

## EXAMPLE OF Lua $\text{\LaTeX}$ WITH ASMECONF.CLS FOR ODE INTEGRATION

John H. Lienhard V\*

Department of Mechanical Engineering,  
Massachusetts Institute of Technology,  
Cambridge, MA

### ABSTRACT

*This paper is an example of using asmeconf with Lua $\text{\LaTeX}$  to solve an ODE initial value problem using a fourth-order Runge-Kutta method and to plot the result using PGFPLOTS. The use of a landscape figure is also illustrated. References are given for further reading.*

**Keywords:** asmeconf, Lua $\text{\LaTeX}$ , ODE, pgfplots, landscape

### NOMENCLATURE

$A$	Constant parameter [–]
$t$	Time [s]
$y(t)$	Position [m]

### 1. INTRODUCTION

Lua $\text{\LaTeX}$  is built upon the Lua programming language [1]. By directly using Lua code in a  $\text{\LaTeX}$  file, we can accomplish a wide range of tasks, as illustrated in the open-access paper by Montijano et al. [2]. In the present example, we follow Montijano et al. in solving a nonlinear first-order ordinary differential equation and plotting the result—all within a single  $\text{\LaTeX}$  file!

### 2. SOLUTION TO AN INITIAL VALUE PROBLEM

We consider an initial value problem like that of Montijano et al.:

$$y'(t) = A \cdot y(t) \cos\left(t + \sqrt{1 + y(t)}\right) \text{ with } y(0) = 1 \quad (1)$$

Here,  $A$  is a constant. We may adopt a fourth-order Runge-Kutta algorithm for the integration, which we shall perform to  $t = 30$  s using a 400 point discretization. The details of the Runge-Kutta algorithm and a listing of the code are given in Montijano et al. (You can also read the code in the present .tex file.)

The algorithm is implemented directly in the preamble of this file, and the results are plotted in Fig. 1 for  $A =$

$\{0.25, 0.5, 0.75, 1.0\}$ . Plotting is done using the PGFPLOTS package [3].

Landscape figures, such as Fig. 1, may be produced at full-page size by putting `\usepackage[figuresright]{rotating}` in your .tex file's preamble and using the `sidewaysfigure*` environment [4].

### 3. CONCLUSION

Lua $\text{\LaTeX}$  enables numerical computations within a  $\text{\LaTeX}$  environment. By combining this capability with PGFPLOTS, the need for separate numerical and/or graphics packages can be reduced.

### ACKNOWLEDGMENTS

The example shown in this paper is directly based on an example given by Montijano et al. [2]. Additional examples, such as the Lorenz attractor, are contained in that paper.

### REFERENCES

- [1] Ierusalimsky, Roberto, de Figueiredo, Luiz Henrique and Celes, Waldemar. *Lua 5.3 Reference Manual*. Pontifical Catholic University, Rio de Janeiro, Brazil (2017). URL <https://www.lua.org/manual/5.3/>.
- [2] Montijano, Juan I., Pérez, Mario, Rández, Luis and Varona, Juan Luis. “Numerical methods with Lua $\text{\LaTeX}$ .” *TUGboat* Vol. 35 No. 1 (2014): pp. 51–56. URL <https://tug.org/TUGboat/tb35-1/tb109montijano.pdf>. Open access.
- [3] Feuersänger, Christian. *Manual for Package PGFPLOTS*, Version 1.17. Comprehensive  $\text{\TeX}$  Archive Network (2020). Accessed January 4, 2021, URL <https://ctan.org/pkg/pgfplots>.
- [4] Fairbairns, Robin, Rahtz, Sebastian and Barroca, Leonor. “A Package for Rotated Objects in  $\text{\LaTeX}$ .” Version 2.16d. Comprehensive  $\text{\TeX}$  Archive Network (2016). Accessed October 2, 2019, URL <https://www.ctan.org/pkg/rotating>.

