

Code_Algorithms_Schemes_Test

July 14, 2021

Configuration

<IPython.core.display.HTML object>

1 TDMA in 1D

1.1 TDMA with noncyclic b.c.

The test equation is

The theoretical solution is

The calculated solution is

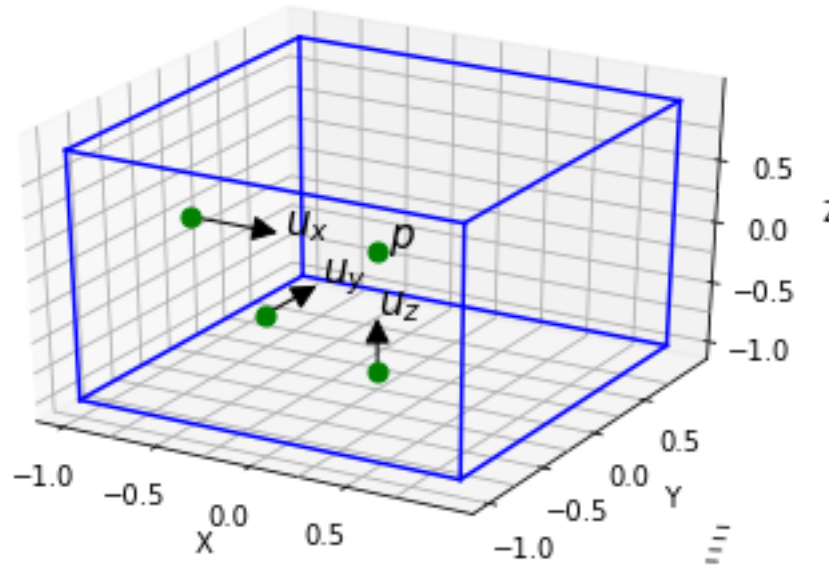
1.2 TDMA with cyclic b.c.

The test equation is

The theoretical solution is

The calculated solution is

2 Fullystaggered mesh



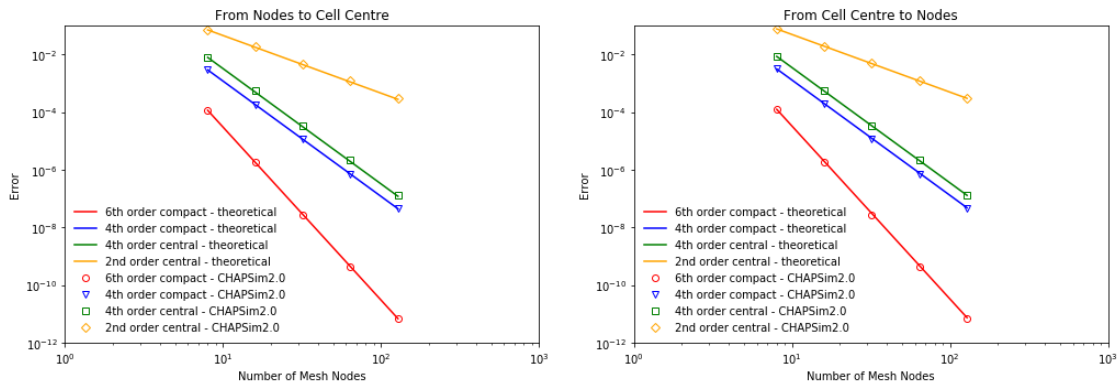
3 Interpolation

The input velocity is $u(x, y, z) = \sin(x) + \sin(y) + \sin(z)$. $x \in \{0, 2\pi\}$, $y \in \{-\pi, \pi\}$.

The interpolation involve two types of interpolation, * interpolation of u in x direction, from face (i', j, k) to cell centre (i, j, k) , called P2C * interpolation of u in y direction, from face (i', j, k) to edge (i', j', k) , called C2P

3.1 Uniform mesh with Periodic B.C.

Accuracy Validation of Interpolation



3.2 Uniform mesh with Dirichlet B.C.

```
File "<ipython-input-112-93039d2b99b8>", line 22
[9.5434761E-03, 1.7211037E-03, 9.5434761E-03]\,
```

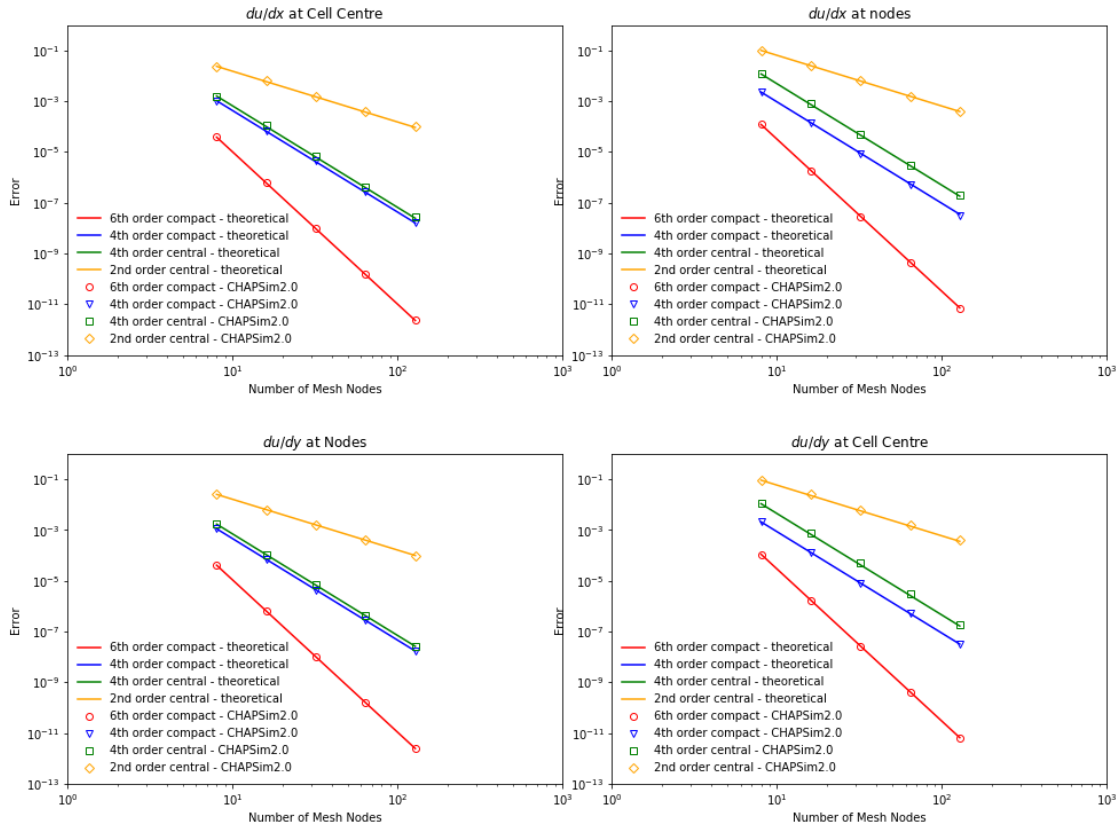
^

SyntaxError: unexpected character after line continuation character

4 First Derivative

The input velocity stored at staggered (i', j, k) is $u(x, y, z) = \sin(x) + \sin(y) + \sin(z)$ The first derivative involves, * du/dx, from face (i', j, k) to cell centre (i, j, k) , called P2C * du/dx, from face (i', j, k) to face (i', j, k) , called P2P * du/dy, from face (i', j, k) to edge (i', j', k) , called C2P * du/dy, from face (i', j, k) to face (i', j, k) , called P2P ## Uniform mesh with Periodic B.C.

Accuracy Validation of First Derivative



4.1 Uniform mesh with Dirichlet B.C.