A FAST ONE-DIMENSIONAL NONLINEAR THERMOACOUSTIC CODE

J.A. de Jong[1*], Y.H. Wijnant[1], D. Wilcox[2] and A. de Boer[1]

[1] *Applied Mechanics, Engineering Technology, University of Twente, P.O. Box 217, 7500AE, Enschede, The Netherlands.*
[*]Corresponding author's e-mail: j.a.dejong@utwente.nl
[2] *Chart Inc. - Qdrive, 302 Thenth St., Troy, NY 12180, USA*

Keywords: *Nonlinear thermoacoustics, numerical modeling, design tool*

Introduction

The design of thermoacoustic (TA) systems is often aided with computational models. These computational models should be both fast, to quickly obtain feedback on design decisions, and accurate to know that the results are representative for a real TA system. However, in many cases these demands counteract each other. This often leads to a trade-off between computational cost and accuracy when choosing a specific design tool.

Currently, several computer codes are available to aid the design of TA systems. For example, the linear TA theory is implemented in the widely used computer code DeltaEC [5]. DeltaEC also has capabilities to compute nonlinear effects such as Gedeon streaming up to second order. For detailed simulations of the full nonlinear governing equations, commercial Computational Fluid Dynamics (CFD) software can be used. However, CFD tends to be computationally costly and time consuming.

This abstract describes the development and the first tests of a new one-dimensional nonlinear TA code, called the *ThermoAcoustic System Modeling Environment Twente* (TASMET). The code is capable of solving one-dimensional TA systems in a computationally efficient manner. It makes use of one-dimensional models to retain a low computational cost. However, to achieve better accuracy for high amplitude oscillations, the existing one-dimensional linear thermoacoustic theory is extended to the nonlinear domain.

The code

The low computational cost of the code is achieved by directly solving the periodic steady state of a TA system. For solving a TA system, a periodic (but not harmonic) solution is assumed *a priori*. This solution is expressed in terms of a truncated Fourier series. For an arbitrary dependent variable ξ this Fourier series is

$$\xi(t) \approx \sum_{n=0}^{N_f} \hat{\xi}_n e^{in\omega t},$$

where N_f is the number of frequencies (harmonics) taken into account to describe the solution, $\hat{\xi}_n$ are the Fourier coefficients, n is the harmonic number and ω the fundamental angular frequency. Then, instead of doing a time integration, the numerical algorithm solves for the Fourier coefficients of each dependent variable in the system [2].

The actual numerical routines are written in C++ and use the Armadillo [4] linear algebra package. The nonlinear system of equations is solved using the Newton-Raphson method with an exact evaluation of the Jacobian matrix. At each Newton iteration, the sparse algebraic system is solved

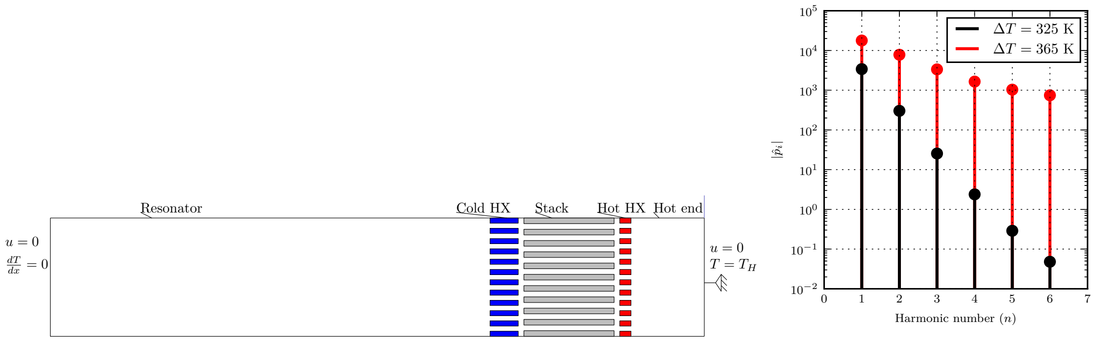

Figure 1: *Left: schematic picture of the geometry of a standing wave thermoacoustic engine. On the bottom, the wall temperature as a function of the position is shown. Right: results for the amplitudes of 6 pressure harmonics at the leftmost end of the geometry for two temperature differences across the stack.*

using the SuperLU [3] sparse system solver. The interface to the code has been made accessible using the portable and publicly available scripting language Python [1].

Example

A TA system consists of several tube-like segments, such as heat exchangers, a stack/regenerator and a feedback tube. In these segments, for the fluid domain the one-dimensional governing equations are solved, including the thermal equation of state. An example of a simple TA system is given in Figure (1). For each of the segments, a model is available. The segments are coupled at the ends, where the coupling equations dictate conservation of mass, momentum and energy.

Solving this system is done in less than a minute. The code outputs the Fourier coefficients for each of the dependent variables. For example, the spatial distribution of the absolute value of the first six acoustic pressure phasors is given on the right in Figure (1).

Future work

In future work, the modeling capabilities of the code will be expanded. Several commonly used models will be added, such as a model for a thermal buffer tube, and a porous material (random fiber, stacked screen) regenerator model.

Acknowledgements

This research has been carried out as a part of the Agentschap NL EOS-KTO (KTOT03009) research program. The financial support is gratefully acknowledged.

References

[1] Guido van Rossum. "The Python Programming Language - www.python.org" (**2015**).

[2] de Jong, J. A., Wijnant, Y. H., Wilcox, D., and de Boer, A. "Modeling of thermoacoustic systems using the nonlinear frequency domain method". To be published.

[3] Li, X. S., Demmel, J. W., Gilbert, J. R., Grigori, i., Shao, M., and Yamazaki, I. *SuperLU Users' Guide*. Technical Report LBNL-44289, Lawrence Berkeley National Laboratory (**1999**).

[4] Sanderson, C. *Armadillo: An Open Source C++ Linear Algebra Library for Fast Prototyping and Computationally Intensive Experiments*. Technical report, NICTA, Australia (**2010**).

[5] Ward, W. C. and Swift, G. W. "Design environment for low-amplitude thermoacoustic engines". The Journal of the Acoustical Society of America, **95**, (1994), 3671–3672.