars x y z
x = symbols("x")

# diff(cos(x), x)
# diff(exp(x**2), x)
# save for ** becoming ^ this is the same

integrate(3x^3 + 2x - 5)
```

```
┌ Info: Precompiling SymPy [24249f21-da20-56a4-8eb1-6a02cf4ae2e6]
└ @ Base loading.jl:1423
```

Out[11]: $\frac{3x^4}{4} + x^2 - 5x$

To compute a definite integral, pass the argument (integration_variable, lower_limit, upper_limit)

In [12]:
```
# To compute a definite integral

using SymPy
x = symbols("x")

integrate(exp(-x), (x, 0, oo))
```

Out[12]: 1

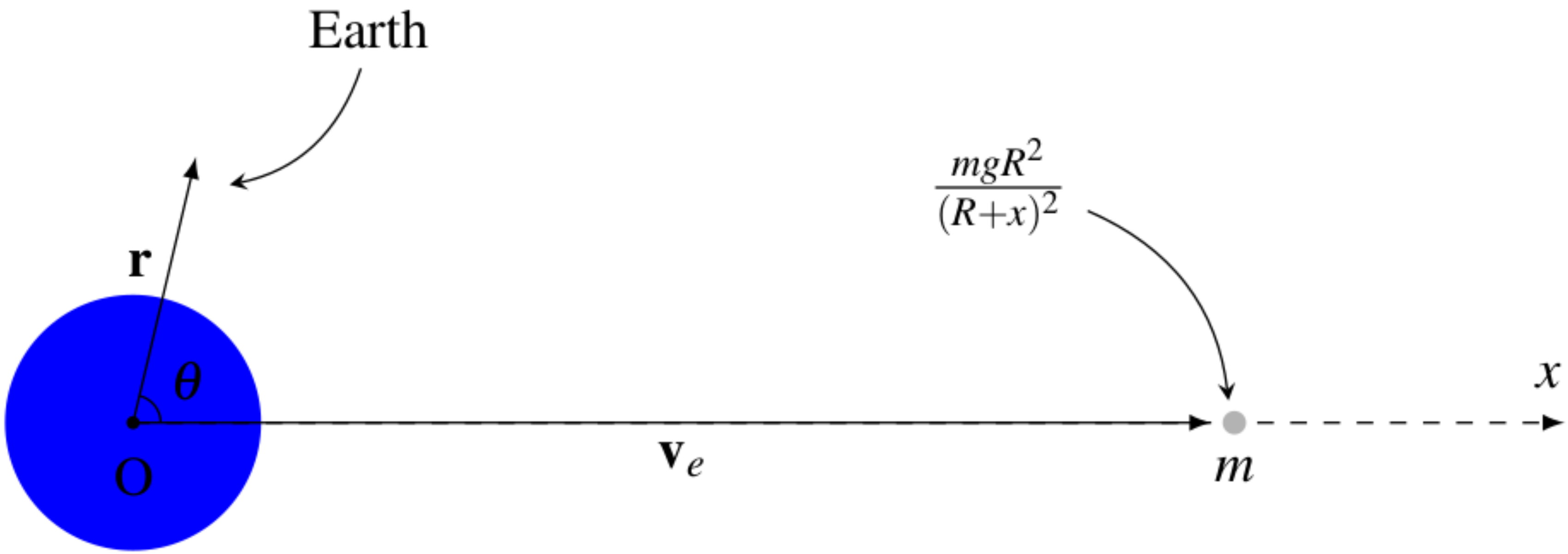
In [16]:
```
# To compute a definite multiple integral

using SymPy
x, y = symbols("x y")

integrate(exp(-x^2 - y^2), (x, -oo, oo), (y, -oo, oo))
```

Out[16]: π

In [ ]:
```
# To compute a differentiation
using SymPy

# we cal also use @vars x y z
x = symbols("x")


diff(cos(x), x)
# diff(exp(x**2), x)
```

Example: Escape Velocity

A body of constant mass m is projected away from the earth in a direction perpendicular to the earth's surface with an initial velocity v_0. Assuming that there is no air resistance, but taking into account the variation of the earth's gravitational field with distance, find an expression for the velocity during the ensuing motion.



Also find the initial velocity that is required to lift the body to a given maximum altitude ζ above the surface of the earth, and find the least initial velocity for which the body will not return to the earth; the latter is the escape velocity.

Solution

1)

- Let the positive x-axis point away from the center of the earth along the line of motion with $x = 0$ lying on the earth's surface.
- Gravity is directed toward the center of the earth, which is not necessarily downward from a perspective away from the earth's surface.
- The gravitational force acting on the body (weight) is inversely proportional to the square of the distance from the center of the earth and is given by $w(x) = -k/(x+R)^2$, where k is a constant, R is the radius of the earth, and the minus sign signifies that $w(x)$ is directed in the negative x direction.

2)

- On the earth's surface $w(0)$ is given by $-mg$, where g is the acceleration due to gravity at sea level. Therefore $k = mgR^2$ and

$$w(x) = -\frac{mgR^2}{(R+x)^2}$$

$$m\frac{dv}{dt} = -\frac{mgR^2}{(R+x)^2}$$

and the initial condition is

$v(0) = v_0$

3)

- Since there are many variables, we can eliminate variable t by thinking of x, rather than t, as the independent variable. We can express dv/dt in terms of dv/dx by using the chain rule

$$\frac{dv}{dt} = \frac{dv}{dx}\frac{dx}{dt} = v\frac{dv}{dx}$$

thus

$$m\frac{dv}{dt} = -\frac{mgR^2}{(R+x)^2}$$

$$v\frac{dv}{dx} = -\frac{gR^2}{(R+x)^2}$$

(by separating the variables)

$$v\,dv = -\frac{gR^2}{(R+x)^2}\,dx$$

(thus integrating)

$$\frac{v^2}{2} = \frac{gR^2}{R+x} + c$$

4)

The initial condition at $t = 0$ can be replaced by the condition that $v = v_0$ when $x = 0$. Hence $c = (v_0^2/2) - gR$ and

$$v = \pm\sqrt{v_0^2 - 2gR + \frac{2gR^2}{R+x}}$$

The equation above gives the velocity as a function of altitude rather than as a function of time.

- The plus sign must be chosen if the body is rising
- The minus sign must be chosen if the body is falling back to earth.

5) To determine the maximum altitude ζ that the body reaches, we set $v = 0$ and $x = \zeta$ then solve for ζ obtaining

$$\zeta = \frac{v_0^2 R}{2gR - v_0^2}$$

$$\zeta(2gR - v_0^2) = v_0^2 R$$

$$\zeta(2gR) = \zeta v_0^2 + v_0^2 R$$

$$\zeta(2gR) = v_0^2(\zeta + R)$$

In [3]:
```
# escape velocity in km/s
g = 9.8
R = 6371000

v = sqrt(2g*R)
```

Out[3]: 11174.596189572132

Example: Radiocarbon Dating

An important tool in archeological research is radiocarbon dating, developed by the American chemist Willard F. Libby.

This is a means of determining the age of certain wood and plant remains, hence of animals or human bones or artifacts found buried at the same levels.

Radiocarbon dating is based on the fact that some wood or plant remains contain residual amounts of carbon-14, a radioactive isotope of carbon. This isotope is accumulated during the lifetime of the plant and begins to decay at its death.

Since the half-life of carbon-14 is long (approximately 5730 years), measurable amounts of carbon-14 remain after many thousands of years. If even a tiny fraction of the original amount of carbon-14 is still present, then by appropriate laboratory measurements the proportion of the original amount of carbon-14 at time t and Q_0 is the original amount, then the ratio $Q(t)/Q_0$ can be determined, as long as this quantity is not too small.

Present measurement techniques permit the use of this method for time periods of 50,000 years or more.

a) Assuming that Q satisfies the differential equation $Q' = -rQ$, determine the decay constant r for carbon-14

b) Find an expression for $Q(t)$ at any time t, if $Q(0) = Q_0$

c) Suppose that certain remains are discovered in which the current residual amount of carbon-14 is 20% of the original amount. Determine the age of these remains.

Solution

a) First we are going to find the solution $Q(t)$ or the amount of carbon-14 at time t.

$$\begin{aligned}\frac{dQ}{dt} &= -rQ \\ \frac{dQ}{Q} &= -r\,dt \\ ln|Q| &= -rt + C \\ ln(Q) &= -rt + C \\ Q(t) &= ce^{-rt}\end{aligned}$$

then we are going to use the fact that the half-life of carbon-14 is approximately 5730 years, thus

$$\begin{aligned}\frac{Q(t)}{Q_0} &= ce^{-rt} \\ \text{(assuming } c = 1) \\ \frac{1}{2} &= e^{-5730r} \\ \ln\left|\frac{1}{2}\right| &= -5730r \\ r &= -0.00012097\end{aligned}$$

thus the decay constant r for carbon-14 is 0.00012097 yr^{-1}

b) $Q_t = Q_0 e^{-0.00012097t}$ with t in year.

c) We can use the formula $Q(t)$ to determine the age of these remains

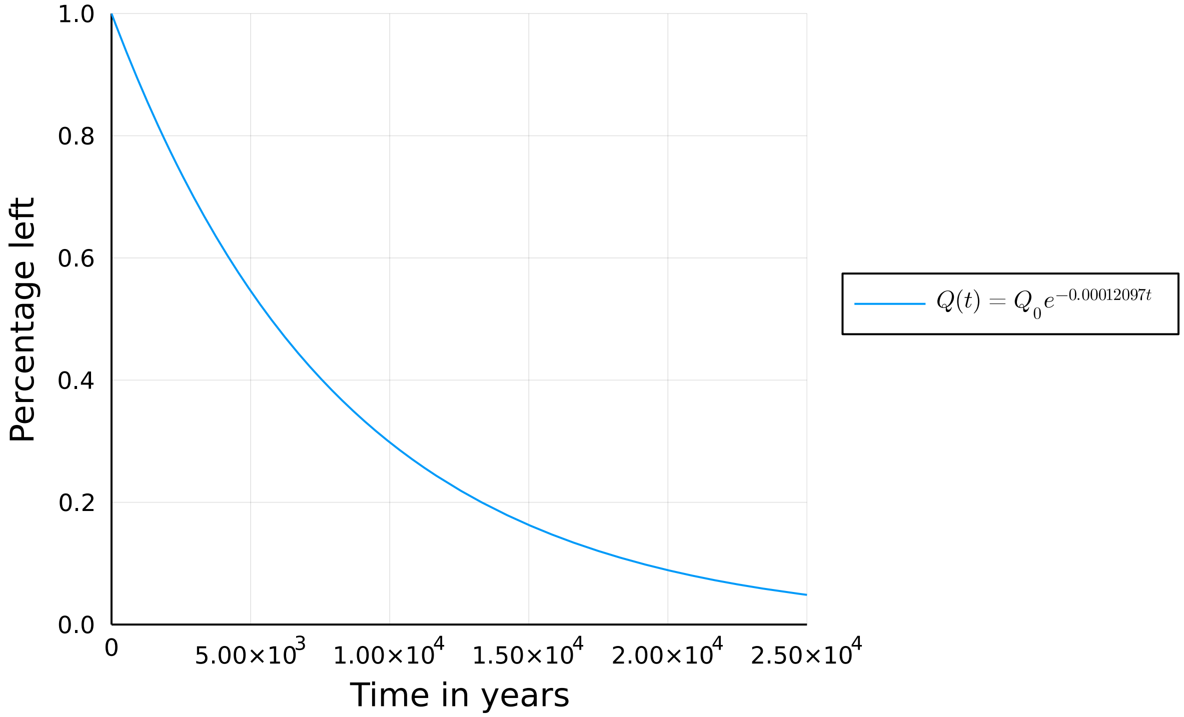
$$\frac{Q(t)}{Q_0} = e^{-0.00012097t}$$

$$\frac{20}{100} = e^{-0.00012097t}$$

$$ln\left|\frac{1}{5}\right| = -0.00012097t$$

In [39]:
```
# Solution of the initial value problem above

using Plots, LaTeXStrings, Plots.PlotMeasures
gr()

Qt = 20
Q0 = 100

f(x) = (exp.(-0.00012097x))

p = plot(f, 0,25000, xlims=(0, 25000), ylims=(0, 1),
    bottom_margin=10mm, label=L"Q(t)=Q_{0}e^{-0.00012097t}",
    xaxis = "Time in years", yaxis="Percentage left",
    framestyle=:zerolines,
    legend=:outerright)
```

Out[39]:



Example: Pollutant in a Lake

Consider a lake of constant volume V containing at time t an amount $Q(t)$ of pollutant, evenly distributed throughout the lake with a concentration $c(t)$, where $c(t) = Q(t)/V$.

Assume that water containing a concentraion k of pollutant enters the lake at a rate r, and that water leaves the lake at at the same rate.

Suppose that pollutants are also added directly to the lake at a constant rate P. Note that given assumptions neglect a number of factors that may, in some cases, be important-for example, the water added or lost by precipitation, absorption, and evaporation; the stratifying effect of temperature differences in a deep lake; the tendency of irregularities in the coastline to produce sheltered bays; and the fact that pollutants are not deposited evenly throughout the lake but (usually) at isolated points around its periphery.

The result below must be interpreted in the light of the neglect of such factors as these

a) If at time $t = 0$ the concentration of pollutant is c_0, find an expression for the concentration $c(t)$ at any time. What is the limiting concentration as $t \to \infty$

b) If the addition of pollutants to the lake is terminated ($k = 0$ and $P = 0$ for $t > 0$), determine the time interval T that must elapse before the concentration of pollutants is reduced to 50% of its original value; to 10% of its original value

c) The table below contains data for several of the Great Lakes. Using these data, determine from part b) the time T necessary to reduce the contamination of each of these lakes to 10% of the original value.

Lake	V (km$^3 \times 10^3$)	r (km^3/year)
Superior	12.2	65.2
Michigan	4.9	158
Erie	0.46	175
Ontario	1.6	209

Solution

a)

$c = k + (P/r) + [c_0 - k - (P/r)]e^{-rt/V}$

$\lim_{t \to \infty} c = k + (P/r)$ b)

In [ ]:
```
# naive definition:
Base.cbrt(z::Complex) = cbrt(abs(z)) * cis(angle(z)/3)

cbrt(-3+0im)
```

Differences Between Linear and Nonlinear Equations

Autonomous Equations and Population Dynamics

Second Order Linear Equations

In [ ]:
```
```

DifferentialEquations.jl Tutorial (October 23rd 2022)

This is a great package to solve differential equations, it can be used for your own needs, I just going to make a summary that can help readers besides me to use this package correctly.

If you are using Julia and activate the environment add this package first:

```
julia --project="."

julia> ]

st> add DifferentialEquations
```

About Differential Equations

In this package, we define an ordinary differential equation as an equation which describes the way that a variable u changes, that is

$$u' = f(u, p, t)$$

where p are the parameters of the model, t is the time variable, and f is the nonlinear model of how u changes. The initial value problem also includes the information about the starting value:

$$u(t_0) = u_0$$

If we know the starting value and know how the value will change with time, then we are able to know what the value will be at any time point in the future. This is the intuitive definition of a differential equation.

Example: Exponential Growth

Our first model will be the canonical exponential growth model. This model says that the rate of change is proportional to the current value, and could be represented with this differential equation:

$$u' = au$$

where we have a starting value $u(0) = u_0$

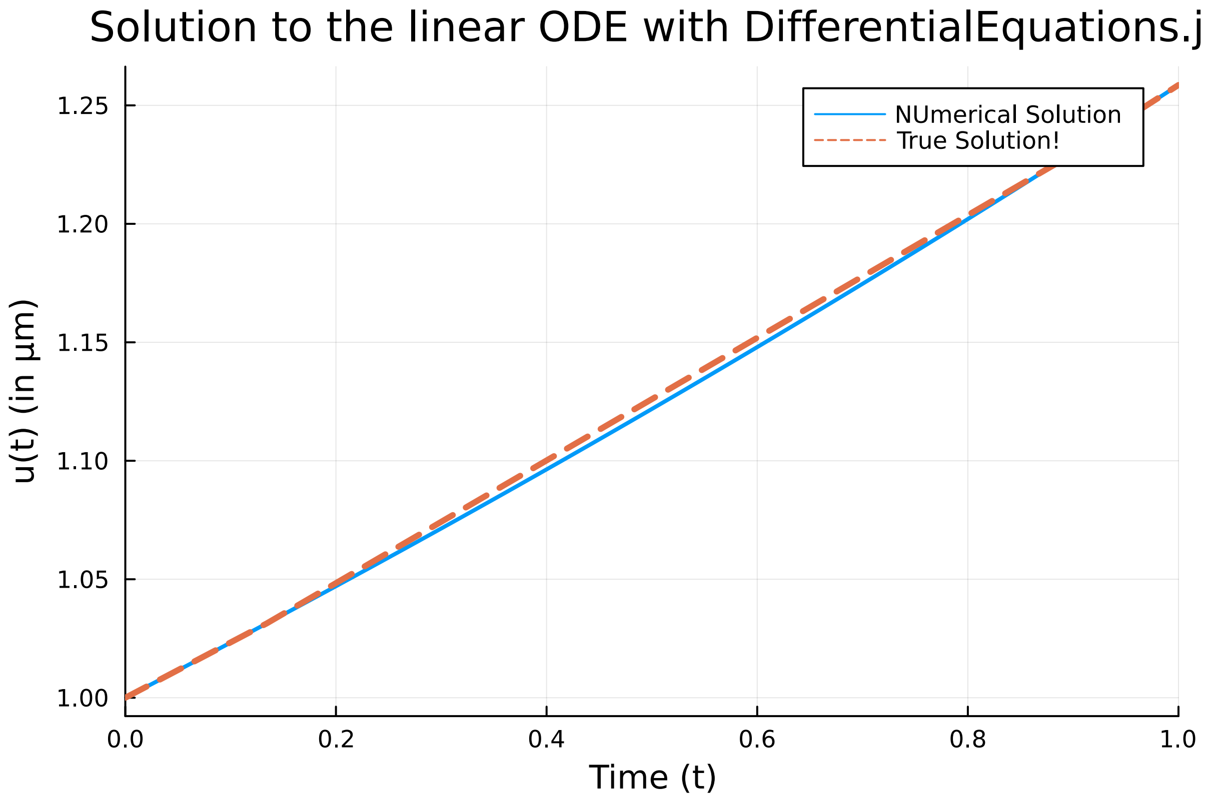
Let's say we put 1 USD to buy 1 GBP (at parity) which is increasing at a rate of 23 per year, the initial time now is $t = 0$, and t is in years, our model is

$$u' = 0.23u$$

and $u(0) = 1$, if we want to solve this model on a time span from $t = 0.0$ to $t = 1.0$, then we define an ODEProblem by specifying this function f, this

In [17]:
```
using DifferentialEquations
using Plots; gr()

