The string method

Daniel Thomas Hayes

June 8, 2020

A method for finding exact solutions of differential equations is proposed.

Many exact solutions of differential equations are known. For instance we know that the problem

$$\frac{du}{dt} = u, \ \ u(0) = 1 \tag{1}$$

has the exact solution

$$u = e^t. \tag{2}$$

The problem

$$\frac{du}{dt} = u^2, \ \ u(0) = 1 \tag{3}$$

has the exact solution

$$u = \frac{1}{1-t}. \tag{4}$$

An interesting observation is that often a solution is a finite length formula and its symbols belong to a finite language.

Definition 1

I will define a finite language to be a finite length list of strings containing mathematical symbols. For example

$$L = \text{“/”}, \text{“}-\text{”}, \text{“ ”}, \text{“(”}, \text{“)”}, \text{“1”}, \text{“t”}, \text{“exp(”} \tag{5}$$

is a finite language.

Definition 2

I will define a finite length formula to be a formula created from a finite language which uses a finite number of strings from that language. For example, we can concatenate the strings

$$\text{“exp(”}, \text{“t”}, \text{“)”} \tag{6}$$

from the finite language L to construct the solution (2). Likewise, we can concatenate the strings

$$\text{“1”}, \text{“/”}, \text{“(”}, \text{“1”}, \text{“}-\text{”}, \text{“t”}, \text{“)”} \tag{7}$$

from the finite language L to construct the solution (4). Some formulas created this way will not make sense. For example, when we concatenate

$$\text{“)”}, \text{“1”}, \text{“/”}, \text{“}+\text{”} \tag{8}$$

the formula will not make sense and so it would need to be excluded.

String method

Begin with a finite language L and construct all possible finite length formulas up to a certain length N. Each of these finite length formulas that make sense are substituted into the problem to see if any of them solve the problem. The exact solution will be found provided that the exact solution to the problem is a solution in L with length at most N.

This method can easily be implemented in computer software such as Maple to solve the problems (1), (3). Other problems could require much more computing power.