

Numerical Differentiation

using Finite Difference and
Complex-Step Approximations

Tamas Kis | tamas.a.kis@outlook.com

Tamas Kis

<https://tamaskis.github.io>

Copyright © 2021 Tamas Kis.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.



LICENSE

Contents

Contents	iii
----------	-----

List of Algorithms	v
--------------------	---

I Definitions

1	Differentiation	2
1.1	Derivative of a Univariate, Scalar-Valued Function	2
1.2	Derivative of a Univariate, Vector-Valued Function	2
1.3	Partial Derivative of a Multivariate, Scalar-Valued Function	3
1.4	Partial Derivative of a Multivariate, Vector-Valued Function	3
1.5	Gradient of a Multivariate, Scalar-Valued Function	4
1.6	Directional Derivative of a Multivariate, Scalar-Valued Function	4
1.7	Jacobian of a Multivariate, Vector-Valued Function	5
1.8	Hessian of a Multivariate, Scalar-Valued Function	5
2	Finite Difference Approximations	7
2.1	Forward Difference Approximation	7
2.2	Backward Difference Approximation	8
2.3	Central Difference Approximation	9
2.4	Choosing a Step Size	11
3	The Complex-Step Approximation	12
3.1	Definition	12
3.2	Choosing a Step Size	13
3.3	Complexification and Transposes	13
3.3.1	Transposes	13
3.3.2	“Complexified” Functions Included in the <i>Numerical Differentiation Toolbox</i>	13
3.3.3	idot	14
3.3.4	imod	14
3.3.5	inorm	14
3.4	Limitations	14
3.4.1	Higher-Order Derivatives	15
3.4.2	Functions That Yield Incorrect Results	15
4	Generalizing the Approximations to Higher Dimensions	16
4.1	Derivatives of Vector-Valued Functions	16
4.2	Partial Derivatives of Scalar-Valued Functions	17
4.3	Partial Derivatives of Vector-Valued Functions	17
4.4	Summary	18

II Implementation

5	Numerical Differentiation Using the Forward Difference Approximation	21
5.1	Derivatives	21
5.2	Partial Derivatives	22
5.3	Gradients	23
5.4	Directional Derivatives	24
5.5	Jacobians	25
5.6	Hessians	27
6	Numerical Differentiation Using the Central Difference Approximation	30
6.1	Derivatives	30
6.2	Partial Derivatives	31
6.3	Gradients	32
6.4	Directional Derivatives	33
6.5	Jacobians	35
6.6	Hessians	36
7	Numerical Differentiation Using the Complex-Step Approximation	40
7.1	Derivatives	40
7.2	Partial Derivatives	41
7.3	Gradients	42
7.4	Directional Derivatives	43
7.5	Jacobians	44
7.6	Hessians	45
	References	47

List of Algorithms

Forward Difference Approximations

Algorithm 1	<code>fderivative</code>	Derivative of a univariate, vector-valued function using the forward difference approximation	21
Algorithm 2	<code>fpartial</code>	Partial derivative of a multivariate, vector-valued function using the forward difference approximation	22
Algorithm 3	<code>fgradient</code>	Gradient of a multivariate, scalar-valued function using the forward difference approximation	23
Algorithm 4	<code>fdirectional</code>	Directional derivative of a multivariate, scalar-valued function using the forward difference approximation	24
Algorithm 5	<code>fjacobian</code>	Jacobian matrix of a multivariate, vector-valued function using the forward difference approximation	26
Algorithm 6	<code>fhessian</code>	Hessian matrix of a multivariate, scalar-valued function using the forward difference approximation	27

Central Difference Approximations

Algorithm 7	<code>cderivative</code>	Derivative of a univariate, vector-valued function using the central difference approximation	30
Algorithm 8	<code>cpartial</code>	Partial derivative of a multivariate, vector-valued function using the central difference approximation	31
Algorithm 9	<code>cgradient</code>	Gradient of a multivariate, scalar-valued function using the central difference approximation	32
Algorithm 10	<code>cdirectional</code>	Directional derivative of a multivariate, scalar-valued function using the central difference approximation	34
Algorithm 11	<code>cjacobian</code>	Jacobian matrix of a multivariate, vector-valued function using the central difference approximation	35
Algorithm 12	<code>chessian</code>	Hessian matrix of a multivariate, scalar-valued function using the central difference approximation	37

Complex-Step Approximations

Algorithm 13	<code>iderivative</code>	Derivative of a univariate, vector-valued function using the complex-step approximation	40
--------------	--------------------------	---	----

Algorithm 14	<code>ipartial</code>	Partial derivative of a multivariate, vector-valued function using the complex-step approximation	41
Algorithm 15	<code>igradient</code>	Gradient of a multivariate, scalar-valued function using the complex-step approximation	42
Algorithm 16	<code>idirectional</code>	Directional derivative of a multivariate, scalar-valued function using the complex-step approximation	43
Algorithm 17	<code>ijacobian</code>	Jacobian matrix of a multivariate, vector-valued function using the complex-step approximation	44
Algorithm 18	<code>ihessian</code>	Hessian matrix of a multivariate, scalar-valued function using the complex-step and central difference approximations	46

PART I

Definitions

Differentiation

1.1 Derivative of a Univariate, Scalar-Valued Function

Consider a univariate, scalar-valued function $f : \mathbb{R} \rightarrow \mathbb{R}$. Its derivative with respect to $x \in \mathbb{R}$ is defined as

$$\frac{df}{dx} = \lim_{h \rightarrow 0} \left[\frac{f(x+h) - f(x)}{h} \right]$$

The derivative of f with respect to x , *evaluated at* the point $x = x_0$, is then

$$\left. \frac{df}{dx} \right|_{x=x_0} = \lim_{h \rightarrow 0} \left[\frac{f(x_0+h) - f(x_0)}{h} \right] \quad (1.1)$$

1.2 Derivative of a Univariate, Vector-Valued Function

Consider a univariate, vector-valued function $\mathbf{f} : \mathbb{R} \rightarrow \mathbb{R}^m$. Its derivative with respect to $x \in \mathbb{R}$ is defined as [17, pp. 895-896]

$$\frac{d\mathbf{f}}{dx} = \lim_{h \rightarrow 0} \left[\frac{\mathbf{f}(x+h) - \mathbf{f}(x)}{h} \right]$$

