



ELSEVIER

Contents lists available at ScienceDirect

Data in brief

journal homepage: www.elsevier.com/locate/dib

Data Article

Coefficient datasets for high-order, stable, and conservative boundary schemes for central and compact finite differences



P.T. Brady*, D. Livescu

CCS-2, Los Alamos National Laboratory, Los Alamos, NM 87544, United States

ARTICLE INFO

Article history:

Received 12 February 2019

Received in revised form 24 April 2019

Accepted 23 May 2019

Available online 1 June 2019

ABSTRACT

Stable and conservative numerical boundary schemes, for both compact and explicit (central) finite differences require a number of parameters that must be tuned for stability. Values of these coefficients for 4th, 6th, and 8th boundary schemes are given in this article. The stability of the schemes is demonstrated through a series of numerical tests in “High-Order, Stable, and Conservative Boundary Schemes for Central and Compact Finite Differences” Brady and Livescu, 2019. These tests include: a neutrally stable constant coefficient hyperbolic system, a two-dimensional varying coefficient hyperbolic scalar equation and, examining the transport of an inviscid vortex using the compressible Euler equations. The error norms for the variety of tests associated with different the schemes for different grid resolutions and time-step constraints are given in the accompanying databases.

© 2019 The Authors. Published by Elsevier Inc. This is an open access article under the CC BY license (<http://creativecommons.org/licenses/by/4.0/>).

1. Data

There are two types of data presented in this article. The first type is an ascii representation of the constraints on the coefficients for each numerical boundary scheme. There are six schemes in total. Explicit (central) differencing schemes of 4th, 6th, and 8th order are labeled “E4.txt”, “E6.txt”, and

DOI of original article: <https://doi.org/10.1016/j.compfluid.2018.12.010>.

* Corresponding author.

E-mail address: ptb@lanl.gov (P.T. Brady).

<https://doi.org/10.1016/j.dib.2019.104086>

2352-3409/© 2019 The Authors. Published by Elsevier Inc. This is an open access article under the CC BY license (<http://creativecommons.org/licenses/by/4.0/>).

Specifications table

Subject area	<i>Computational physics</i>
More specific subject area	<i>Numerical methods</i>
Type of data	<i>ASCII-text files and sqlite3 databases</i>
How data was acquired	<i>Numerical simulation</i>
Data format	<i>analyzed</i>
Experimental factors	<i>A large number of numerical simulations were performed</i>
Experimental features	<i>Stable numerical schemes located using a gradient ascent approach</i>
Data source location	<i>Los Alamos National Laboratory</i>
Data accessibility	<i>Data is with the article</i>
Related research article	<i>Brady, P. T & Livescu, D.; High-Order, Stable, and Conservative Boundary Schemes for Central and Compact Finite Differences; Computers & Fluids, Vol 183 (2019) pp. 84-101</i>

Value of the data

- With this data, any researcher may easily choose a high-order, stable numerical boundary scheme to couple with their choice of interior scheme.
- Researchers may also try out different free parameters to generate their own numerical boundary schemes, which automatically satisfy discrete conservation constraints.
- Analysis of these datasets may provide insight into the development of novel stability theories.

“E8.txt”, respectively. Compact differencing schemes of 4th, 6th, and 8th order are labeled “T4.txt”, “T6.txt”, and “T8.txt”, where the “T” prefix indicates a compact scheme with tri-diagonal structure. The equations are placed in an ascii format because their length can make copy/paste from a pdf document error prone. The variable names in txt files assumes that a stencil of length “t”, approximating the first derivative, at a point “i”, near the left boundary can be written as:

$$\sum_{k=-s_1}^{k=s} \beta_{ik} f'_{i+k} = \frac{1}{h} \sum_{j=0}^{j=t-1} \alpha_{ij} f_j$$

Where $s = 0$ for the explicit (central) differencing schemes and $s = 1$ for the tridiagonal compact schemes. For all schemes, $\beta_{i0} = 1$. To ease the processing of the txt files, the coefficients, $\beta_{i,\pm 1}$ are represented as “beta_i_p1/m1”. Each numerical boundary scheme is designed to be coupled to the appropriate centered differencing scheme in the interior of the domain.