f(u,p,t) = 0.23u
u0 = 1.0
tspan = (0.0,1.0)
prob = ODEProblem(f,u0,tspan)
sol = solve(prob)

plot(sol, linewidth=2,
     title="Solution to the linear ODE with DifferentialEquations.jl",
     xaxis="Time (t)", yaxis="u(t) (in μm)",
     label="Numerical Solution")

plot!(sol.t, t->1.0*exp(0.23t), lw=3, ls=:dash, label="True Solution!")
```

Out[17]:



Example: Radioactive Decay of Carbon-14

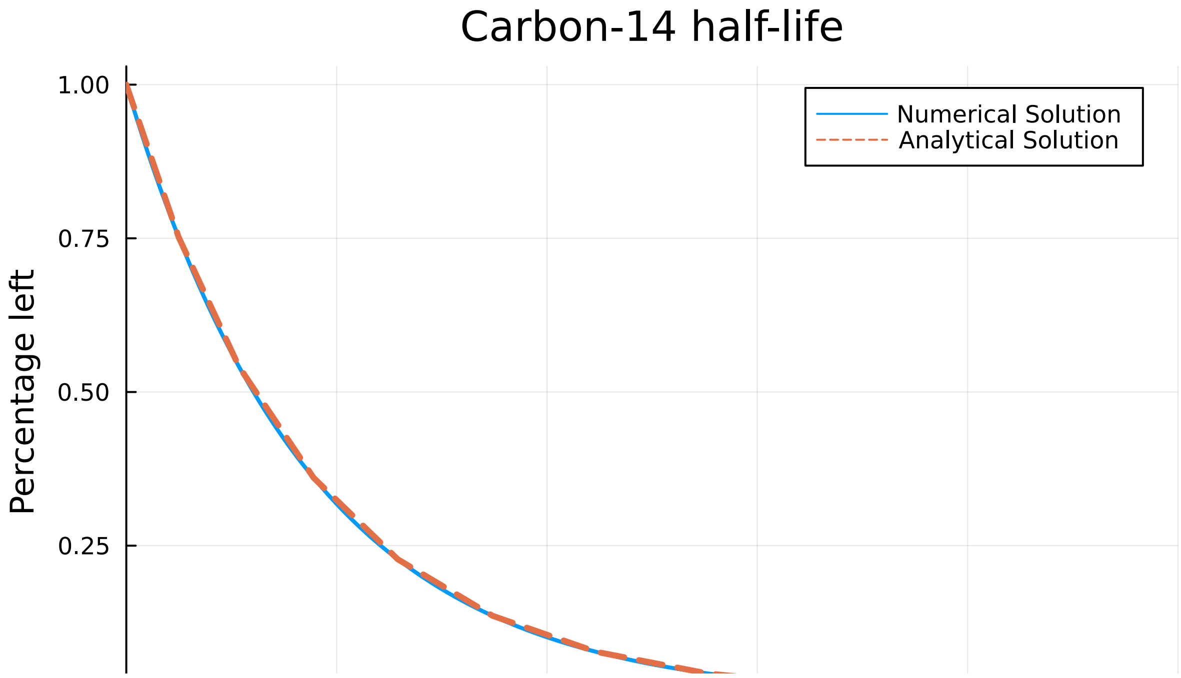
The Radioactive decay problem is the first order linear ODE problem of an exponential with a negative coefficient, which represents the half-life of the process in question. Should the coefficient be positive, this would represent a population growth equation.

This equation is represented by:

$f(t, u) = \frac{du}{dt}$

In [7]:
```
using OrdinaryDiffEq, Plots
gr()

#Half-life of Carbon-14 is 5,730 years.
C₁ = 5.730

#Setup
u₀ = 1.0
tspan = (0.0, 1.0)

#Define the problem
radioactivedecay(u,p,t) = -C₁*u

#Pass to solver
prob = ODEProblem(radioactivedecay,u₀,tspan)
sol = solve(prob,Tsit5())

#Plot
plot(sol,linewidth=2,title ="Carbon-14 half-life", xaxis = "Time in thousands of years", yaxis = "Percentage
plot!(sol.t, t->exp(-C₁*t),lw=3,ls=:dash,label="Analytical Solution")
```

Out[7]:



Second Order Linear ODE

Example: Simple Harmonic Oscillator

The differential equation for the harmonic oscillator is given by

$$\ddot{x} + \omega^2 x = 0$$

with the known analytical solution:

$$x(t) = A\cos(\omega t - \phi) v(t) = -A\omega\sin(\omega t - \phi)$$

where

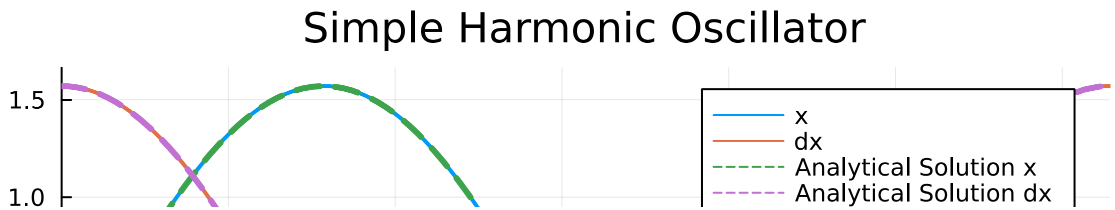
$$A = \sqrt{c_1 + c_2} \quad \text{and} \quad \tan\phi = \frac{c_2}{c_1}$$

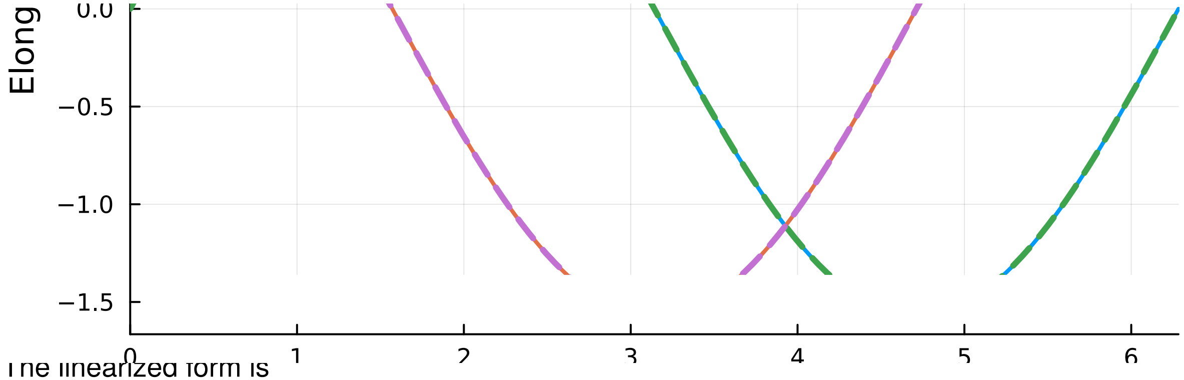
with c_1, c_2 constants determined by the initial conditions such that c_1 is the initial position, and ωc_2 is the initial velocity.

In [8]:
```
# Simple Harmonic Oscillator Problem
# The order of the variables (and initial conditions) is dx, x.
# Thus, if we want the first series to be x, we have to flip the order with vars=[2,1].
using OrdinaryDiffEq, Plots

#Parameters
ω = 1

#Initial Conditions
x₀ = [0.0]
dx₀ = [π/2]
tspan = (0.0, 2π)

ϕ = atan((dx₀[1]/ω)/x₀[1])
A = √(x₀[1]^2 + dx₀[1]^2)

#Define the problem
function harmonicoscillator(ddu,du,u,ω,t)
    ddu .= -ω^2 * u
end

#Pass to solvers
prob = SecondOrderODEProblem(harmonicoscillator, dx₀, x₀, tspan, ω)
sol = solve(prob, DPRKN6())

#Plot
plot(sol, vars=[2,1], linewidth=2, title ="Simple Harmonic Oscillator", xaxis = "Time", yaxis = "Elongation'
plot!(t->A*cos(ω*t-ϕ), lw=3, ls=:dash, label="Analytical Solution x")
plot!(t->-A*ω*sin(ω*t-ϕ), lw=3, ls=:dash, label="Analytical Solution dx")
```

```
┌ Warning: To maintain consistency with solution indexing, keyword argument vars will be removed in a futu
re version. Please use keyword argument idxs instead.
│   caller = ip:0x0
└ @ Core :-1
```

Out[8]:





proximation we will get an elliptic integral which

The linearized form is

$$\ddot{\theta} + \frac{g}{L}\theta = 0$$

and the real simple pendulum problem form is

$$\ddot{\theta} + \frac{g}{L}\sin(\theta) = 0$$

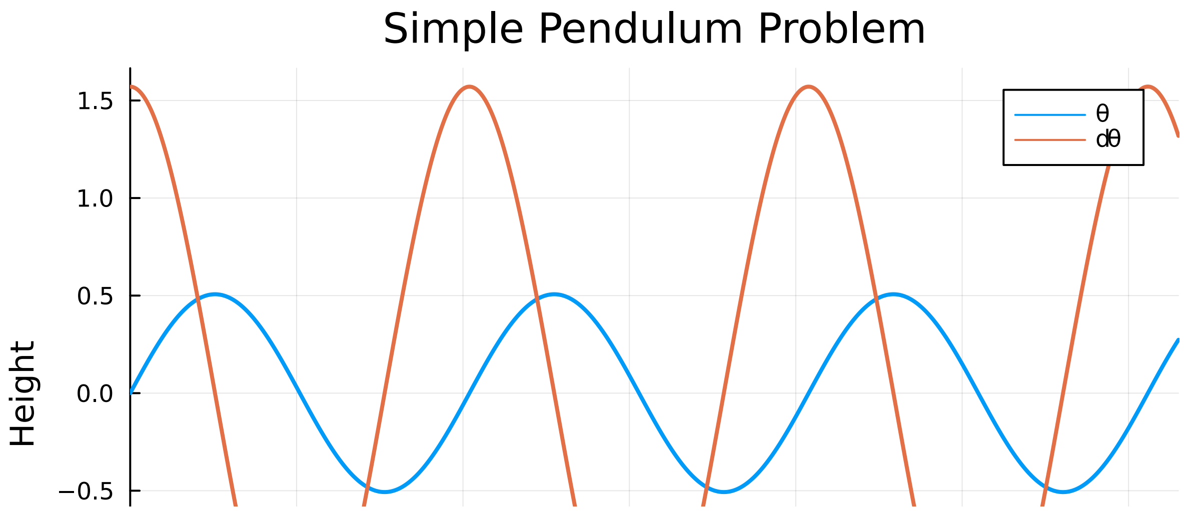
The double dots on θ means that it is the second derivative of $theta$ / second order ODE. Now, transform the problem above into the first order ODE by employing the notation $d\theta = \dot{\theta}$

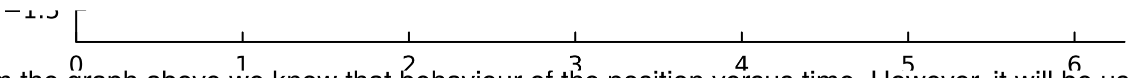
$$\dot{\theta} = d\theta$$

$$d\dot{\theta} = -\frac{g}{L}\sin(\theta)$$

In [9]:
```
# Simple Pendulum Problem
using OrdinaryDiffEq, Plots

#Constants
const g = 9.81
L = 1.0

#Initial Conditions
u₀ = [0,π/2]
tspan = (0.0,6.3)

#Define the problem
function simplependulum(du,u,p,t)
    θ = u[1]
    dθ = u[2]
    du[1] = dθ
    du[2] = -(g/L)*sin(θ)
end

#Pass to solvers
prob = ODEProblem(simplependulum, u₀, tspan)
sol = solve(prob,Tsit5())