The derivative of \mathbf{f} with respect to x , *evaluated at* the point $x = x_0$, is then

$$\left. \frac{d\mathbf{f}}{dx} \right|_{x=x_0} = \lim_{h \rightarrow 0} \left[\frac{\mathbf{f}(x_0+h) - \mathbf{f}(x_0)}{h} \right] \quad (1.2)$$

Alternatively, we can note that the function, \mathbf{f} , can be written as

$$\mathbf{f}(x) = \begin{bmatrix} f_1(x) \\ \vdots \\ f_m(x) \end{bmatrix}$$

The derivative of \mathbf{f} with respect to $x \in \mathbb{R}$, evaluated at the point $x = x_0$, is then

$$\left. \frac{d\mathbf{f}}{dx} \right|_{x=x_0} = \begin{bmatrix} \left. \frac{df_1}{dx} \right|_{x=x_0} \\ \vdots \\ \left. \frac{df_m}{dx} \right|_{x=x_0} \end{bmatrix} \quad (1.3)$$

The individual derivatives in Eq. (1.3) are defined using Eq. (1.1). In a more compact form, this can be expressed as the summation [18]

$$\left. \frac{d\mathbf{f}}{dx} \right|_{x=x_0} = \sum_{j=1}^m \mathbf{e}_j \left. \frac{df_j}{dx} \right|_{x=x_0} \quad (1.4)$$

1.3 Partial Derivative of a Multivariate, Scalar-Valued Function

Consider a multivariate, scalar-valued function $f : \mathbb{R}^n \rightarrow \mathbb{R}$. The partial derivative of f with respect to $x_k \in \mathbb{R}$ is defined as [14]

$$\frac{\partial f}{\partial x_k} = \lim_{h \rightarrow 0} \left[\frac{f(\mathbf{x} + h\mathbf{e}_k) - f(\mathbf{x})}{h} \right]$$

where \mathbf{e}_k is the k th standard basis vector.

$$\mathbf{e}_k = \begin{bmatrix} 0 \\ \vdots \\ 1 \\ \vdots \\ 0 \end{bmatrix} \leftarrow k\text{th element} \quad (1.5)$$

Note that \mathbf{e}_k is also the k th column of the $n \times n$ identity matrix, $\mathbf{I}_{n \times n}$.

The partial derivative of f with respect to x_k , *evaluated at the point* $\mathbf{x} = \mathbf{x}_0$, is then

$$\left. \frac{\partial f}{\partial x_k} \right|_{\mathbf{x}=\mathbf{x}_0} = \lim_{h \rightarrow 0} \left[\frac{f(\mathbf{x}_0 + h\mathbf{e}_k) - f(\mathbf{x}_0)}{h} \right] \quad (1.6)$$

1.4 Partial Derivative of a Multivariate, Vector-Valued Function

Consider a multivariate, vector-valued function $\mathbf{f} : \mathbb{R}^n \rightarrow \mathbb{R}^m$. Its partial derivative with respect to $x_k \in \mathbb{R}$ is defined as [15]

$$\frac{\partial \mathbf{f}}{\partial x_k} = \lim_{h \rightarrow 0} \left[\frac{\mathbf{f}(\mathbf{x} + h\mathbf{e}_k) - \mathbf{f}(\mathbf{x})}{h} \right]$$

The partial derivative of \mathbf{f} with respect to x_k , *evaluated at the point* $\mathbf{x} = \mathbf{x}_0$, is then

$$\left. \frac{\partial \mathbf{f}}{\partial x_k} \right|_{\mathbf{x}=\mathbf{x}_0} = \lim_{h \rightarrow 0} \left[\frac{\mathbf{f}(\mathbf{x}_0 + h\mathbf{e}_k) - \mathbf{f}(\mathbf{x}_0)}{h} \right] \quad (1.7)$$

Alternatively, we can note that the function, \mathbf{f} , can be written as

$$\mathbf{f}(\mathbf{x}) = \begin{bmatrix} f_1(\mathbf{x}) \\ \vdots \\ f_m(\mathbf{x}) \end{bmatrix}$$

The partial derivative of \mathbf{f} with respect to $x_k \in \mathbb{R}$, evaluated at the point $\mathbf{x} = \mathbf{x}_0$, is then

$$\left. \frac{\partial \mathbf{f}}{\partial x_k} \right|_{\mathbf{x}=\mathbf{x}_0} = \begin{bmatrix} \left. \frac{\partial f_1}{\partial x_k} \right|_{\mathbf{x}=\mathbf{x}_0} \\ \vdots \\ \left. \frac{\partial f_m}{\partial x_k} \right|_{\mathbf{x}=\mathbf{x}_0} \end{bmatrix} \quad (1.8)$$

The individual partial derivatives in Eq. (1.8) are defined using Eq. (1.6). In a more compact form, this can be expressed as the summation [18]

$$\left. \frac{\partial \mathbf{f}}{\partial x_k} \right|_{\mathbf{x}=\mathbf{x}_0} = \sum_{j=1}^m \mathbf{e}_j \left. \frac{\partial f_j}{\partial x_k} \right|_{\mathbf{x}=\mathbf{x}_0} \quad (1.9)$$

1.5 Gradient of a Multivariate, Scalar-Valued Function

Consider a multivariate, scalar-valued function $f : \mathbb{R}^n \rightarrow \mathbb{R}$. The gradient of f with respect to $\mathbf{x} \in \mathbb{R}^n$, evaluated at $\mathbf{x} = \mathbf{x}_0$, is defined as [3]

$$\mathbf{g}(\mathbf{x}_0) = \nabla f(\mathbf{x}_0) = \begin{bmatrix} \left. \frac{\partial f}{\partial x_1} \right|_{\mathbf{x}=\mathbf{x}_0} \\ \vdots \\ \left. \frac{\partial f}{\partial x_n} \right|_{\mathbf{x}=\mathbf{x}_0} \end{bmatrix} \quad (1.10)$$

The individual partial derivatives in Eq. (1.10) are defined using Eq. (1.6).