The second type of data is a sqlite3 database for each of the schemes. Each sqlite database contains 6 tables. The most important table is “nbs”, for which the columns are a unique “id” followed by the free parameters for the scheme. The 5 other tables record the performance of the numerical boundary scheme on a variety of numerical experiments, which will be described in the next section. The full set of coefficients for any numerical boundary scheme can be computed by substituting the free parameters associated with an “id” from the nbs table into the constraints given in the corresponding txt file. As an example, the following python script (requires python 3) prints the coefficients associated with 15th T6 scheme. Note that this assumes the files “T6.db” and “T6.txt” are in the current directory.

```

import sqlite3 as db
from fractions import Fraction

scheme, nbs_id = "T6", 15

conn = db.connect(scheme+".db")
conn.row_factory = db.Row
cur = conn.cursor()
cur.execute("select * from nbs where id=?", (nbs_id,))
row = cur.fetchone()

params = {}
for k, v in zip(row.keys()[1:], row[1:]):
    params[k] = Fraction(v)

with open(scheme+".txt") as f:
    for line in f:
        k, v = line.split('=')
        params[k] = eval(v.strip(), globals(), params)
conn.close()

for k, v in params.items():
    print("{:10s} = {}".format(k, float(v)))

```

2. Experimental design, materials, and methods

The values of the free parameters contained in the “nbs” table were arrived at via the optimization procedure described in [1]. The schemes were subjected to a variety of numerical tests to verify their stability properties. The error norms of these numerical experiments are recorded in the remaining tables in each database. In all numerical tests described below, the spacial discretization is done a uniform Cartesian mesh and time integration is carried out using a standard RK4 method.

The first numerical experiment is recorded in section 4.2 of [1] and is reproduced here for completeness. It consists of solving the hyperbolic system:

$$\frac{\partial u}{\partial t} = \frac{\partial v}{\partial x}, \quad \frac{\partial v}{\partial t} = \frac{\partial u}{\partial x}, \quad x \in [0, 1]$$

with boundary and initial conditions:

$$u(0, t) = v(1, t) = 0$$

$$u(x, 0) = -\frac{3\pi}{2} \sin \frac{3\pi x}{2}, \quad v(x, 0) = 0$$

In the table “linear_system_cfl”, the simulations are run to a time of 500 for a large and small CFL at various grid resolutions (denoted by the “nx” column) and the maximum error in u and v are recorded in the “Linf_u” and “Linf_v” columns, respectively.

The table “linear_system_dt” records the same data but with constant timestep simulations so that the order of accuracy of the simulations can be computed. These are recorded in the “order_u” and “order_v” columns.

The second numerical experiment is described in section 4.3 of [1] and is reproduced here for completeness. It consists of solving the two-dimensional, varying coefficient scalar wave equation:

$$\frac{\partial u}{\partial t} + \frac{\partial \varphi}{\partial x} \frac{\partial u}{\partial x} + \frac{\partial \varphi}{\partial y} \frac{\partial u}{\partial y} = 0, \quad 0 \leq (x, y) \leq \sqrt{2}$$

Where $\varphi(x, y) = \sqrt{x(x - 0.25)^2 + (y - 0.25)^2}$, and with initial and boundary conditions:

$$u(x, y, 0) = \sin 2 \pi \varphi$$

$$u(0, y, t) = \sin 2 \pi (\varphi(0, y) - t)$$

$$u(x, 0, t) = \sin 2 \pi (\varphi(x, 0) - t)$$

In the table “linear_wave2d_cfl”, the simulations are run to a time of 1000 for a large and small CFL at various grid resolutions and the maximum error in u is recorded as “Linf_u”.

The table “linear_wave2d_dt” records the same data but with constant timestep simulations so that the order of accuracy of the simulations can be computed. This is recorded in the “order_u” column.

The final numerical experiment is also described in section 4.4 of [1]. In this test, the two-dimensional Euler equations are solved to examine the transport of vortex through a domain. A vortex of nondimensional circulation, $\varepsilon = 1.5$, is placed in the center of a computational domain of length $x \in [0, 20]$, $y \in [0, 10]$ discretized with $N_x N_y$ grid points, where $N_y = (N_x + 1)/2$. The background flow is a uniform flow in the x -direction with a Mach number of 2. The simulations are run to a non-dimensional time of 1000 at various grid resolutions and CFL's. The maximum and final error in pressure is recorded in the “vortex” table in the “Linf_p”, and “Linf_p_final” columns, respectively.

Acknowledgments

This work was supported by the US Department of Energy through the Los Alamos National Laboratory. Los Alamos National Laboratory is operated by Triad National Security, LLC, for the National Nuclear Security Administration of U.S. Department of Energy (Contract No. 89233218CNA000001). Computational resources were provided by the LANL Institutional Computing (IC) Program and Sequoia Capability Computing Campaign at Lawrence Livermore National Laboratory.

Conflict of Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Appendix A. Supplementary data

Supplementary data to this article can be found online at <https://doi.org/10.1016/j.dib.2019.104086>.

References

- [1] P.T. Brady, D. Livescu, High-order, stable, and conservative boundary schemes for central and compact finite differences, *Comput. Fluids* 183 (2019) 84–101.