#Plot
plot(sol,linewidth=2,title ="Simple Pendulum Problem", xaxis = "Time", yaxis = "Height", label = ["\\theta"
```

Out[9]:



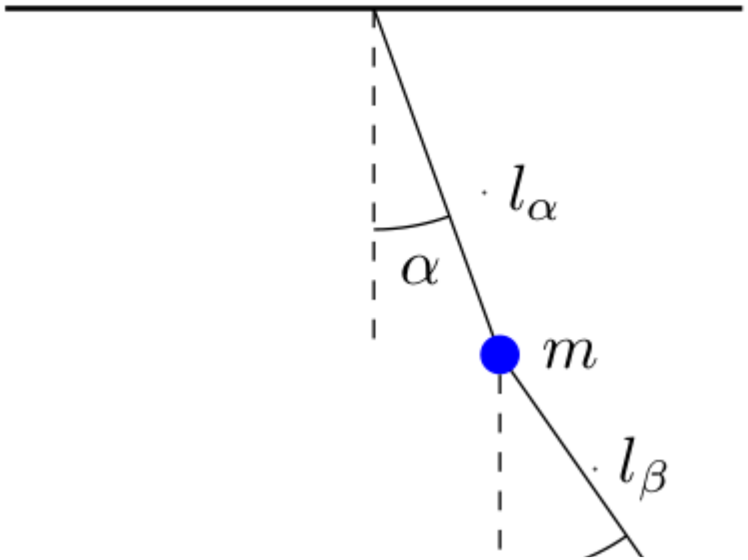


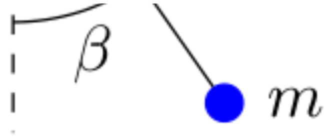
From the graph above we know that behaviour of the position versus time. However, it will be useful for us to look at the phase space of the pendulum, i.e., and representation of all possible states of the system in question (the pendulum) by looking at its velocity and position. Phase space analysis is ubiquitous in the analysis of dynamical systems, and thus we will provide a few facilities for it.

In [ ]:
```
p = plot(sol,vars = (1,2), xlims = (-9,9), title = "Phase Space Plot", xaxis = "Velocity", yaxis = "Positio
function phase_plot(prob, u0, p, tspan=2pi)
    _prob = ODEProblem(prob.f,u0,(0.0,tspan))
    sol = solve(_prob,Vern9()) # Use Vern9 solver for higher accuracy
    plot!(p,sol,vars = (1,2), xlims = nothing, ylims = nothing)
end
for i in -4pi:pi/2:4π
    for j in -4pi:pi/2:4π
        phase_plot(prob, [j,i], p)
    end
end
plot(p,xlims = (-9,9))
```

Example: Double Pendulum

This is an example if we have two pendulums, the second pendulum depend / tied to the first pendulum





Derivation of the Equations of Motion

The derivation of the double pendulum equations of motion using the Lagrangian formulation has become a standard exercise in introductory classical mechanics.

The two degrees of freedom are taken to be θ_α and θ_β, the angle of each pendulum rod from the vertical. The components of the bob positions and velocities are

$$
\begin{aligned}
x_\alpha &= l_\alpha \sin\theta_\alpha & \dot{x}_\alpha &= l_\alpha \dot{\theta}_\alpha \cos\theta_\alpha \\
y_\alpha &= -l_\alpha \cos\theta_\alpha & \dot{y}_\alpha &= l_\alpha \dot{\theta}_\alpha \sin\theta_\alpha \\
x_\beta &= l_\alpha \sin\theta_\alpha + l_\beta \sin\theta_\beta & \dot{x}_\beta &= l_\alpha \dot{\theta}_\alpha \cos\theta_\alpha + l_\beta \dot{\theta}_\beta \cos\theta_\beta \\
y_\beta &= -l_\alpha \cos\theta_\alpha - l_\beta \cos\theta_\beta & \dot{y}_\beta &= l_\alpha \dot{\theta}_\alpha \sin\theta_\alpha + l_\beta \dot{\theta}_\beta \sin\theta_\beta
\end{aligned}
$$

The potential and kinetic energies are then:

$$
\begin{aligned}
V &= m_\alpha g y_\alpha + m_\beta g y_\beta = -(m_\alpha + m_\beta) l_\alpha g \cos\theta_\alpha - m_\beta l_\beta g \cos\theta_\beta \\
T &= \frac{1}{2} m_\alpha v_\alpha^2 + \frac{1}{2} m_\beta v_\beta^2 = \frac{1}{2} m_\alpha (\dot{x}_\alpha^2 + \dot{y}_\alpha^2) + \frac{1}{2} m_\beta (\dot{x}_\beta^2 + \dot{y}_\beta^2) \\
&= \frac{1}{2} m_\alpha l_\alpha^2 \dot{\theta}_\alpha^2 + \frac{1}{2} m_\beta [l_\alpha^2 \dot{\theta}_\alpha^2 + l_\beta^2 \dot{\theta}_\beta^2 + 2 l_\alpha l_\beta \dot{\theta}_\alpha \dot{\theta}_\beta \cos(\theta_\alpha - \theta_\beta)]
\end{aligned}
$$

The Lagrangian, $\mathcal{L} = T - V$ is therefore

$$
\mathcal{L} = \frac{1}{2}(m_\alpha + m_\beta) l_\alpha^2 \dot{\theta}_\alpha^2 + \frac{1}{2} m_\beta l_\beta^2 \dot{\theta}_\beta^2 + m_\beta l_\alpha l_\beta \dot{\theta}_\alpha \dot{\theta}_\beta \cos(\theta_\alpha - \theta_\beta) + (m_\alpha + m_\beta) l_\alpha g \cos\theta_\alpha + m_\beta g l_\beta \cos\theta_\beta
$$

The Euler-Lagrange equations are:

$$\frac{d}{dt}\left(\frac{\partial \mathcal{L}}{\partial \dot{q}_i}\right) - \frac{\partial \mathcal{L}}{\partial q_i} = 0 \quad \text{for } q_i = \theta_\alpha, \theta_\beta$$

For these coordinates, after some calculus and algebra, we get:

$$(m_\alpha + m_\beta) l_\alpha \ddot{\theta}_\alpha + m_\beta l_\beta \ddot{\theta}_\beta \cos(\theta_\alpha - \theta_\beta) + m_\beta l_\beta \dot{\theta}_\beta^2 sin(\theta_\alpha - \theta_\beta) + (m_\alpha + m_\beta) g \sin\theta_\alpha = 0$$
$$m_\beta l_\beta \ddot{\theta}_\beta + m_\beta l_\alpha \ddot{\theta}_\alpha \cos(\theta_\alpha - \theta_\beta) - m_\beta l_\alpha \dot{\theta}_\alpha^2 \sin(\theta_\alpha - \theta_\beta) + m_\beta g \sin\theta_\beta = 0$$

The differential equation to model the Double Pendulum trouble is

$$\frac{d}{dt}\begin{pmatrix} \alpha \\ l_\alpha \\ \beta \\ l_\beta \end{pmatrix} = \begin{pmatrix} 2\frac{l_\alpha - (1+\cos\beta) l_\beta}{3-\cos\ 2\beta} \\ -2\sin\alpha - \sin(\alpha+\beta) \\ 2\frac{-(1+\cos\beta) l_\alpha + (3+2\cos\beta) l_\beta}{3-\cos\ 2\beta} \\ -\sin(\alpha+\beta) - 2\sin(\beta)\frac{(l_\alpha - l_\beta) l_\beta}{3-\cos\ 2\beta} + 2\sin(2\beta)\frac{l_\alpha^2 - 2(1+\cos\ \beta) l_\alpha l_\beta + (3+2\cos\beta) l_\beta^2}{(3-\cos\ 2\beta)^2} \end{pmatrix}$$

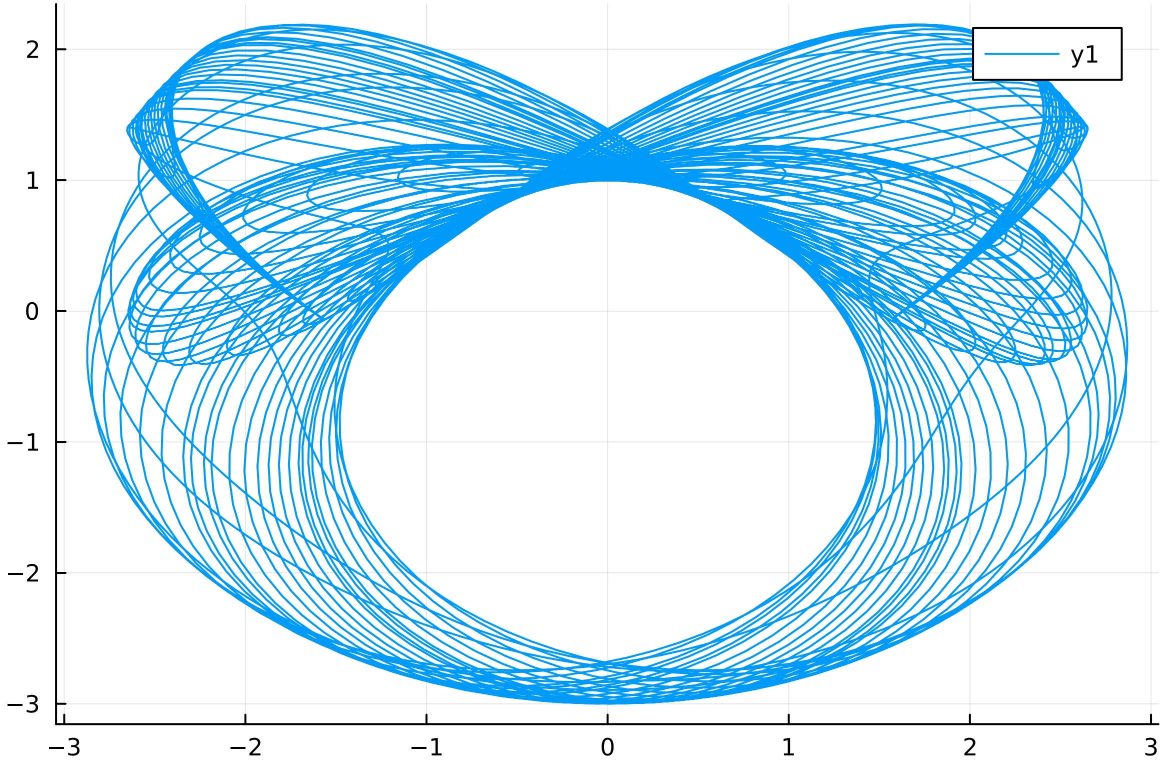
In [4]:
```
#Double Pendulum Problem
using OrdinaryDiffEq, Plots

#Constants and setup
const g = 9.81
const m₁, m₂, L₁, L₂ = 1, 2, 1, 2
initial = [0, π/3, 0, 3pi/5]
tspan = (0.,50.)

