



# Wrapping CALE Commands using SWIG



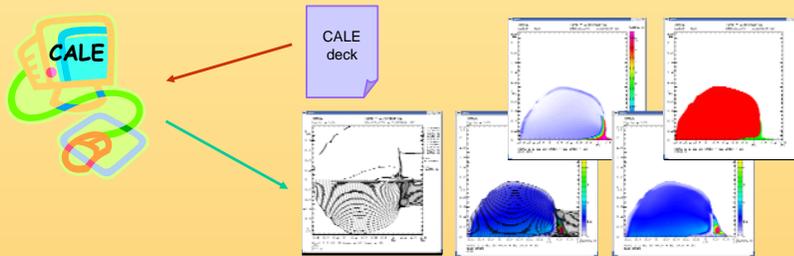
Nija Shi, University of California, Davis

AX Division, Paul Amala and Rob Managan  
Lawrence Livermore National Laboratory

*CALE is a 2D ALE hydrodynamics computer program written in ANSI C. While CALE is portable in various platforms, we want to increase CALE's usability by making CALE available in popular programming languages, such as Python. Our goal is to build a Python front-end for CALE using SWIG. In this poster, we show our investigation on SWIG's capability on wrapping CALE commands, illustrate our preliminary results on wrapping several CALE commands, and finally discuss our future plan.*

## Introduction

- CALE (C Arbitrary Lagrangian Eulerian) is a user-interactive 2D ALE hydrodynamics computer program written in the ANSI C.
- CALE is also itself a command language recognized by its embedded interpreter.

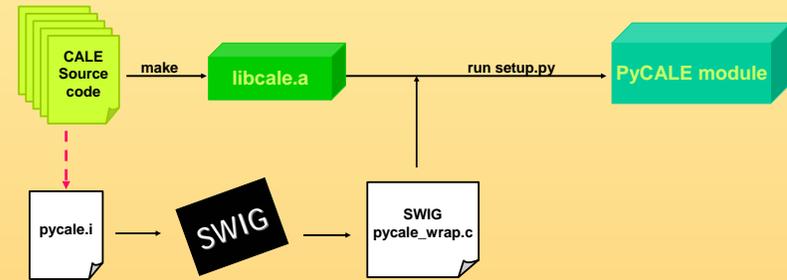


## Goals

- Minimal modification of the original CALE source code
- Pythonize CALE's original syntax in the new PyCALE module

CALE	PyCALE
rdmp, fm	pycale.rdmp(), pycale.fm()
tv 600 600	pycale.tv(600, 600)
winp -5 5 0 .1	pycale.winp(-5,5,0,.1)
pltstep 10 1	pycale.pltstep(10,1)
plm/plmoff	pycale.plm()/pycale.plmoff()
plc den	pycale.plc("den")
{GAMMA}	pycale.gamma['GAMMA'].val
kmax	pycale.cvar.kmax

- SWIG (Simple Wrapper and Interface Generator)



## Discussion

- **Problem:** CALE commands are processed through its embedded interpreter. Multiple commands can be mapped to the same embedded function by passing different arguments. Most embedded functions require information from standard input.
- **Solution:** To retain the original look-and-feel, we use various SWIG directives and redirect parameters passed from PyCALE to the standard input for CALE.

## Future Plan

We want to automate this process using source code analysis, such as data-flow, control-flow, and interprocedural analyses.