SOLVCON: Software-Engineering Simulations of Conservation Laws
SciPy in PyCon APAC 2014
Yung-Yu Chen

Description

SOLVCON is a collection of Python-based conservation-law solvers that use the space-time Conservation Element and Solution Element (CESE) method. SOLVCON targets at solving problems that can be formulated as a system of first-order, linear or non-linear partial differential equations (PDEs), by providing a well-engineered code base. In addition to high-performance numerical methods, modern conservation-law simulations require versatile supportive functionalities including parallel computing (sometimes hybrid parallel computing for heterogeneous architectures) and flexible input, output, and work flow. Most code in the software is written for these supportive functionalities, although researchers are supposed to focus their time on the physical processes or numerical methods. To address this challenge, SOLVCON organizes these supportive functionalities and lays out a common structure for conservation-law solvers. SOLVCON further wants to emphasize the connections among documenting, coding, and testing. Many Python-based tools support the development, like Sphinx, NumPy, Cython, nose, etc. SOLVCON is open-source with a BSD-style license, and we hope people interested in this subject can take a look and provide comments.

Additional Information

Project website: http://solvcon.net/
About the author: http://solvcon.net/yyc/

Abstract

SOLVCON is a collection of Python-based conservation-law solvers that use the space-time Conservation Element and Solution Element (CESE) method. SOLVCON targets at solving problems that can be formulated as a system of first-order, linear or non-linear partial differential equations (PDEs), by providing a well-engineered code base. SOLVCON has been applied to solve problems of supersonic flows, stress waves, waves in bio-tissues, and hydro-acoustics. SOLVCON has been used in simulations utilizing more than 1,000 cores with hybrid parallel computing. The website and documents can be accessed at http://solvcon.net/. SOLVCON uses a BSD-style open-source license.

Modern conservation-law simulations require versatile supportive functionalities including parallel computing (sometimes hybrid parallel computing for heterogeneous architectures) and flexible input, output, and work flow, in addition to high-performance numerical methods. However, most code in the software is written for these supportive functionalities. Because researchers are supposed to focus their time on the physical processes or numerical methods, it is common for them to minimize the efforts in coding the supportive functionalities. The implementation of the functionalities usually depend on a random software or hardware configuration and is fragile. Results produced by these research codes are not always reproducible. Any change in OS, runtime libraries, CPU, memory, etc., might affect the simulation results, and there is usually no systematic way to verify that.

The productivity of the code development also suffers from careless designs of supportive functionalities. Inflexible work flow forbids the researchers from quickly or systematically analyzing the produced results. Because debugging numerical simulations requires examining the results from various aspects, usually with different visualizations, weak implementation of supportive functionalities causes trouble.

To address these challenges, SOLVCON organizes supportive functionalities and lays out a common structure for conservation-law solvers. The space-time CESE method was chosen as the

model numerical method for its novelty, versatility, and simplicity. On top of the supportive functionalities, SOLVCON starts to emphasize the connections among documenting, coding, and testing. Because of the complexity in the computer code, hyperlinked online documents are a better medium for delivering the research results than traditional papers published in journals. The computer code, theoretical development, and result analysis should be organized in harmony to make the best explanation of the research or application.

Under all the development Python is the key. Without the scientific-computing ecosystem provided by the Python community, nothing in SOLVCON can be done. The employed tools include Sphinx, Mercurial, NumPy, Cython, nose, to name but a few. The development of SOLVCON also need more people to join. Contribution and comment in any form are more than welcomed.