\*Corresponding author: lienhard@mit.edu  
Version 1.0, December 26, 2021

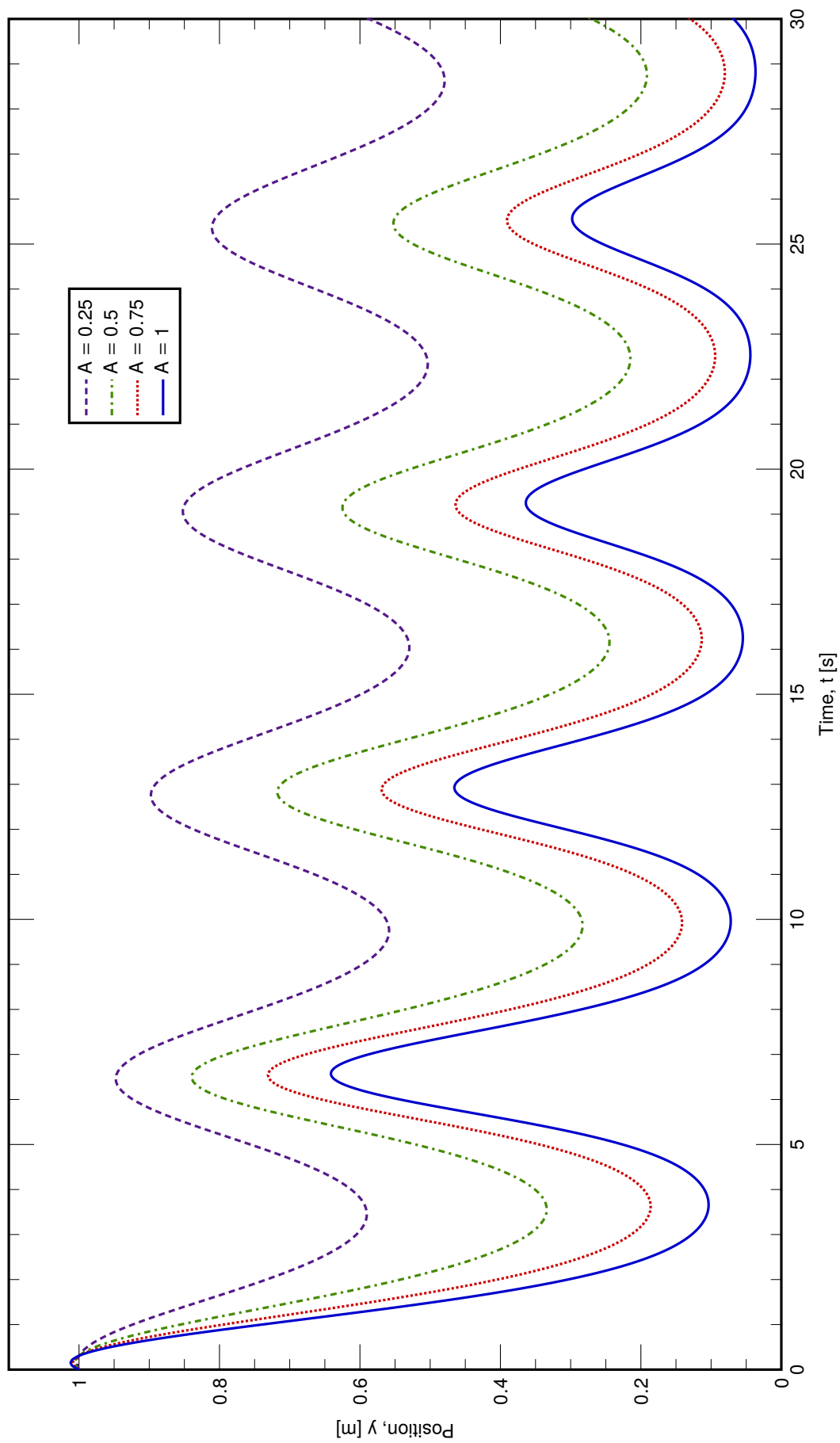


FIGURE 1: A TRIAL OF PGFPLOT WITH LUACODE RUNGE-KUTTA INTEGRATION