1.6 Directional Derivative of a Multivariate, Scalar-Valued Function

Consider a multivariate, scalar-valued function $f : \mathbb{R}^n \rightarrow \mathbb{R}$. The directional derivative of f with respect to $\mathbf{x} \in \mathbb{R}^n$ in the direction of $\mathbf{v} \in \mathbb{R}^n$ is defined as

$$\nabla_{\mathbf{v}} f(\mathbf{x}) = \lim_{h \rightarrow 0} \left[\frac{f(\mathbf{x} + h\mathbf{v}) - f(\mathbf{x})}{h} \right]$$

Alternatively, it can be computed as the inner product between the gradient of f with respect to \mathbf{x} and the direction, \mathbf{v} [8, p. 22].

$$\nabla_{\mathbf{v}} f(\mathbf{x}) = \nabla f(\mathbf{x})^T \mathbf{v}$$

The directional derivative of f with respect to \mathbf{x} in the direction of $\mathbf{v} \in \mathbb{R}^n$, *evaluated at the point* $\mathbf{x} = \mathbf{x}_0$, is then

$$\nabla_{\mathbf{v}} f(\mathbf{x}_0) = \nabla f(\mathbf{x}_0)^T \mathbf{v} \quad (1.11)$$

Note that gradients are discussed in Section 1.5.

1.7 Jacobian of a Multivariate, Vector-Valued Function

The Jacobian of a multivariate, vector-valued function $\mathbf{f} : \mathbb{R}^n \rightarrow \mathbb{R}^m$ with respect to $\mathbf{x} \in \mathbb{R}^n$ is defined as

$$\mathbf{J} = \frac{\partial \mathbf{f}}{\partial \mathbf{x}} = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \cdots & \frac{\partial f_1}{\partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_m}{\partial x_1} & \cdots & \frac{\partial f_m}{\partial x_n} \end{bmatrix}$$

The Jacobian can also be written in terms of its column vectors as [7]

$$\mathbf{J} = \begin{bmatrix} \frac{\partial \mathbf{f}}{\partial x_1} & \cdots & \frac{\partial \mathbf{f}}{\partial x_n} \end{bmatrix}$$

The Jacobian of \mathbf{f} with respect to \mathbf{x} , *evaluated at the point* $\mathbf{x} = \mathbf{x}_0$, is then

$$\mathbf{J}(\mathbf{x}_0) = \frac{\partial \mathbf{f}}{\partial \mathbf{x}} \Big|_{\mathbf{x}=\mathbf{x}_0} = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} \Big|_{\mathbf{x}=\mathbf{x}_0} & \cdots & \frac{\partial f_1}{\partial x_n} \Big|_{\mathbf{x}=\mathbf{x}_0} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_m}{\partial x_1} \Big|_{\mathbf{x}=\mathbf{x}_0} & \cdots & \frac{\partial f_m}{\partial x_n} \Big|_{\mathbf{x}=\mathbf{x}_0} \end{bmatrix} = \begin{bmatrix} \frac{\partial \mathbf{f}}{\partial x_1} \Big|_{\mathbf{x}=\mathbf{x}_0} & \cdots & \frac{\partial \mathbf{f}}{\partial x_n} \Big|_{\mathbf{x}=\mathbf{x}_0} \end{bmatrix} \quad (1.12)$$

The partial derivatives forming each column of the Jacobian are discussed in Section 1.4.

1.8 Hessian of a Multivariate, Scalar-Valued Function

The Hessian of a multivariate, scalar-valued function $f : \mathbb{R}^n \rightarrow \mathbb{R}$ with respect to $\mathbf{x} \in \mathbb{R}^n$, evaluated at $\mathbf{x} = \mathbf{x}_0$, is defined as

$$\mathbf{H} = \begin{bmatrix} \frac{\partial^2 f}{\partial x_1^2} & \frac{\partial^2 f}{\partial x_1 \partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_1 \partial x_n} \\ \frac{\partial^2 f}{\partial x_2 \partial x_1} & \frac{\partial^2 f}{\partial x_2^2} & \cdots & \frac{\partial^2 f}{\partial x_2 \partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 f}{\partial x_n \partial x_1} & \frac{\partial^2 f}{\partial x_n \partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_n^2} \end{bmatrix}$$

In a more compact form, we can write the (j, k) th element of the Hessian as

$$[\mathbf{H}]_{j,k} = \frac{\partial^2 f}{\partial x_j \partial x_k}$$

The Hessian of f with respect to \mathbf{x} , *evaluated at* the point $\mathbf{x} = \mathbf{x}_0$, is then

$$\mathbf{H}(\mathbf{x}_0) = \begin{bmatrix} \left. \frac{\partial^2 f}{\partial x_1^2} \right|_{\mathbf{x}=\mathbf{x}_0} & \left. \frac{\partial^2 f}{\partial x_1 \partial x_2} \right|_{\mathbf{x}=\mathbf{x}_0} & \cdots & \left. \frac{\partial^2 f}{\partial x_1 \partial x_n} \right|_{\mathbf{x}=\mathbf{x}_0} \\ \left. \frac{\partial^2 f}{\partial x_2 \partial x_1} \right|_{\mathbf{x}=\mathbf{x}_0} & \left. \frac{\partial^2 f}{\partial x_2^2} \right|_{\mathbf{x}=\mathbf{x}_0} & \cdots & \left. \frac{\partial^2 f}{\partial x_2 \partial x_n} \right|_{\mathbf{x}=\mathbf{x}_0} \\ \vdots & \vdots & \ddots & \vdots \\ \left. \frac{\partial^2 f}{\partial x_n \partial x_1} \right|_{\mathbf{x}=\mathbf{x}_0} & \left. \frac{\partial^2 f}{\partial x_n \partial x_2} \right|_{\mathbf{x}=\mathbf{x}_0} & \cdots & \left. \frac{\partial^2 f}{\partial x_n^2} \right|_{\mathbf{x}=\mathbf{x}_0} \end{bmatrix} \quad (1.13)$$

where the (j, k) th element is given by

$$[\mathbf{H}(\mathbf{x}_0)]_{j,k} = \left. \frac{\partial^2 f}{\partial x_j \partial x_k} \right|_{\mathbf{x}=\mathbf{x}_0} \quad (1.14)$$

Note that from Schwarz's theorem, we know

$$\therefore \frac{\partial^2 f}{\partial x_j \partial x_k} = \frac{\partial^2 f}{\partial x_k \partial x_j}$$

which implies that the Hessian is symmetric and satisfies the property [5]

$$[\mathbf{H}]_{j,k} = [\mathbf{H}]_{k,j} \quad (1.15)$$

Since the Hessian is symmetric, we only have to evaluate the derivatives in the upper triangle of the matrix (shown in red below) to form the full matrix:

$$\begin{bmatrix} H_{1,1} & H_{1,2} & \cdots & H_{1,n} \\ H_{2,1} & H_{2,2} & \cdots & H_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ H_{n,1} & H_{n,2} & \cdots & H_{n,n} \end{bmatrix}$$



