

# 1 Radial Derivative Scheme in ASH

We approximate the derivative of a function at some point  $x_i$  in ASH as a linear combination of some number of neighboring points. We refer to the collection of neighboring points that enter into this linear combination as the *stencil* for point  $x_i$ . In general, a larger stencil size yields a derivative with a higher formal accuracy. We have chosen a five-point stencil, centered on each  $x_i$ , as the ASH default, but the code has been generalized to allow any stencil size to be attempted.

For a five point stencil, the derivative of some function  $f(x)$  at point  $x_i$  is approximated as

$$\frac{\partial^n f}{\partial x^n}|_{x_i} = d_{n1} f(x_{i-2}) + d_{n2} f(x_{i-1}) + d_{n3} f(x_i) + d_{n4} f(x_{i+1}) + d_{n5} f(x_{i+2}), \quad (1)$$

where the  $d_{ij}$ 's are a set of coefficients determined at initialization time. On a uniform grid, a five-point stencil corresponds to first and second derivatives with 4th-order accuracy and a third derivative with 2nd-order accuracy. A fourth derivative (or higher) is never taken in the current version of ASH.

## 1.1 Determination of the $d_{ij}$ 's

The  $d_{ij}$  coefficients of equation 1 may be determined by Taylor expanding about the point  $x_i$  to approximate  $f$  at the other points in the stencil. By doing so, we generate a set of linear equations that can be solved to yield the  $d_{ij}$ 's. To begin with, let us define the distances between the point  $x_i$  and the other points in the stencil as

$$\begin{aligned} \Delta_{m2} &= x_{i-2} - x_i \\ \Delta_{m1} &= x_{i-1} - x_i \\ \Delta_1 &= x_{i+1} - x_i \\ \Delta_2 &= x_{i+2} - x_i \end{aligned} \quad (2)$$

We now adopt the following notational convention

$$f_{n,i} \equiv \frac{\partial^n f}{\partial x^n}|_{x_i}, \quad (3)$$

and Taylor expand about  $x_i$  to arrive at the following 5 equations:

$$\begin{aligned} f_{i-2} &= f_{0,i} + \Delta_{m2} f_{1,i} + \frac{\Delta_{m2}^2}{2} f_{2,i} + \frac{\Delta_{m2}^3}{3!} f_{3,i} + \frac{\Delta_{m2}^4}{4!} f_{4,i} + \dots \\ f_{i-1} &= f_{0,i} + \Delta_{m1} f_{1,i} + \frac{\Delta_{m1}^2}{2} f_{2,i} + \frac{\Delta_{m1}^3}{3!} f_{3,i} + \frac{\Delta_{m1}^4}{4!} f_{4,i} + \dots \\ f_i &= f_{0,i} \\ f_{i+1} &= f_{0,i} + \Delta_1 f_{1,i} + \frac{\Delta_1^2}{2} f_{2,i} + \frac{\Delta_1^3}{3!} f_{3,i} + \frac{\Delta_1^4}{4!} f_{4,i} + \dots \\ f_{i+2} &= f_{0,i} + \Delta_2 f_{1,i} + \frac{\Delta_2^2}{2} f_{2,i} + \frac{\Delta_2^3}{3!} f_{3,i} + \frac{\Delta_2^4}{4!} f_{4,i} + \dots \end{aligned} \quad (4)$$

By expanding only out to fourth order (our stencil size minus 1), we arrive at five equations with five unknowns (derivatives zero through four). The central equation is trivial, but we

will include it to keep the representation of the system clear. We can write this system of equations in matrix form as

$$\begin{pmatrix} 1 & \Delta_{m2} & (1/2!)\Delta_{m2}^2 & (1/3!)\Delta_{m2}^3 & (1/4!)\Delta_{m2}^4 \\ 1 & \Delta_{m1} & (1/2!)\Delta_{m1}^2 & (1/3!)\Delta_{m1}^3 & (1/4!)\Delta_{m1}^4 \\ 1 & 0 & 0 & 0 & 0 \\ 1 & \Delta_{p1} & (1/2!)\Delta_{p1}^2 & (1/3!)\Delta_{p1}^3 & (1/4!)\Delta_{p1}^4 \\ 1 & \Delta_{p2} & (1/2!)\Delta_{p2}^2 & (1/3!)\Delta_{p2}^3 & (1/4!)\Delta_{p2}^4 \end{pmatrix} \begin{pmatrix} f_{0,i} \\ f_{1,i} \\ f_{2,i} \\ f_{3,i} \\ f_{4,i} \end{pmatrix} = \begin{pmatrix} f_{i-2} \\ f_{i-1} \\ f_i \\ f_{i+1} \\ f_{i+2} \end{pmatrix}, \quad (5)$$

or more compactly as

$$\mathbf{N} \mathbf{f}_n = \mathbf{f}, \quad (6)$$

Where  $\mathbf{N}$  is the matrix of  $\Delta$ 's,  $\mathbf{f}_n$  is the derivative vector, and  $\mathbf{f}$  is the function vector. If we denote  $\mathbf{N}^{-1}$  as  $\mathbf{D}$ , we have

$$\mathbf{f}_n = \mathbf{D} \mathbf{f}, \quad (7)$$

and we see that the  $d_{ij}$ 's from Equation 1 are the elements of the inverse matrix  $\mathbf{D}$ .

In ASH FD, at initialization time, the inverse matrix  $\mathbf{D}$  is solved for at each grid point and at each derivative order (in case different stencil sizes/derivative accuracies are requested). Thus for a  $N$  grid points, we do  $3N$  inversions, but these are small matrices, and the initialization is generally very fast. More specifics are given below for anyone who is interested (also see the subroutine `gen_coefs` in `Derivatives.F`).

## 1.2 Preconditioning of the Derivative Matrix

The matrix in equation 5 has columns with widely disparate values. For a solar convection zone simulation with a uniform radial grid with 200 points, the  $\Delta$ 's are of order  $10^8$ , and the fifth column of the matrix will have values of order  $10^{32}$  times larger than the first column. To avoid any numerical precision issues in the matrix inversion, we recast the system of equations as

$$\begin{pmatrix} 1 & (\Delta_{m2}/\delta) & (\Delta_{m2}/\delta)^2 & (\Delta_{m2}/\delta)^3 & (\Delta_{m2}/\delta)^4 \\ 1 & (\Delta_{m1}/\delta) & (\Delta_{m1}/\delta)^2 & (\Delta_{m1}/\delta)^3 & (\Delta_{m1}/\delta)^4 \\ 1 & 0 & 0 & 0 & 0 \\ 1 & (\Delta_{p1}/\delta) & (\Delta_{p1}/\delta)^2 & (\Delta_{p1}/\delta)^3 & (\Delta_{p1}/\delta)^4 \\ 1 & (\Delta_{p2}/\delta) & (\Delta_{p2}/\delta)^2 & (\Delta_{p2}/\delta)^3 & (\Delta_{p2}/\delta)^4 \end{pmatrix} \begin{pmatrix} f_{0,i} \\ \delta f_{1,i} \\ (\delta^2/2!)f_{2,i} \\ (\delta^3/3!)f_{3,i} \\ (\delta^4/4!)f_{4,i} \end{pmatrix} = \begin{pmatrix} f_{i-2} \\ f_{i-1} \\ f_i \\ f_{i+1} \\ f_{i+2} \end{pmatrix}, \quad (8)$$

where  $\delta$ , the average over distances *within the stencil*, is defined as

$$\delta = \frac{|\Delta_{m2}| + |\Delta_{m1}| + |\Delta_{p1}| + |\Delta_{p2}|}{4}. \quad (9)$$

For a uniform grid, this yields

$$\mathbf{N} = \begin{pmatrix} 1 & -4/3 & 16/9 & -64/27 & 256/81 \\ 1 & -2/3 & 4/9 & -8/27 & 16/81 \\ 1 & 0 & 0 & 0 & 0 \\ 1 & 2/3 & 4/9 & 8/27 & 16/81 \\ 1 & 4/3 & 16/9 & 64/27 & 256/81 \end{pmatrix} \approx \begin{pmatrix} 1 & -1.33 & 1.78 & -2.37 & 3.16 \\ 1 & -0.67 & 0.44 & -0.30 & 0.20 \\ 1 & 0 & 0 & 0 & 0 \\ 1 & 0.67 & 0.44 & 0.30 & 0.20 \\ 1 & 1.33 & 1.78 & 2.37 & 3.16 \end{pmatrix}, \quad (10)$$

and all matrix elements are within a factor of 10 of each other. Once the inverse matrix  $\mathbf{D}$  is found, we need to multiply the  $n$ th row by  $n!/\delta^n$  before extracting the  $d_{ij}$ 's. The subroutine `gen_coefs` (`Derivatives.F`) uses the algorithm above to compute the  $d_{ij}$ 's.

