Boundary Value Problems 2.

We want to solve

\[ x''(t) = f(t, x, x') \]
\[ a_0 x(a) - a_4 x'(a) = \gamma_1 \]
\[ b_0 x(b) + b_4 x'(b) = \gamma_2 \]

We have observed that the solution to the initial value problem

\[ x(a) = a_4 s - c_4 \gamma_1 \]
\[ x'(a) = a_0 s - c_0 \gamma_1 \]

with \( a_4 c_0 - a_0 c_4 = 1 \), denoted \( x(t; s) \), obeys our boundary conditions at \( a \). At \( b \), we define

\[ \varphi(s) := b_0 x(b; s) - b_4 x(b; s) - \gamma_2. \]
we are searching for \( s_* \) so that \( \Phi(s_*) = 0 \). We want to compute

\[
\Phi'(s) = b_0 \frac{\partial}{\partial s} x(t; s) + b_1 \frac{\partial^2}{\partial s^2} x(t; s)
\]

But how do we compute \( \frac{\partial}{\partial s} x(t; s) \)?

Here's the idea: for any \( s \) we have

\[
x''(t; s) = f(t, x(t; s), x'(t; s))
\]

If we differentiate both sides w.r.t. \( s \), we have an o.d.e in the auxiliary function

\[
y(t) = \frac{\partial}{\partial s} x(t; s)
\]

given by

\[
y''(t) = f_2(t, x(t; s), x'(t; s)) y(t)
\]
\[+ f_3(t, x(t; s), x'(t; s)) y'(t).
\]

with initial conditions
given by

\[ y(a) = a_1, \quad y'(a) = a_0. \]

If we introduce the additional auxiliary functions

\[ z(t) = x'(t) \]
\[ \omega(t) = y'(t) \]

this is a new system of ODE:

\[ x'(t) = z(t) \]
\[ z'(t) = f(t, x(t), z(t)) \]
\[ y'(t) = \omega(t) \]

\[ \omega'(t) = f_2(t, x(t), z(t)) y(t) + f_3(t, x(t), z(t)) \omega(t) \]

where \( f_2 \) and \( f_3 \) are the partials of \( f \) w.r.t. the second and third
variables.

Example.

\[ x'' = -x + \frac{2(x')^2}{x}, \quad -1 < \epsilon < 1 \]

with

\[ x(1) = x(-1) = \frac{1}{e + \epsilon/e}. \]

The true solution is

\[ x(\xi) = \frac{1}{e^x + e^{-x}} = \frac{1}{2} \sech x. \]

Since implementing the shooting method is part of the project, we won't do this in Mathematica.
But we will compute the system to solve:

\[
X'(t) = Z(t) \\
Z'(t) = -X(t) + \frac{2(x')^2}{x} \\
y'(t) = \omega(t) \\
\omega'(t) = (-1 - \frac{2(x')^2}{x^2})y(t) + 4 \frac{\delta_x Z(t)^2 \omega(t)}{x(t)}
\]

with initial conditions

\[
X(-1) = \frac{1}{e + \sqrt{e}} \\
Z(-1) = 5 \\
y(-1) = 0 \\
\omega(-1) = 1
\]
We would solve this and use

\[ \Phi'(s) = y(1), \quad \Phi(s) = X(1) - \frac{1}{e^{+1/e}}. \]

as initial data for Newton's method.

Now there are various possibilities.

Remark. We replicated a general theorem about how solutions of an ODE depend on parameters in the equation. The general theorem here is due to Peano, and can be found on page 95 of Hartmann's O.D.E. book.
What can go wrong here?

* How do we guess $s_0$?
* Does Newton's method converge?
* What if the problem is very sensitive to $s$? {This is likely, as }

$$|x(b; s) - x(b, s_{th})| \approx e^{b-a} h$$

* How many times must we solve the IVP? With what stepsizes?

For all these reasons, it can be valuable to have different methods where the convergence to a solution is better controlled.
Idea. Suppose we choose to divide the interval \([a, b]\) into equal intervals by \(t_0, t_1, \ldots, t_n\).

We think of the set of

\[ X_i = x(t_i) \]

as a set of \(n+1\) variables, obeying some equations, and solve the equations.

Suppose that we are in the linear case

\[ x''(t) = u(t) + v(t)x(t) + w(t)x'(t) \]
If we let
\[ X'(t) \approx \frac{1}{2h} [x(t+h) - x(t-h)] \]
\[ X''(t) \approx \frac{1}{h^2} [x(t+h) - 2x(t) + x(t-h)] \]

We can write the ODE as a system
\[ \frac{1}{h^2} [x_{i+1} - 2x_i + x_{i-1}] = u(t_i) + v(t_i) x_i + \omega(t_i) \left[ \frac{1}{2h} [x_{i+h} - x_{i-h}] \right] \]

of linear equations in the \( x_i \), which can be written
\[ \left( \frac{1}{h^2} + \omega_i \frac{1}{2h} \right) x_{i-1} + \left( -\frac{2}{h^2} - v_i \right) x_i + \left( \frac{1}{h^2} - \omega_i \right) x_{i+1} = u_i \]

Multiplying through by \(-h^2\), we get
\[ -(1 + h\omega_i/2) x_{i-1} + (2 + h^2v_i) x_i + (h^2/2 - \omega_i - 1) x_{i+1} = -h^2 u_i \]

Now the free variables are really only \( x_4, \ldots, x_{n-1} \) since the boundary
conditions specify \( x_0 = X(a) \) and \( x_n = X(b) \). We have a matrix in the form

\[
\begin{bmatrix}
\vdots & \cdots & \cdots & \cdots \\
& & & \\
& & & \\
& & & \\
& & & \\
\end{bmatrix}
\begin{bmatrix}
X_1 \\
X_2 \\
\vdots \\
X_{n-1} \\
\end{bmatrix}
= 
\begin{bmatrix}
\uparrow \\
- \hbar^2 u_i \\
\downarrow \\
\end{bmatrix}
\]

This is a special type of matrix, called Tridiagonal, for which \( AX = b \) can be solved in time \( O(n) \), as we will see in the next unit!

This is called a discretization or a finite element method.

(Mathematica demo)