#Convenience function for transforming from polar to Cartesian coordinates
function polar2cart(sol;dt=0.02,l1=L₁,l2=L₂,vars=(2,4))
    u = sol.t[1]:dt:sol.t[end]

    p1 = l1*map(x->x[vars[1]], sol.(u))
    p2 = l2*map(y->y[vars[2]], sol.(u))

    x1 = l1*sin.(p1)
    y1 = l1*-cos.(p1)
    (u, (x1 + l2*sin.(p2),
     y1 - l2*cos.(p2)))
end

#Define the Problem
function double_pendulum(xdot,x,p,t)
    xdot[1]=x[2]
    xdot[2]=-((g*(2*m₁+m₂)*sin(x[1])+m₂*(g*sin(x[1]-2*x[3])+2*(L₂*x[4]^2+L₁*x[2]^2*cos(x[1]-x[3]))*sin(x[1]-
    xdot[3]=x[4]
    xdot[4]=(((m₁+m₂)*(L₁*x[2]^2+g*cos(x[1]))+L₂*m₂*x[4]^2*cos(x[1]-x[3]))*sin(x[1]-x[3]))/(L₂*(m₁+m₂-m₂*cos
end

#Pass to Solvers
double_pendulum_problem = ODEProblem(double_pendulum, initial, tspan)
sol = solve(double_pendulum_problem, Vern7(), abs_tol=1e-10, dt=0.05);
```

```
Unrecognized keyword arguments: [:abs_tol]


┌ Warning: Unrecognized keyword arguments found. Future versions will error.
│ The only allowed keyword arguments to `solve` are:
│ (:dense, :saveat, :save_idxs, :tstops, :tspan, :d_discontinuities, :save_everystep, :save_on, :save_star
```

```
t, :save_end, :initialize_save, :adaptive, :abstol, :reltol, :dt, :dtmax, :dtmin, :force_dtmin, :internaln
orm, :controller, :gamma, :beta1, :beta2, :qmax, :qmin, :qsteady_min, :qsteady_max, :qoldinit, :failfacto
r, :calck, :alias_u0, :maxiters, :callback, :isoutofdomain, :unstable_check, :verbose, :merge_callbacks, :
progress, :progress_steps, :progress_name, :progress_message, :timeseries_errors, :dense_errors, :weak_tim
eseries_errors, :weak_dense_errors, :wrap, :calculate_error, :initializealg, :alg, :save_noise, :delta, :s
eed, :alg_hints, :kwargshandle, :trajectories, :batch_size, :sensealg, :advance_to_tstop, :stop_at_next_ts
top, :default_set, :second_time, :prob_choice, :alias_jump, :alias_noise)
│
│ See https://diffeq.sciml.ai/stable/basics/common_solver_opts/ (https://diffeq.sciml.ai/stable/basics/com
mon_solver_opts/) for more details.
│
│ Set kwargshandle=KeywordArgError for an error message.
│ Set kwargshandle=KeywordArgSilent to ignore this message.
└ @ DiffEqBase /home/browni/.julia/packages/DiffEqBase/Lq1gG/src/solve.jl:868
```

In [5]:
```
#Obtain coordinates in Cartesian Geometry
ts, ps = polar2cart(sol, l1=L₁, l2=L₂, dt=0.01)
plot(ps...)
```

Out[5]:



In [1]:
```
# The two codes below work if you run it in REPL using Julia 1.7.3
# Save each as a julia files(pendulum.jl and pendulumplot.jl) and run the first and the second

using OrdinaryDiffEq

G = 9.8  # acceleration due to gravity, in m/s^2
L1 = 1.0  # length of pendulum 1 in m
L2 = 1.0  # length of pendulum 2 in m
L = L1 + L2  # maximal length of the combined pendulum
M1 = 1.0  # mass of pendulum 1 in kg
M2 = 1.0  # mass of pendulum 2 in kg
t_stop = 5  # how many seconds to simulate
history_len = 500  # how many trajectory points to display

function pendulum!(du, u, p, t)
    (; M1, M2, L1, L2, G) = p

    du[1] = u[2]

    delta = u[3] - u[1]
    den1 = (M1 + M2) * L1 - M2 * L1 * cos(delta) * cos(delta)
    du[2] = (
        (
            M2 * L1 * u[2] * u[2] * sin(delta) * cos(delta) +
            M2 * G * sin(u[3]) * cos(delta) +
            M2 * L2 * u[4] * u[4] * sin(delta) - (M1 + M2) * G * sin(u[1])
        ) / den1
    )

    du[3] = u[4]

    den2 = (L2 / L1) * den1
    du[4] = (
        (
            -M2 * L2 * u[4] * u[4] * sin(delta) * cos(delta) +
            (M1 + M2) * G * sin(u[1]) * cos(delta) -
            (M1 + M2) * L1 * u[2] * u[2] * sin(delta) - (M1 + M2) * G * sin(u[3])
        ) / den2
    )
    return nothing
```

```
end
```

```
┌ Info: Precompiling OrdinaryDiffEq [1dea7af3-3e70-54e6-95c3-0bf5283fa5ed]
└ @ Base loading.jl:1423
ERROR: LoadError: TypeError: in CartesianVIndex, in T, expected T<:Tuple{Vararg{Union{VectorizationBase.NullStep, Integer}, N}}, got Type{NTuple{4, Static.StaticInt{1}}}
Stacktrace:
 [1] _precompile_()
   @ LoopVectorization ~/.julia/packages/LoopVectorization/kVenK/src/precompile.jl:52
 [2] top-level scope
   @ ~/.julia/packages/LoopVectorization/kVenK/src/LoopVectorization.jl:116
 [3] include
   @ ./Base.jl:418 [inlined]
 [4] include_package_for_output(pkg::Base.PkgId, input::String, depot_path::Vector{String}, dl_load_path::Vector{String}, load_path::Vector{String}, concrete_deps::Vector{Pair{Base.PkgId, UInt64}}, source::String)
   @ Base ./loading.jl:1318
 [5] top-level scope
   @ none:1
 [6] eval
   @ ./boot.jl:373 [inlined]
```

th1 and th2 are the initial angles (degrees)

w10 and w20 are the initial angular velocities (degrees per second)

In [2]:
```
th1 = 120.0
w1 = 0.0
th2 = -10.0
w2 = 0.0

p = (; M1, M2, L1, L2, G)
prob = ODEProblem(pendulum!, deg2rad.([th1, w1, th2, w2]), (0.0, t_stop), p)
sol = solve(prob, Tsit5())



x1 = +L1 * sin.(sol[1, :])
y1 = -L1 * cos.(sol[1, :])

x2 = +L2 * sin.(sol[3, :]) + x1
y2 = -L2 * cos.(sol[3, :]) + y1



using Plots
anim = @animate for i in eachindex(x2)

    x = [0, x1[i], x2[i]]
    y = [0, y1[i], y2[i]]

    plot(x, y, legend = false)
    plot!(xlims = (-2, 2), xticks = -2:0.5:2)
    plot!(ylims = (-2, 1), yticks = -2:0.5:1)
    scatter!(x, y)

    x = x2[1:i]
    y = y2[1:i]

    plot!(x, y, linecolor = :orange)
    plot!(xlims = (-2, 2), xticks = -2:0.5:2)
    plot!(ylims = (-2, 1), yticks = -2:0.5:1)
    scatter!(
        x,
        y,
        color = :orange,
```

```
        markersize = 2,
        markerstrokewidth = 0,
        markerstrokecolor = :orange,
    )
    annotate!(-1.25, 0.5, "time= $(rpad(round(sol.t[i]; digits=2),4,"0")) s")
end
gif(anim, fps = 12)
```

```
UndefVarError: M1 not defined

Stacktrace:
 [1] top-level scope
   @ In[2]:6
 [2] eval
   @ ./boot.jl:373 [inlined]
 [3] include_string(mapexpr::typeof(REPL.softscope), mod::Module, code::String, filename::String)
   @ Base ./loading.jl:1196
```

Example: Poincaré section

In this case the phase space is 4 dimensional and it cannot be easily visualized. Instead of looking at the full phase space, we can look at Poincaré sections, which are sections through a higher-dimensional phase space diagram. This helps to understand the dynamics of interactions and is wonderfully pretty.

The Poincaré section in this is given by the collection of (β, l_β) when $\alpha = 0$ and $\frac{d\alpha}{dt} > 0$

In [6]:
```
#Constants and setup
using OrdinaryDiffEq
initial2 = [0.01, 0.005, 0.01, 0.01]
tspan2 = (0.,500.)

