Homework 8 Finite Volume

ATSC 507

Christopher Rodell

Question 1

(/5) Show that T_i (i.e. $T(x)$ at the centroid of control-volume CV_i) and $\bar{T}_i$ (i.e. the control-volume averaged value of $T(x)$ in CV_i) are the same only to second-order accuracy.
Hint 1 : Try expanding $\bar{T}_i$ at $x = x_i$
Hint 2 : $x_i = \frac{x_{i+\frac{1}{2}} + x_{i-\frac{1}{2}}}{2}$

In [1]:
```
from __future__ import print_function
from sympy import *
from devito import *


x, x0, h = symbols('x, x_i, h')
Fi, Fim1, Fip1 = symbols('F_{i}, F_{i-1}, F_{i+1}')
n = 3 # there are the coefficients c_0=Fi,  c_1=dF/h,  c_2=d**2F/h**2
c = symbols('c:3')
# define a polynomial of degree n
def P(x, x0, c, n):
   return sum( ((1/factorial(i))*c[i] * (x-x0)**i for i in range(n)) )
# now we make a matrix consisting of the coefficients
# of the c_i in the nth degree polynomial P
# coefficients of c_i evaluated at x_i
m11 = P(x0 , x0, c, n).diff(c[0])
m12 = P(x0 , x0, c, n).diff(c[1])
m13 = P(x0 , x0, c, n).diff(c[2])
# coefficients of c_i evaluated at x_i - h
m21 = P(x0-h, x0, c, n).diff(c[0])
m22 = P(x0-h, x0, c, n).diff(c[1])
m23 = P(x0-h, x0, c, n).diff(c[2])
# coefficients of c_i evaluated at x_i + h
m31 = P(x0+h, x0, c, n).diff(c[0])
m32 = P(x0+h, x0, c, n).diff(c[1])
m33 = P(x0+h, x0, c, n).diff(c[2])
# matrix of the coefficients is 3x3 in this case
M = Matrix([[m11, m12, m13], [m21, m22, m23], [m31, m32, m33]])


# matrix of the function values...actually a vector of right hand sides
R = Matrix([[Fi], [Fim1], [Fip1]])
# matrix form of the three equations for the c_i is M*X = R
# solution directly inverting the 3x3 matrix M:
X =  M.inv() * R
# note that all three coefficients make up the solution
# the first derivative is coefficient c_1 which is X[1].
print("The finite difernce third-order accurate approximation for the fi
rst derivative is: ")
print(together(X[1]))


########################################################################
################

########################################################################
################
Fi, Fim1, Fip1 = symbols('F_{i}, F_{i-1/2}, F_{i+1/2}')

# now we make a matrix consisting of the coefficients
# of the c_i in the nth degree polynomial P
# coefficients of c_i evaluated at x_i
m11 = P(x0 , x0, c, n).diff(c[0])
m12 = P(x0 , x0, c, n).diff(c[1])
m13 = P(x0 , x0, c, n).diff(c[2])
# coefficients of c_i evaluated at (x_i - h/2) / 2
# x0_h_n1 = (x0-(h/2)/2)
```

```
h_2 = h/2
m21 = P(x0-h_2, x0, c, n).diff(c[0])
m22 = P(x0-h_2, x0, c, n).diff(c[1])
m23 = P(x0-h_2, x0, c, n).diff(c[2])
# coefficients of c_i evaluated at (x_i + h/2) / 2
# x0_h_p1 = (x0+(h/2)/2)
m31 = P(x0+h_2, x0, c, n).diff(c[0])
m32 = P(x0+h_2, x0, c, n).diff(c[1])
m33 = P(x0+h_2, x0, c, n).diff(c[2])
# matrix of the coefficients is 3x3 in this case
M = Matrix([[m11, m12, m13], [m21, m22, m23], [m31, m32, m33]])


# matrix of the function values...actually a vector of right hand sides
R = Matrix([[Fi], [Fim1], [Fip1]])
# matrix form of the three equations for the c_i is M*X = R
# solution directly inverting the 3x3 matrix M:
X =  M.inv() * R
# note that all three coefficients make up the solution
# the first derivative is coefficient c_1 which is X[1].
print("The finite volume third-order accurate approximation for the firs
t derivative is: ")
print(together(X[1]))


d = symbols('c:8')
dfdxcheck1 = (P(x0+h, x0, d, 8) - P(x0-h, x0, d, 8))/(2*h)
print(simplify(dfdxcheck1)) # so the appropriate cancellation of terms i
nvolving `h` happens

dfdxcheck2 = (P(x0+h_2, x0, d, 8) - P(x0-h_2, x0, d, 8))/(2*h_2)
print(simplify(dfdxcheck2)) # so the appropriate cancellation of terms i
nvolving `h` happens
```

```
The finite difernce third-order accurate approximation for the first de
rivative is:
(F_{i+1} - F_{i-1})/(2*h)
The finite volume third-order accurate approximation for the first deri
vative is:
(F_{i+1/2} - F_{i-1/2})/h
c1 + c3*h**2/6 + c5*h**4/120 + c7*h**6/5040
c1 + c3*h**2/24 + c5*h**4/1920 + c7*h**6/322560
```

Q1 answer

From the print statement above we can see that the finite difference and finite volume approximations differ for second-order accurate approximation for the first derivative.

The finite difernce thir-order accurate approximation for the first derivative:

$$\frac{2c_1 h + \frac{c_3 h^3}{3} + \frac{c_5 h^5}{60} + \frac{c_7 h^7}{2520}}{2h}$$

The finite volume third-order accurate approximation for the first derivative:

$$\frac{c_1 h + \frac{c_3 h^3}{24} + \frac{c_5 h^5}{1920} + \frac{c_7 h^7}{322560}}{h}$$

Question 2

(/15) Derive the 2nd-order centred difference form for the 3-dimensional Poisson's equation using the finite-volume method:

$$\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} + \frac{\partial^2 T}{\partial z^2} = S$$

where $T = T(x, y, z)$ is the temperature, and $S = S(x, y, z)$ is the source/sink term. Assume the mesh is structured and rectangular, with CV dimensions $\Delta x \times \Delta y \times \Delta z$

In [1]:
```
from devito import Grid, Function, TimeFunction, Operator, configuration
, Eq, solve

nx = 50
ny = 50
nz = 50
nt  = 100

# Silence the runtime performance logging
configuration['log-level'] = 'ERROR'

# Now with Devito we will turn `T` into `TimeFunction` object
# to make all the buffer switching implicit
grid = Grid(shape=(nx, ny, nz))
T = Function(name='T', grid=grid, space_order=2)
Td = Function(name='Td', grid=grid, space_order=2)
T.data[:] = 0.
Td.data[:] = 0.

# Initialise the source term `S`
S = Function(name='S', grid=grid)
S.data[:] = 0.
S.data[int(nx / 4), int(ny / 4)]  = 100
S.data[int(3 * nx / 4), int(3 * ny / 4)] = -100

# Create Laplace equation base on `Td`
eq = Eq(Td.laplace, S, subdomain=grid.interior)
# Let SymPy solve for the central stencil point
stencil = solve(eq, Td)
# Now we let our stencil populate our second buffer `T`
eq_stencil = Eq(T, stencil)
print(eq_stencil)
```

```
Eq(T(x, y, z), -0.5*(h_x**2*h_y**2*h_z**2*S(x, y, z) - h_x**2*h_y**2*Td
(x, y, z - h_z) - h_x**2*h_y**2*Td(x, y, z + h_z) - h_x**2*h_z**2*Td(x,
y - h_y, z) - h_x**2*h_z**2*Td(x, y + h_y, z) - h_y**2*h_z**2*Td(x - h_
x, y, z) - h_y**2*h_z**2*Td(x + h_x, y, z))/(h_x**2*h_y**2 + h_x**2*h_z
**2 + h_y**2*h_z**2))
```

Q2 answer

Stencil of the 2nd-order centered difference form for the 3-D Poisson's equation using the finite-volume method:

$$T(x, y, z) = -\frac{0.5\left(h_x^2 h_y^2 h_z^2 S(x, y, z) - h_x^2 h_y^2 \operatorname{Td}(x, y, z - h_z) - h_x^2 h_y^2 \operatorname{Td}(x, y, z + h_z) - h_x^2 h_z^2 \operatorname{Td}\left(x, y - h\right. \\ \left. - h_y^2 h_z^2 \operatorname{Td}(x + h_x, y, z)\right)}{h_x^2 h_y^2 + h_x^2 h_z^2 + h_y^2 h_z^2}$$

In [ ]: