Advanced Mathematics – WS2021 – Lab 6 – ODE (again)

Submission is for next Monday, December the 21th! Do your best! 😊 Good luck!

Exercise 1 – Substitution and Wronskian

Let y_1 and y_2 be two solutions of $(1.1)\ y'' + p(x)y' + q(x)y = 0$.

a- prove that $\frac{dW}{dx} = -p(x)W$ where $W(y_1, y_2)$ is the Wronskian.

b- prove that if $p(x) = 0$ then $W(y_1, y_2)$ is always a constant.

c- verify b- by direct calculation for $y'' + k^2y = 0\ with\ k \neq 0$ whose general solution is $y_1 = c_1 sinkx + c_2 coskx$.

Now, take $(1.2)\ y'' - 2y' + y = 0$ with a given solution $y_1 = e^x$.

d- find a second solution y_2 to (1.2) putting $y_2 = ue^x$ and determining $u(x)$ by substitution into the ODE.

e- find a second solution y_2 to (1.2) by determining first $W(y_1, y_2)$ using a-.

f- what is the most general form for y_2?

Exercise 2 – Set-up an ODE

Given a linear differential equation of second order in the form $Ay'' + By' + y = 0$.

Verify that the choice of 2 linear independent solutions $y_1 = x^a$ and $y_2 = x^b$ with $a, b \epsilon \mathbb{C}$ leads necessarily to an Euler-Cauchy differential equation. Express the coefficients depending on (a, b, x).

Hint: Cramer's rule might lead to an elegant and compact solution. ;)

Exercise 3 – Horner scheme on Matlab

In case of constant coefficients, the procedure can be extended to higher order differential equations, which requires the roots of a polynomial $P_n(x) = a_n x^n + a_{n-1}x^{n-1} + \ldots + a_1 x + a_0$ with degree n. Implement the *Horner scheme*

$$\begin{array}{cccccc} a_n & a_{n-1} & a_{n-2} & \ldots & a_1 & a_0 \\ & b_n x_0 & b_{n-1}x_0 & \ldots & b_2 x_0 & b_1 x_0 \\ \hline b_n & b_{n-1} & b_{n-2} & \ldots & b_1 & b_0 = P(x_0) \end{array} \qquad \text{with } b_i = \begin{cases} a_n & i = n \\ a_i + b_{i+1}x_0 & \text{else} \end{cases}$$

and note down the solution of the differential equation

$$2y'''' + 4y''' - 34y'' - 36y' + 144y = 0$$

- function call: `horner(an,x0)` with the coefficient vector `an` $=[a_n, a_{n-1}, \ldots, a_0]$ and the guess `x0` for the root