#Define the problem
function double_pendulum_hamiltonian(udot,u,p,t)
    α  = u[1]
    lα = u[2]
    β  = u[3]
    lβ = u[4]
    udot .=
    [2(lα-(1+cos(β))lβ)/(3-cos(2β)),
    -2sin(α) - sin(α+β),
    2(-(1+cos(β))lα + (3+2cos(β))lβ)/(3-cos(2β)),
    -sin(α+β) - 2sin(β)*(((lα-lβ)lβ)/(3-cos(2β))) + 2sin(2β)*((lα^2 - 2(1+cos(β))lα*lβ + (3+2cos(β))lβ^2)/(3
end

# Construct a ContinousCallback
condition(u,t,integrator) = u[1]
affect!(integrator) = nothing
cb = ContinuousCallback(condition,affect!,nothing,
                        save_positions = (true,false))

# Construct Problem
poincare = ODEProblem(double_pendulum_hamiltonian, initial2, tspan2)
sol2 = solve(poincare, Vern9(), save_everystep = false, save_start=false, save_end=false, callback=cb, absto

function poincare_map(prob, u₀, p; callback=cb)
    _prob = ODEProblem(prob.f, u₀, prob.tspan)
    sol = solve(_prob, Vern9(), save_everystep = false, save_start=false, save_end=false, callback=cb, absto
    scatter!(p, sol, vars=(3,4), markersize = 3, msw=0)
end
```

Out[6]:
```
poincare_map (generic function with 1 method)
```

In [ ]:
```
lβrange = -0.02:0.0025:0.02
p = scatter(sol2, vars=(3,4), leg=false, markersize = 3, msw=0)
for lβ in lβrange
    poincare_map(poincare, [0.01, 0.01, 0.01, lβ], p)
end
plot(p, xlabel="\\beta", ylabel="l_\\beta", ylims=(0, 0.03))
```

Example: Hénon-Heiles System

The Hénon-Heiles potential occurs when non-linear motion of a star around a galactic center with the motion restricted to a plane.

$$\frac{d^2x}{dt^2} = -\frac{\partial V}{\partial x}$$

$$\frac{d^2y}{dt^2} = -\frac{\partial V}{\partial y}$$

where

$$V(x,y) = \frac{1}{2}(x^2 + y^2) + \lambda\left(x^2y - \frac{y^3}{3}\right)$$

We pick $\lambda = 1$ in this case, thus

$$V(x,y) = \frac{1}{2}\left(x^2 + y^2 + 2x^2y - \frac{2y^3}{3}\right)$$

Then the total energy of the system can be expressed by

$$E = T + V = V(x,y) + \frac{1}{2}(\dot{x}^2 + \dot{y}^2)$$

The total energy should conserve as this system evolves.

In [7]:
```
using OrdinaryDiffEq, Plots

#Setup
initial = [0.,0.1,0.5,0]
tspan = (0,100.)

#Remember, V is the potential of the system and T is the Total Kinetic Energy, thus E will
#the total energy of the system.
V(x,y) = 1//2 * (x^2 + y^2 + 2x^2*y - 2//3 * y^3)
E(x,y,dx,dy) = V(x,y) + 1//2 * (dx^2 + dy^2);

#Define the function
function Hénon_Heiles(du,u,p,t)
    x  = u[1]
    y  = u[2]
    dx = u[3]
    dy = u[4]
    du[1] = dx
    du[2] = dy
    du[3] = -x - 2x*y
    du[4] = y^2 - y -x^2
end

#Pass to solvers
prob = ODEProblem(Hénon_Heiles, initial, tspan)
sol = solve(prob, Vern9(), abs_tol=1e-16, rel_tol=1e-16);
```

```
Unrecognized keyword arguments: [:abs_tol, :rel_tol]


┌ Warning: Unrecognized keyword arguments found. Future versions will error.
│ The only allowed keyword arguments to `solve` are:
│ (:dense, :saveat, :save_idxs, :tstops, :tspan, :d_discontinuities, :save_everystep, :save_on, :save_star
t, :save_end, :initialize_save, :adaptive, :abstol, :reltol, :dt, :dtmax, :dtmin, :force_dtmin, :internaln
orm, :controller, :gamma, :beta1, :beta2, :qmax, :qmin, :qsteady_min, :qsteady_max, :qoldinit, :failfacto
r, :calck, :alias_u0, :maxiters, :callback, :isoutofdomain, :unstable_check, :verbose, :merge_callbacks, :
progress, :progress_steps, :progress_name, :progress_message, :timeseries_errors, :dense_errors, :weak_tim
eseries_errors, :weak_dense_errors, :wrap, :calculate_error, :initializealg, :alg, :save_noise, :delta, :s
eed, :alg_hints, :kwargshandle, :trajectories, :batch_size, :sensealg, :advance_to_tstop, :stop_at_next_ts
```

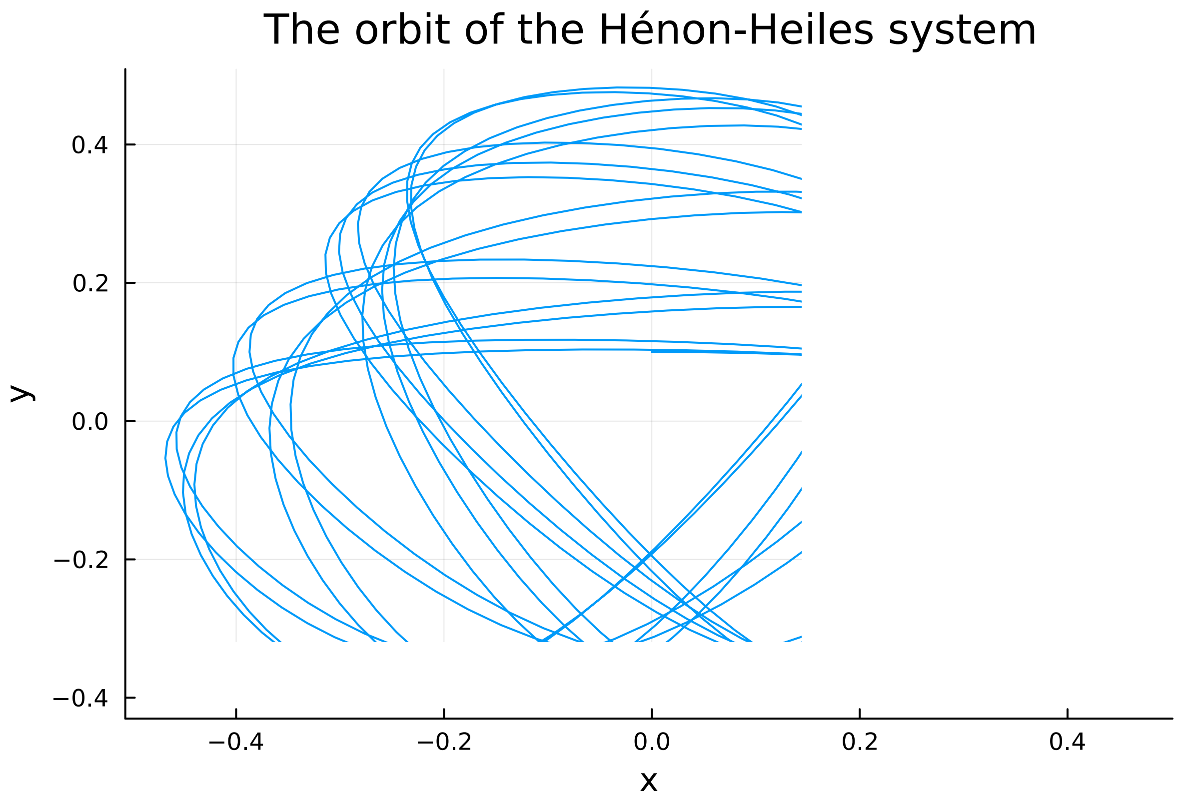
```
top, :default_set, :second_time, :prob_choice, :alias_jump, :alias_noise)
│
│ See https://diffeq.sciml.ai/stable/basics/common_solver_opts/ (https://diffeq.sciml.ai/stable/basics/common_solver_opts/) for more details.
│
│ Set kwargshandle=KeywordArgError for an error message.
│ Set kwargshandle=KeywordArgSilent to ignore this message.
└ @ DiffEqBase /home/browni/.julia/packages/DiffEqBase/Lq1gG/src/solve.jl:868
```

In [8]:

```julia
# Plot the orbit
plot(sol, vars=(1,2), title = "The orbit of the Hénon-Heiles system", xaxis = "x", yaxis = "y", leg=false)
```

```
┌ Warning: To maintain consistency with solution indexing, keyword argument vars will be removed in a future version. Please use keyword argument idxs instead.
│   caller = ip:0x0
└ @ Core :-1
```

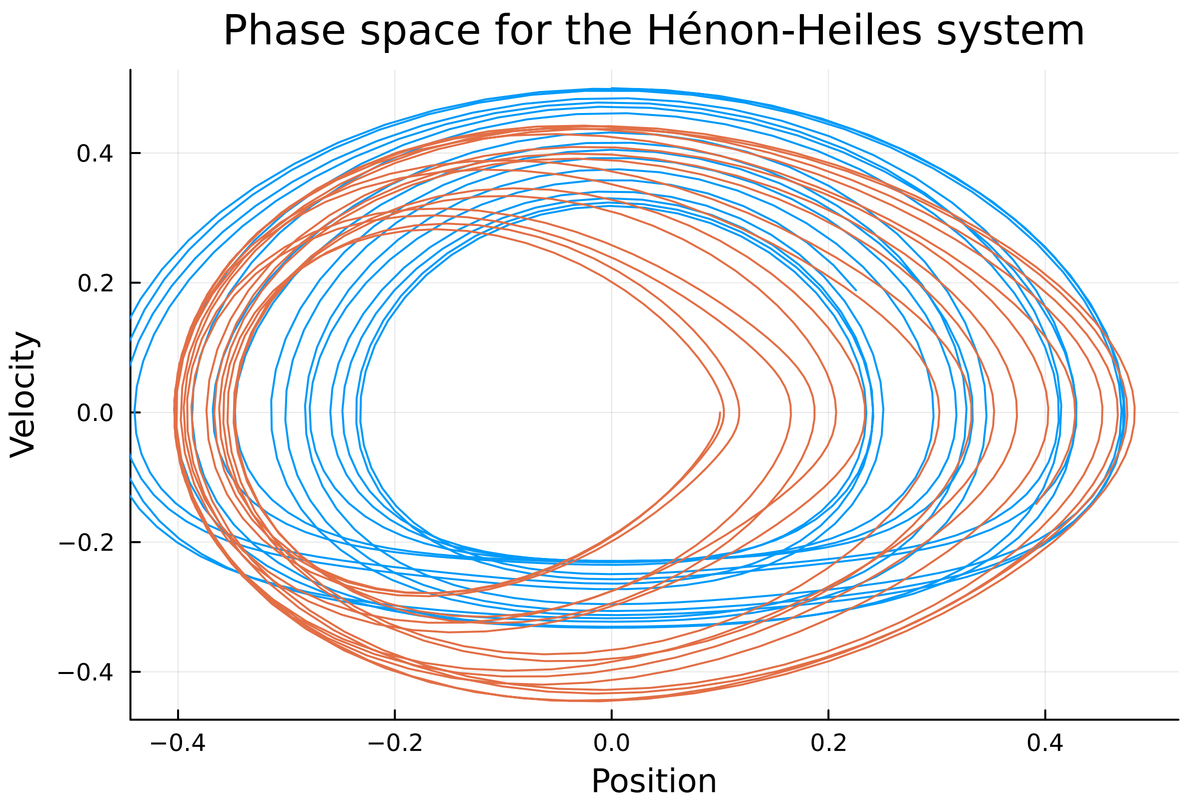
Out[8]:



In [9]:
```
#Optional Sanity check - what do you think this returns and why?
@show sol.retcode

#Plot -
plot(sol, vars=(1,3), title = "Phase space for the Hénon-Heiles system", xaxis = "Position", yaxis = "Veloc
plot!(sol, vars=(2,4), leg = false)
```

```
sol.retcode = SciMLBase.ReturnCode.Success
```

Out[9]:



In [ ]:
```
#We map the Total energies during the time intervals of the solution (sol.u here) to a new vector
#pass it to the plotter a bit more conveniently
energy = map(x->E(x...), sol.u)

#We use @show here to easily spot erratic behaviour in our system by seeing if the loss in energy was too gı
@show ΔE = energy[1]-energy[end]

#Plot
plot(sol.t, energy .- energy[1], title = "Change in Energy over Time", xaxis = "Time in iterations", yaxis =
```

Example: Symplectic Integration

To prevent energy drift, we can instead use a symplectic integrator. We can directly define and solve the SecondOrderODEProblem:

In [ ]:
```
function HH_acceleration!(dv,v,u,p,t)
    x,y  = u
    dx,dy = dv
    dv[1] = -x - 2x*y
    dv[2] = y^2 - y -x^2
end
initial_positions = [0.0,0.1]
initial_velocities = [0.5,0.0]
prob = SecondOrderODEProblem(HH_acceleration!,initial_velocities,initial_positions,tspan)
sol2 = solve(prob, KahanLi8(), dt=1/10);
```

In [ ]:
```
# Plot the orbit
plot(sol2, vars=(3,4), title = "The orbit of the Hénon-Heiles system", xaxis = "x", yaxis = "y", leg=false)
```

In [ ]:
```
plot(sol2, vars=(3,1), title = "Phase space for the Hénon-Heiles system", xaxis = "Position", yaxis = "Veloc
plot!(sol2, vars=(4,2), leg = false)
```

but now the energy is essentially zero

In [ ]:
```
energy = map(x->E(x[3], x[4], x[1], x[2]), sol2.u)
#We use @show here to easily spot erratic behaviour in our system by seeing if the loss in energy was too g
@show ΔE = energy[1]-energy[end]

#Plot
plot(sol2.t, energy .- energy[1], title = "Change in Energy over Time", xaxis = "Time in iterations", yaxis
```

Example: Runge-Kutta-Nyström solver

And let's try to use a Runge-Kutta-Nyström solver to solve the problem from the example above.

Note that:

1) Runge-Kutta-Nyström isn't symplectic.

2) We are using the DPRKN6 sovler at reltol=1e-3 (the default), yet it has a smaller energy variation than Vern9 at abs_tol=1e-16, rel_tol=1e-16. Therefore, using specialized solvers to solve its particular problem is very efficient.

In [ ]:
```
sol3 = solve(prob, DPRKN6());
energy = map(x->E(x[3], x[4], x[1], x[2]), sol3.u)
@show ΔE = energy[1]-energy[end]
gr()
plot(sol3.t, energy .- energy[1], title = "Change in Energy over Time", xaxis = "Time in iterations", yaxis
```

Sources:

1. Boyce, William E., DiPrima, Richard C. Elementary Differential Equations and Boundary Value Problems 9th Edition
2. Differential Equations Foundation Pre-Master Course for RealMaths Double Degree at L'Aquila University (Prof. Michele Palladino)
3. Simmons, George F., Differential Equations With Applications and Historical Notes 3rd Edition
4. https://tutorial.math.lamar.edu/classes/de/directionfields.aspx (https://tutorial.math.lamar.edu/classes/de/directionfields.aspx)
5. https://x-engineer.org/plot-vector-field/ (https://x-engineer.org/plot-vector-field/)
6. https://nextjournal.com/sosiris-de/ode-diffeq (https://nextjournal.com/sosiris-de/ode-diffeq)
7. https://tutorials.sciml.ai/html/models/01-classical_physics.html (https://tutorials.sciml.ai/html/models/01-classical_physics.html)

8. https://scipython.com/blog/the-double-pendulum/ (https://scipython.com/blog/the-double-pendulum/)

Appendix

In [14]:
```
# To activate project designated for Differential Equations
# Create an empty folder named DifferentialEquations that is in one folder with this notebook
import Pkg
Pkg.activate("DifferentialEquations")
```

```
  Activating project at `~/LasthrimProjection/JupyterLab/DifferentialEquations`
```

In [16]:
```
]st
```

```
      Status `~/LasthrimProjection/JupyterLab/DifferentialEquations/Project.toml`
  [0c46a032] DifferentialEquations v7.6.0
  [1dea7af3] OrdinaryDiffEq v6.29.3
  [91a5bcdd] Plots v1.35.6
  [f2b01f46] Roots v2.0.8
  [24249f21] SymPy v1.1.7
```

In [ ]:
```
ENV["PYTHON"] = "/home/browni/.julia/conda/3/envs/lasthrim_env/bin/python"
```

In [6]:
```
#Add a package
import Pkg;

Pkg.add("Roots")
```

```
    Updating registry at `~/.julia/registries/General.toml`
   Resolving package versions...
   Installed ConstructionBase ─ v1.4.1
   Installed Setfield ───────── v1.1.1
   Installed Roots ──────────── v2.0.7
    Updating `~/LasthrimProjection/JupyterLab/DifferentialEquations/Project.toml`
  [f2b01f46] + Roots v2.0.7
    Updating `~/LasthrimProjection/JupyterLab/DifferentialEquations/Manifest.toml`
  [38540f10] + CommonSolve v0.2.1
  [187b0558] + ConstructionBase v1.4.1
  [f2b01f46] + Roots v2.0.7
  [efcf1570] + Setfield v1.1.1
  [9fa8497b] + Future
Precompiling project...
  ✓ ConstructionBase
  ✓ Setfield
  ✓ Roots
  3 dependencies successfully precompiled in 30 seconds (221 already precompiled)
```

In [ ]: