Method of undetermined coefficients

Suppose we want to derive a finite difference approximation to $u'(\bar{x})$ based on some given set of points. We can use Taylor series to derive an appropriate formula, using the *method of undetermined coefficients*.

Derivation

Consider a formula of the form

$$D_2 u(\bar{x}) = au(\bar{x}+2h) + bu(\bar{x}) + cu(\bar{x}-h).$$

We aim to determine a, b, and c to give the best possible accuracy for D_2.

From Taylor series, we have

$$\begin{aligned} u(\bar{x}+2h) &= u(\bar{x}) + 2hu'(\bar{x}) + \frac{4h^2}{2}u''(\bar{x}) + \frac{8h^3}{6}u'''(\bar{x}) + O(h^4), \\ u(\bar{x}-h) &= u(\bar{x}) - hu'(\bar{x}) + \frac{h^2}{2}u''(\bar{x}) - \frac{h^3}{6}u'''(\bar{x}) + O(h^4). \end{aligned}$$

Then,

$$D_2 u(\bar{x}) = \overbrace{(a+b+c)}^{=0} u(\bar{x}) + \overbrace{h(2a-c)}^{=1} u'(\bar{x}) + \overbrace{h^2(4a+c)}^{=0} u''(\bar{x}) + \frac{h^3}{6}(8a-c)u'''(\bar{x}) + O(h^4).$$

(Why only three conditions? This needs to agree with $u'(\bar{x})$ to high order. We might like to require even higher order coefficients to be zero as well, but since there are only three unknowns, we cannot satisfy more than three conditions.)

We thus require

$$\begin{cases} a+b+c=0, & \textcircled{1} \\ 2a-c=1/h, & \textcircled{2} \\ 4a+c=0, & \textcircled{3} \end{cases}$$

Solving this systems of equations ($\textcircled{2}+\textcircled{3}$), we have

$$a = \frac{1}{6h}, \quad b = \frac{3}{6h}, \quad c = -\frac{4}{6h}.$$

Hence,

$$D_2 u(\bar{x}) = \frac{u(\bar{x}+2h) + 3u(\bar{x}) - 4u(\bar{x}-h)}{6h} = u'(\bar{x}) + \frac{h^2}{3}u'''(\bar{x}) + O(h^3)$$

is second-order accurate.

See `deriv.py` for testing the accuracy of the second-order finite-difference formula.

A general formulation for finite-difference formula

The previous method can be extended to compute the finite difference coefficents for approximating $u^{(k)}(\bar{x})$, the k-th derivative of $u(x)$ evaluated at $\bar{x}$, based on an arbitaray stecil of $n \geq k+1$ points $x_1, x_2, \ldots, x_n$ where $x_i = \bar{x} + s_i h$. Assume $u(x)$ is sufficiently smooth, so the Taylor expansions below are valid.

Derivation

Consider a formula for $u'(\bar{x})$ using shifts $\{s_1, s_2, \ldots, s_n\}$. Evaluate the function at points

$$u(\bar{x} + s_i h) = u(\bar{x}) + h s_i u'(\bar{x}) + \frac{h^2 s_i^2}{2} u''(\bar{x}) + \ldots + \frac{h^{n-1} s_i^{n-1}}{(n-1)!} u^{(n-1)}(\bar{x}) + O(h^n).$$

Search for a general formula,

$$Du(\bar{x}) = \frac{1}{h} \sum_{i=1}^{n} r_i u(\bar{x} + s_i h),$$

with coefficients of

$$\begin{aligned}
\frac{1}{h} u(\bar{x}) : \sum_{i=1}^{n} r_i & = 0 \\
u'(\bar{x}) : \sum_{i=1}^{n} r_i s_i & = 1 \\
\frac{h}{2} u''(\bar{x}) : \sum_{i=1}^{n} r_i s_i^2 & = 0 \\
& \vdots \\
\frac{h^{n-2}}{(n-1)!} u^{(n-1)}(\bar{x}) : \sum_{i=1}^{n} r_i s_i^{n-1} & = 0
\end{aligned}$$

Find a solution to the matrix system

$$\underbrace{\begin{bmatrix} 1 & 1 & 1 & \cdots & 1 \\ s_1 & s_2 & s_3 & \cdots & s_n \\ s_1^2 & s_2^2 & s_3^2 & \cdots & s_n^2 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ s_1^{n-1} & s_2^{n-1} & s_3^{n-1} & \cdots & s_n^{n-1} \end{bmatrix}}_{\text{Transpose of the Vandermonde matrix}} \begin{bmatrix} r_1 \\ r_2 \\ r_3 \\ \vdots \\ r_n \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}.$$

See `deriv_gen.py` for computing the coefficients with the Vandermonde matrix.

Quick review of the Vandermonde matrix

For a given set of points $\{x_1, x_2, \ldots, x_n\}$, the Vandermonde matrix is

$$V = \begin{bmatrix} 1 & x_1 & x_1^2 & \cdots & x_1^{n-1} \\ 1 & x_2 & x_2^2 & \cdots & x_2^{n-1} \\ 1 & x_3 & x_3^2 & \cdots & x_3^{n-1} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_{n-1} & x_{n-1}^2 & \cdots & x_{n-1}^{n-1} \\ 1 & x_n & x_n^2 & \cdots & x_n^{n-1} \end{bmatrix}.$$

The Vandermonde matrix is useful for polynomial fitting and linear least square problems. (See AM205 Unit 1.) Note that Python uses an alternative convention where the columns are reversed.