### 1.3 Non-Symmetric Stencils

A centered stencil is not possible for points  $N_R-1$  and 2, so we use a 4-point stencil for these points. The stencil includes the boundary point, the point itself, and the 2 interior neighbor points. We find the coefficients for point  $x_2$  by solving:

$$\begin{aligned}
 f_{i-1} &= f_{0,i} + \Delta_{m1} f_{1,i} + \frac{\Delta_{m1}^2}{2} f_{2,i} + \frac{\Delta_{m1}^3}{3!} f_{3,i} + \dots \\
 f_i &= f_{0,i} \\
 f_{i+1} &= f_{0,i} + \Delta_1 f_{1,i} + \frac{\Delta_1^2}{2} f_{2,i} + \frac{\Delta_1^3}{3!} f_{3,i} + \dots \\
 f_{i+2} &= f_{0,i} + \Delta_2 f_{1,i} + \frac{\Delta_2^2}{2} f_{2,i} + \frac{\Delta_2^3}{3!} f_{3,i} + \dots
 \end{aligned} \tag{11}$$

and those for point  $x_{N_R-1}$  with

$$\begin{aligned}
 f_{i-2} &= f_{0,i} + \Delta_{m2} f_{1,i} + \frac{\Delta_{m2}^2}{2} f_{2,i} + \frac{\Delta_{m2}^3}{3!} f_{3,i} + \dots \\
 f_{i-1} &= f_{0,i} + \Delta_{m1} f_{1,i} + \frac{\Delta_{m1}^2}{2} f_{2,i} + \frac{\Delta_{m1}^3}{3!} f_{3,i} + \dots \\
 f_i &= f_{0,i} \\
 f_{i+1} &= f_{0,i} + \Delta_1 f_{1,i} + \frac{\Delta_1^2}{2} f_{2,i} + \frac{\Delta_1^3}{3!} f_{3,i} + \dots
 \end{aligned} \tag{12}$$

Boundary points are handled using forward/backward differences. By default, the boundary accuracy is set to 2nd order for the first and second derivatives. This involves the boundary point and two interior points for the first derivative, but three interior points for the second derivative. To calculate the first derivative at point  $x_{N_R}$ , our system of equations becomes

$$\begin{aligned}
 f_{i-2} &= f_{0,i} + \Delta_{m2} f_{1,i} + \frac{\Delta_{m2}^2}{2} f_{2,i} + \dots \\
 f_{i-1} &= f_{0,i} + \Delta_{m1} f_{1,i} + \frac{\Delta_{m1}^2}{2} f_{2,i} + \dots \\
 f_i &= f_{0,i}
 \end{aligned} \tag{13}$$

For the second derivative on a boundary, we need one additional point, and the appropriate system of equations to solve at point  $x_1$  would be

$$\begin{aligned}
 f_i &= f_{0,i} \\
 f_{i+1} &= f_{0,i} + \Delta_1 f_{1,i} + \frac{\Delta_1^2}{2} f_{2,i} + \frac{\Delta_1^3}{3!} f_{3,i} + \dots \\
 f_{i+2} &= f_{0,i} + \Delta_2 f_{1,i} + \frac{\Delta_2^2}{2} f_{2,i} + \frac{\Delta_2^3}{3!} f_{3,i} + \dots \\
 f_{i+3} &= f_{0,i} + \Delta_3 f_{1,i} + \frac{\Delta_3^2}{2} f_{2,i} + \frac{\Delta_3^3}{3!} f_{3,i} + \dots
 \end{aligned} \tag{14}$$

## 2 Additional Notes

### 2.1 The Nonlinear 2nd Derivative

The radial advection of  $\phi$ - and  $\theta$ -momentum involves a term that looks like

$$(\nabla_H^2 W) \frac{\partial^2 W}{\partial r^2}. \quad (15)$$

For reasons unknown, this term leads to instabilities when the second derivative is calculated using coefficients generated with the Taylor expansion methods described above. Fortunately, a double application of the first derivative operator *is* stable (for a variety of stencil sizes). We can generate a separate set of second derivative coefficients for use with these advective terms through a double application of the first derivative operator. These coefficients are stored in the array “dd\_coefs\_nl” and are currently only used in the calculation of  $\partial v_\theta / \partial dr$  and  $\partial v_\phi / \partial dr$  (see the subroutine radial\_velocity\_derivatives in Derivatives.F). These terms only appear in the advective and viscous heating terms. Everywhere else in the code uses the standard second derivative coefficients. To use the special coefficients, set the keyword nl\_deriv\_method in the controls namelist to 3. By default, this keyword is 2, and the 1st derivative is called twice for these terms. This is a little more communication, but it allows you to use fewer radial points per cpu (5 rather than 9 by default).