Finite Difference Approximations

2.1 Forward Difference Approximation

Consider a univariate scalar-valued function $f : \mathbb{R} \rightarrow \mathbb{R}$. Recall that its Taylor series expansion about the point $x = a$ is given by

$$f(x) = \sum_{k=0}^{\infty} \frac{f^{(k)}(a)}{k!} (x-a)^k = f(a) + f'(a)(x-a) + \frac{1}{2}f''(a)(x-a)^2 + \frac{1}{6}f'''(a)(x-a)^3 + \dots$$

Let's replace x with $x+h$ and a with x . Then

$$\begin{aligned} f(x+h) &= f(x) + f'(x)[(x+h)-x] + \frac{1}{2}f''(x)[(x+h)-x]^2 + \dots \\ &= f(x) + hf'(x) + \frac{h^2}{2}f''(x) + \frac{h^3}{6}f'''(x) + \dots \end{aligned}$$

Solving for $f'(x)$,

$$\begin{aligned} hf'(x) &= f(x+h) - \left[f(x) + \frac{h^2}{2}f''(x) + \frac{h^3}{6}f'''(x) + \dots \right] \\ f'(x) &= \frac{f(x+h)}{h} - \frac{1}{h} \left[f(x) + \frac{h^2}{2}f''(x) + \frac{h^3}{6}f'''(x) + \dots \right] \\ &= \frac{f(x+h) - f(x)}{h} - \frac{h}{2}f''(x) - \frac{h^2}{6}f'''(x) - \dots \\ &= \frac{f(x+h) - f(x)}{h} + \mathcal{O}(h) \end{aligned}$$

From this, we can directly extract the approximation

$$f'(x) \approx \frac{f(x+h) - f(x)}{h}$$

The derivative of f with respect to x , *evaluated at* the point $x = x_0$, is then

$$\boxed{\left. \frac{df}{dx} \right|_{x=x_0} \approx \frac{f(x_0+h) - f(x_0)}{h}} \quad (2.1)$$

This approximation is known as the **forward difference approximation** since the additional point we are using to construct the approximation is *forward* of (i.e. greater than) the evaluation point, x_0 . Note that we can also obtain this directly from the definition of the derivative. If h is small, then

$$\frac{df}{dx} = \lim_{h \rightarrow 0} \left[\frac{f(x+h) - f(x)}{h} \right] \approx \frac{f(x+h) - f(x)}{h}$$

Since the error term associated with the forward difference approximation is $\mathcal{O}(h)$, the error in the approximation decreases linearly as h approaches 0. Therefore, the forward difference approximation is a first-order approximation.

The forward difference approximation can be visualized using the stencil shown in Fig. 2.1.

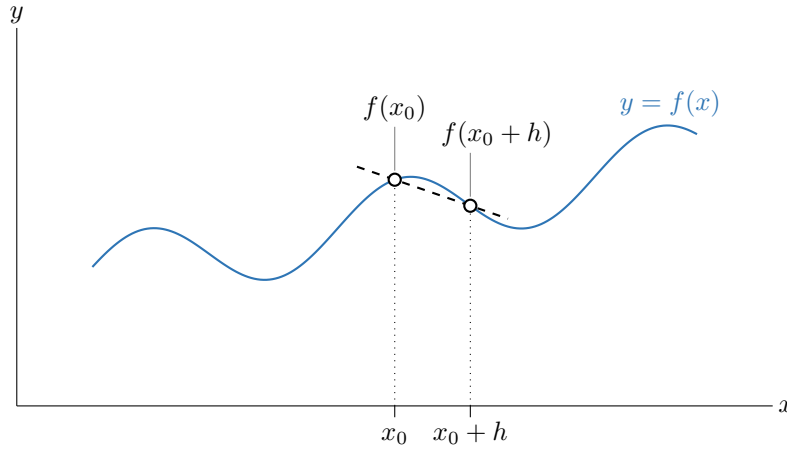


Figure 2.1: Forward difference approximation.

2.2 Backward Difference Approximation

Consider a univariate scalar-valued function $f : \mathbb{R} \rightarrow \mathbb{R}$. Recall that its Taylor series expansion about the point $x = a$ is given by

$$f(x) = \sum_{k=0}^{\infty} \frac{f^{(k)}(a)}{k!} (x-a)^k = f(a) + f'(a)(x-a) + \frac{1}{2}f''(a)(x-a)^2 + \frac{1}{6}f'''(a)(x-a)^3 + \dots$$

Let's replace x with $x-h$ and a with x . Then

$$\begin{aligned} f(x-h) &= f(x) + f'(x)[(x-h)-x] + \frac{1}{2}f''(x)[(x-h)-x]^2 + \dots \\ &= f(x) - hf'(x) + \frac{h^2}{2}f''(x) - \frac{h^3}{6}f'''(x) - \dots \end{aligned}$$

Solving for $f'(x)$,

$$hf'(x) = -f(x-h) + \left[f(x) - \frac{h^2}{2}f''(x) + \frac{h^3}{6}f'''(x) + \dots \right]$$

$$\begin{aligned}
f'(x) &= -\frac{f(x-h)}{h} + \frac{1}{h} \left[f(x) - \frac{h^2}{2} f''(x) + \frac{h^3}{6} f'''(x) + \dots \right] \\
&= \frac{f(x) - f(x-h)}{h} - \frac{h}{2} f''(x) + \frac{h^2}{6} f'''(x) + \dots \\
&= \frac{f(x) - f(x-h)}{h} + \mathcal{O}(h)
\end{aligned}$$

From this, we can directly extract the approximation

$$f'(x) \approx \frac{f(x) - f(x-h)}{h}$$

The derivative of f with respect to x , *evaluated at* the point $x = x_0$, is then

$$\left. \frac{df}{dx} \right|_{x=x_0} \approx \frac{f(x_0) - f(x_0-h)}{h} \quad (2.2)$$

This approximation is known as the **backward difference approximation** since the additional point we are using to construct the approximation is *backward* of (i.e. less than) the evaluation point, x_0 .

Since the error term associated with the backward difference approximation is $\mathcal{O}(h)$, the error in the approximation decreases linearly as h approaches 0. Therefore, the backward difference approximation is a first-order approximation.

The backward difference approximation can be visualized using the stencil shown in Fig. 2.2.

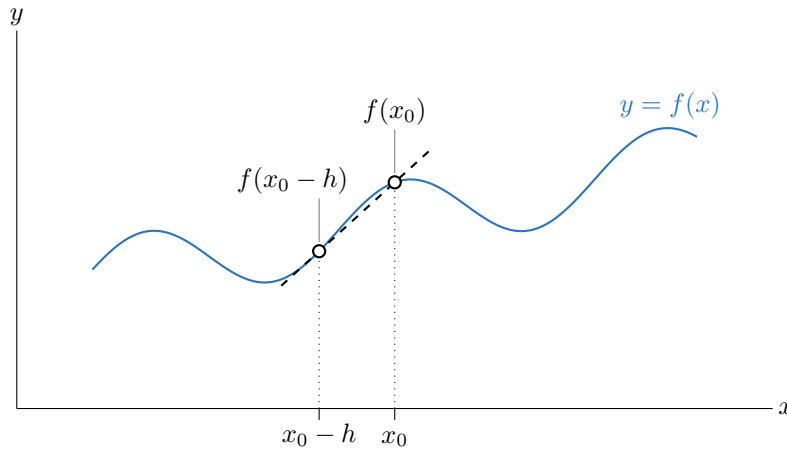


Figure 2.2: Backward difference approximation.

2.3 Central Difference Approximation

Consider a univariate scalar-valued function $f : \mathbb{R} \rightarrow \mathbb{R}$. From the Taylor series expansions developed in Sections 2.1 and 2.2, we can write

$$\begin{aligned}
f(x+h) &= f(x) + hf'(x) + \frac{h^2}{2} f''(x) + \frac{h^3}{6} f'''(x) + \dots \\
f(x-h) &= f(x) - hf'(x) + \frac{h^2}{2} f''(x) - \frac{h^3}{6} f'''(x) - \dots
\end{aligned}$$

It follows that

$$\begin{aligned} f(x+h) - f(x-h) &= \left[f(x) + hf'(x) + \frac{h^2}{2}f''(x) + \frac{h^3}{6}f'''(x) + \dots \right] \\ &\quad - \left[f(x) - hf'(x) + \frac{h^2}{2}f''(x) - \frac{h^3}{6}f'''(x) + \dots \right] \\ &= 2hf'(x) + \frac{h^3}{3}f'''(x) + \dots \end{aligned}$$

Solving for $f'(x)$,

$$\begin{aligned} f'(x) &= \frac{f(x+h) - f(x-h)}{2h} - \frac{h^2}{6}f'''(x) + \dots \\ &= \frac{f(x+h) - f(x-h)}{2h} + \mathcal{O}(h) \end{aligned}$$

From this, we can directly extract the approximation

$$f'(x) \approx \frac{f(x+h) - f(x-h)}{2h}$$

The derivative of f with respect to x , *evaluated at* the point $x = x_0$, is then

$$\left. \frac{df}{dx} \right|_{x=x_0} \approx \frac{f(x_0+h) - f(x_0-h)}{2h} \quad (2.3)$$

This approximation is known as the **central difference approximation** since the point at which we are evaluating the derivative, x_0 , is *centered* between the two points used to construct the approximation.

Since the error term associated with the forward difference approximation is $\mathcal{O}(h^2)$, the error in the approximation decreases quadratically as h approaches 0. Therefore, the central difference approximation is a second-order approximation.

The central difference approximation can be visualized using the stencil shown in Fig. 2.3.

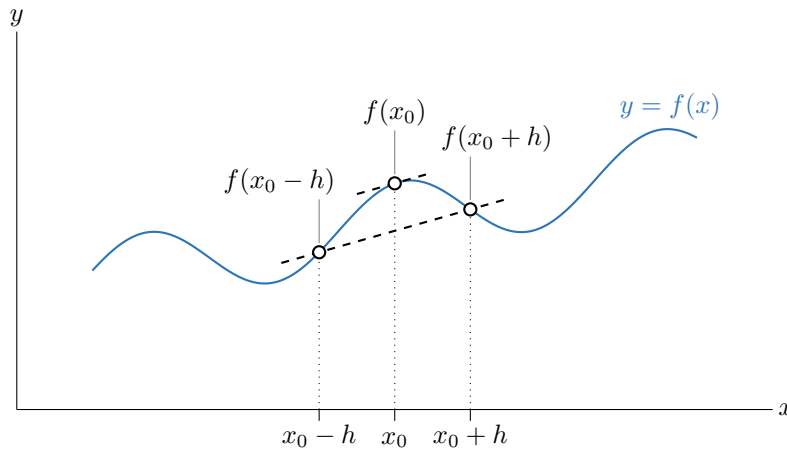


Figure 2.3: Central difference approximation.

2.4 Choosing a Step Size

To improve our derivative estimate, we want to reduce the step size, h , as much as possible. However, as h is reduced, the finite difference approximations become dominated by subtractive cancellation errors [10, pp. 229–230]. Once we decrease h beyond some lower bound, the finite difference approximations will actually start to get worse.

The machine zero, ε , is defined as the smallest positive number ε such that $1 + \varepsilon > 1$ when calculated using a computer). For double precision, $\varepsilon = 2^{-52} \approx 2.2 \times 10^{-16}$ [10, p. 55]. The optimal step size for the forward and backward difference approximations is approximately $\sqrt{\varepsilon}$, while the optimal step size for the central difference approximation is approximately $\varepsilon^{1/3}$ [10, p. 230]. These are the values used by default for h in the *Numerical Differentiation Toolbox*.

However, it is also helpful to scale the step size based on the value of x_0 (i.e. the point at which we are differentiating). [10, p. 231] defines h as the **relative step size** and suggests using the finite difference approximations

$$\left. \frac{df}{dx} \right|_{x=x_0} \approx \frac{f(x_0 + \Delta x) - f(x_0)}{\Delta x} \quad (2.4)$$

$$\left. \frac{df}{dx} \right|_{x=x_0} \approx \frac{f(x_0) - f(x_0 - \Delta x)}{\Delta x} \quad (2.5)$$

$$\left. \frac{df}{dx} \right|_{x=x_0} \approx \frac{f(x_0 + \Delta x) - f(x_0 - \Delta x)}{2\Delta x} \quad (2.6)$$

where the **absolute step size**, Δx , is defined as

$$\Delta x = h(1 + |x_0|) \quad (2.7)$$



The Complex-Step Approximation

3.1 Definition

Recall from Section 2.1 that the Taylor series expansion for a univariate, scalar-valued function $f : \mathbb{R} \rightarrow \mathbb{R}$ is given by

$$f(x+h) = f(x) + hf'(x) + \frac{h^2}{2}f''(x) + \frac{h^3}{6}f'''(x) + \dots$$

However, if we instead take a step, ih , in the imaginary direction (where $i = \sqrt{-1}$), then the Taylor series expansion is

$$\begin{aligned} f(x+ih) &= f(x) + ihf'(x) - \frac{h^2}{2}f''(x) - \frac{ih^3}{6}f'''(x) + \dots \\ &= \left[f(x) - \frac{h^2}{2}f''(x) + \dots \right] + i \left[hf'(x) - \frac{h^3}{6}f'''(x) + \dots \right] \end{aligned}$$

Taking the imaginary component of each side,

$$\text{Im}[f(x+ih)] = hf'(x) - \frac{h^3}{6}f'''(x) + \dots$$

Solving for $f'(x)$,

$$\begin{aligned} hf'(x) &= \text{Im}[f(x+ih)] + \frac{h^3}{6}f'''(x) + \dots \\ f'(x) &= \frac{\text{Im}[f(x+ih)]}{h} + \frac{h^2}{6}f'''(x) + \dots \\ &= \frac{\text{Im}[f(x+ih)]}{h} + \mathcal{O}(h^2) \end{aligned}$$

From this, we can directly extract the approximation

$$f'(x) \approx \frac{\text{Im}[f(x+ih)]}{h}$$

The derivative of f with respect to x , *evaluated at the point* $x = x_0$, is then [8, p. 25][11, 16]

$$\boxed{\left. \frac{df}{dx} \right|_{x=x_0} \approx \frac{\text{Im}[f(x+ih)]}{h}} \quad (3.1)$$

This approximation is known as the **complex-step approximation** since the additional point we are using to construct the approximation is *forward* of (i.e. greater than) the evaluation point, x_0 . Note that we can also obtain this directly from the definition of the derivative. If h is small, then

$$\frac{df}{dx} = \lim_{h \rightarrow 0} \left[\frac{f(x+h) - f(x)}{h} \right] \approx \frac{f(x+h) - f(x)}{h}$$

Since the error term associated with the complex-step approximation is $\mathcal{O}(h^2)$, the error in the approximation decreases quadratically as h approaches 0. Therefore, the complex-step approximation is a second-order approximation.

3.2 Choosing a Step Size

As noted in Cleve Moler’s blog post [13] on the topic, the complex-step approximation of a derivative generally converges to within machine zero (ϵ , see Section 2.4) of the true derivative at a step size of about $h \approx \sqrt{\epsilon}$ (due to the $\mathcal{O}(h^2)$ convergence). However, [10, p. 234] notes that a step size of 10^{-200} works well for double-precision functions. Therefore, for all complex-step approximations, the *Numerical Differentiation Toolbox* uses a step size of $h = 10^{-200}$.

3.3 Complexification and Transposes

Any function that you are trying to differentiate must be able use complex numbers as inputs. Luckily, in MATLAB, most functions can already take complex numbers as inputs. However, there are some special cases of functions where MATLAB will accept complex numbers of inputs, but differentiating them via the complex-step approximation will produce incorrect results. Functions that are known to not be suitable for use with the *Numerical Differentiation Toolbox* (through unit testing; test code is also included with the *Numerical Differentiation Toolbox*) are summarized in Section 3.4.2.

3.3.1 Transposes

It is also vital to note how matrix transposes should be taken when using the complex-step approximation. The transpose operation in MATLAB is typically performed using an apostrophe (`'`), but this is problematic because it actually performs the conjugate transpose (i.e. it also takes the complex conjugate of each element). Therefore, we must use a dot before the apostrophe (`.'`) to perform the non-conjugate transpose [12].

Vector and matrix transposes must be performed using the dot-apostrophe syntax (`.'`).

This also leads to issues with differentiating the standard `norm` and `dot` functions provided by MATLAB. Complexified version of the `norm` and `dot` functions are not included in [2], but we define them in Sections 3.3.3 and 3.3.5.

3.3.2 “Complexified” Functions Included in the Numerical Differentiation Toolbox

The *Numerical Differentiation Toolbox* includes the following functions that have been “complexified” so that they are compatible with the complex-step approximation:

- `iabs` (replaces `abs`)

- `iatan2` (replaces `atan2`)
- `iatan2d` (replaces `atan2d`)
- `idot` (replaces `dot`)
- `imax` (replaces `max` for the maximum of two numbers)
- `imin` (replaces `min` for the minimum of two numbers)
- `imod` (replaces `mod` for a complex dividend and real divisor)
- `inorm` (replaces `norm` for 2-norm of a vector)

The “complexifications” of `abs`, `atan2`, `atan2d`, `max`, and `min` are adapted from a Fortran implementation [2]. [2] also includes additional complexified functions that have not been included here. The remaining complexifications (of `dot`, `mod`, and `norm`) are discussed in Sections 3.3.3–3.3.5.

3.3.3 `idot`

For $\mathbf{x}, \mathbf{y} \in \mathbb{R}^n$, the dot product and 2-norm are defined as

$$\mathbf{x} \cdot \mathbf{y} = \mathbf{x}^T \mathbf{y}$$

However, MATLAB’s `dot` function assumes $\mathbf{x}, \mathbf{y} \in \mathbb{C}^n$ and uses

$$\mathbf{x} \cdot \mathbf{y} = \mathbf{x}^H \mathbf{y}$$

In the *Numerical Differentiation Toolbox*, `idot` (the “complexified” version of `dot`) is implemented using $\mathbf{x} \cdot \mathbf{y} = \mathbf{x}^T \mathbf{y}$.

3.3.4 `imod`

The modulo operator, `mod`, returns the remainder, r , of a/n . MATLAB implements this as

$$\text{mod}(a, n) = a - \left\lfloor \frac{a}{n} \right\rfloor n$$

where $\lfloor \cdot \rfloor$ represents the floor functions. While MATLAB’s `floor` option accepts complex-valued inputs, its `mod` function does not. Therefore, we simply define the `imod` function using the definition above.

Unit testing of the `imod` function is not included in the *Numerical Differentiation Toolbox*. However, it was tested against a finite-difference approximation when differentiating a (rather complicated) function that used the modulo operation with $n \in \mathbb{R}$. Therefore, for the *Numerical Differentiation Toolbox* implementation of `imod`, it should be noted that n should be real-valued.

3.3.5 `inorm`

For $\mathbf{x} \in \mathbb{R}^n$, 2-norm is defined as

$$\|\mathbf{x}\| = \sqrt{\mathbf{x}^T \mathbf{x}}$$

However, MATLAB’s `dot` function assumes $\mathbf{x} \in \mathbb{C}^n$ and uses

$$\|\mathbf{x}\| = \sqrt{\mathbf{x}^H \mathbf{x}}$$

In the *Numerical Differentiation Toolbox*, `inorm` (the “complexified” version of `norm` specifically for the 2-norm of vectors) is implemented using $\|\mathbf{x}\| = \sqrt{\mathbf{x}^T \mathbf{x}}$.

3.4 Limitations

First and foremost, the complex-step approximation is only intended for approximating the derivatives of *real*-valued functions (as evident by its definition in Section 3.1. However, there are additional limitations discussed in Sections 3.4.1 and 3.4.2 that should be noted.

3.4.1 Higher-Order Derivatives

A proposed extension of the complex-step approximation to second derivatives is

$$\left. \frac{df}{dx} \right|_{x=x_0} \approx \frac{2(f(x_0) - \operatorname{Re}[f(x_0 + ih)])}{h^2}$$

Unfortunately, unlike in the first derivative case, the second derivative approximation can introduce errors if the step size is made too small [9]. Therefore, when evaluating second derivatives while forming the Hessian matrix in Section 7.6, we will use a different approach that utilizes a hybrid of complex-step and central difference approximations.

Additionally, we cannot use nested calls on a complex-step differentiation algorithm to obtain higher-order derivatives. Consider trying to approximate a second derivative by nesting one complex-step approximation within another:

$$\left. \frac{df}{dx} \right|_{x=x_0} \approx \left. \frac{d}{dx} \right|_{x=x_0} \left[\frac{\operatorname{Im}[f(x_0 + ih)]}{h} \right] \approx \frac{\operatorname{Im} \left[\overbrace{\frac{\operatorname{Im}[f(x_0 + 2ih)]}{h}}^{\text{this term has no imaginary part}} \right]}{h}$$

Since the term in the bracket has no imaginary part, we would simply get

$$\left. \frac{df}{dx} \right|_{x=x_0} = 0$$

which is incorrect.

3.4.2 Functions That Yield Incorrect Results

The following scalar-valued functions were tested in `iderivative_test` in the *Numerical Differentiation Toolbox* for MATLAB and yield incorrect results:

- $\operatorname{arccsc}(x)$ for $x < -1$
- $\operatorname{arcsec}(x)$ for $x < -1$
- $\operatorname{arccoth}(x)$ for $0 < x < 1$
- $\operatorname{arctanh}(x)$ for $x > 1$
- $\operatorname{arcsech}(x)$ for $-1 < x < 0$
- $\operatorname{arccoth}(x)$ for $-1 < x < 0$
- $\operatorname{arccosh}(x)$ for $x < -1$
- $\operatorname{arctanh}(x)$ for $x < -1$

WARNING: These errors remain uncorrected in the *Numerical Differentiation Toolbox*.

“Complexified” versions of most functions can be found in [2] (as discussed in Section 3.3.2).

4

Generalizing the Approximations to Higher Dimensions

4.1 Derivatives of Vector-Valued Functions

Recall from Section 2.1 that the Taylor series expansion for a univariate, scalar-valued function $f : \mathbb{R} \rightarrow \mathbb{R}$ is given by

$$f(x+h) = f(x) + hf'(x) + \frac{h^2}{2}f''(x) + \frac{h^3}{6}f'''(x) + \cdots$$

Now, consider the case where we have a univariate, vector-valued function $\mathbf{f} : \mathbb{R} \rightarrow \mathbb{R}^m$. The Taylor series expansion of \mathbf{f} is

$$\mathbf{f}(x+h) = \mathbf{f}(x) + h\mathbf{f}'(x) + \frac{h^2}{2}\mathbf{f}''(x) + \frac{h^3}{6}\mathbf{f}'''(x) + \cdots$$

Thus, the forward, backward, and central difference approximations (Sections 2.1–2.3), as well as the complex-step approximation (Section 3.1) retain the exact same form, except we replace the scalar-valued f with the vector-valued \mathbf{f} .

$$\left. \frac{d\mathbf{f}}{dx} \right|_{x=x_0} \approx \underbrace{\frac{\mathbf{f}(x_0+h) - \mathbf{f}(x_0)}{h}}_{\text{forward difference}} \quad (4.1)$$

$$\left. \frac{d\mathbf{f}}{dx} \right|_{x=x_0} \approx \underbrace{\frac{\mathbf{f}(x_0) - \mathbf{f}(x_0-h)}{h}}_{\text{backward difference}} \quad (4.2)$$

$$\left. \frac{d\mathbf{f}}{dx} \right|_{x=x_0} \approx \underbrace{\frac{\mathbf{f}(x_0+h) - \mathbf{f}(x_0-h)}{h}}_{\text{central difference}} \quad (4.3)$$

$$\left. \frac{d\mathbf{f}}{dx} \right|_{x=x_0} \approx \underbrace{\frac{\text{Im}[\mathbf{f}(x_0+ih)]}{h}}_{\text{complex-step}} \quad (4.4)$$

4.2 Partial Derivatives of Scalar-Valued Functions

Recall from Section 2.1 that the Taylor series expansion for a univariate, scalar-valued function $f : \mathbb{R} \rightarrow \mathbb{R}$ is given by

$$f(x+h) = f(x) + hf'(x) + \frac{h^2}{2}f''(x) + \frac{h^3}{6}f'''(x) + \dots$$

Now, consider the case where we have a multivariate, scalar-valued function $f : \mathbb{R}^n \rightarrow \mathbb{R}$. If we only allow the k th component of its independent variable to vary (i.e. x_k), then its Taylor series expansion in the k th direction is

$$f(\mathbf{x} + h\mathbf{e}_k) = f(\mathbf{x}) + h \frac{\partial f}{\partial x_k} + \frac{h^2}{2} \frac{\partial^2 f}{\partial x_k^2} + \frac{h^3}{6} \frac{\partial^3 f}{\partial x_k^3} + \dots$$

Using an identical procedure to the one in Section 2.1, we can find the forward difference approximation of the *partial* derivative to be

$$\left. \frac{\partial f}{\partial x_k} \right|_{\mathbf{x}=\mathbf{x}_0} \approx \underbrace{\frac{f(\mathbf{x}_0 + h\mathbf{e}_k) - f(\mathbf{x}_0)}{h}}_{\text{forward difference}} \quad (4.5)$$

Similarly, the backward difference, central difference, and complex-step approximations are [10, pp. 227–228, 233]

$$\left. \frac{\partial f}{\partial x_k} \right|_{\mathbf{x}=\mathbf{x}_0} \approx \underbrace{\frac{f(\mathbf{x}_0) - f(\mathbf{x}_0 - h\mathbf{e}_k)}{h}}_{\text{backward difference}} \quad (4.6)$$

$$\left. \frac{\partial f}{\partial x_k} \right|_{\mathbf{x}=\mathbf{x}_0} \approx \underbrace{\frac{f(\mathbf{x}_0 + h\mathbf{e}_k) - f(\mathbf{x}_0 - h\mathbf{e}_k)}{h}}_{\text{central difference}} \quad (4.7)$$

$$\left. \frac{\partial f}{\partial x_k} \right|_{\mathbf{x}=\mathbf{x}_0} \approx \underbrace{\frac{\text{Im}[f(\mathbf{x}_0 + ih\mathbf{e}_k)]}{h}}_{\text{complex-step}} \quad (4.8)$$

4.3 Partial Derivatives of Vector-Valued Functions

In the most general case, we can consider a multivariate, vector-valued function $\mathbf{f} : \mathbb{R}^n \rightarrow \mathbb{R}^m$. By combining the analyses from Sections 4.1 and 4.2, we can generalize the approximations to find the partial derivatives of vector-valued functions.

$$\left. \frac{\partial \mathbf{f}}{\partial x_k} \right|_{\mathbf{x}=\mathbf{x}_0} \approx \underbrace{\frac{\mathbf{f}(\mathbf{x}_0 + h\mathbf{e}_k) - \mathbf{f}(\mathbf{x}_0)}{h}}_{\text{forward difference}} \quad (4.9)$$

$$\left. \frac{\partial \mathbf{f}}{\partial x_k} \right|_{\mathbf{x}=\mathbf{x}_0} \approx \underbrace{\frac{\mathbf{f}(\mathbf{x}_0) - \mathbf{f}(\mathbf{x}_0 - h\mathbf{e}_k)}{h}}_{\text{backward difference}} \quad (4.10)$$

$$\left. \frac{\partial \mathbf{f}}{\partial x_k} \right|_{\mathbf{x}=\mathbf{x}_0} \approx \underbrace{\frac{\mathbf{f}(\mathbf{x}_0 + h\mathbf{e}_k) - \mathbf{f}(\mathbf{x}_0 - h\mathbf{e}_k)}{h}}_{\text{central difference}} \quad (4.11)$$

$$\left. \frac{\partial \mathbf{f}}{\partial x_k} \right|_{\mathbf{x}=\mathbf{x}_0} \approx \underbrace{\frac{\text{Im}[\mathbf{f}(\mathbf{x}_0 + ih\mathbf{e}_k)]}{h}}_{\text{complex-step}} \quad (4.12)$$

4.4 Summary

In this section, we summarize all the various approximations. Note that for the finite difference approximations, we replace the relative step size, h , with the absolute step size, Δx , which we can recall¹ is defined as

$$\Delta x = h(1 + |x_0|) \quad (4.13)$$

when evaluating a derivative about the point $x_0 \in \mathbb{R}$. In the multivariate case, when evaluating a derivative about the point $\mathbf{x}_0 = (x_{0,1}, \dots, x_{0,n}) \in \mathbb{R}^n$, we use

$$\Delta x_k = h(1 + |x_{0,k}|) \quad (4.14)$$

Derivatives of univariate, scalar-valued functions.

Consider a univariate, scalar-valued function $f : \mathbb{R} \rightarrow \mathbb{R}$. Approximations for the derivative of f with respect to $x \in \mathbb{R}$, evaluated at the point $x = x_0$, are given by Eq. (4.15) below.

$$\left. \frac{df}{dx} \right|_{x=x_0} \approx \begin{cases} \frac{f(x_0 + \Delta x) - f(x_0)}{\Delta x}, & \text{forward difference} \\ \frac{f(x_0) - f(x_0 - \Delta x)}{\Delta x}, & \text{backward difference} \\ \frac{f(x_0 + \Delta x) - f(x_0 - \Delta x)}{2\Delta x}, & \text{central difference} \\ \frac{\text{Im}[f(x_0 + ih)]}{h}, & \text{complex-step} \end{cases} \quad (4.15)$$

Derivatives of univariate, vector-valued functions.

Consider a univariate, vector-valued function $\mathbf{f} : \mathbb{R} \rightarrow \mathbb{R}^m$. Approximations for the derivative of \mathbf{f} with respect to $x \in \mathbb{R}$, evaluated at the point $x = x_0$, are given by Eq. (4.16) below.

$$\left. \frac{d\mathbf{f}}{dx} \right|_{x=x_0} \approx \begin{cases} \frac{\mathbf{f}(x_0 + \Delta x) - \mathbf{f}(x_0)}{\Delta x}, & \text{forward difference} \\ \frac{\mathbf{f}(x_0) - \mathbf{f}(x_0 - \Delta x)}{\Delta x}, & \text{backward difference} \\ \frac{\mathbf{f}(x_0 + \Delta x) - \mathbf{f}(x_0 - \Delta x)}{2\Delta x}, & \text{central difference} \\ \frac{\text{Im}[\mathbf{f}(x_0 + ih)]}{h}, & \text{complex-step} \end{cases} \quad (4.16)$$

Partial derivatives of multivariate, scalar-valued functions.

¹ See Section 2.4.

Consider a multivariate, scalar-valued function $f : \mathbb{R}^n \rightarrow \mathbb{R}$. Approximations for the partial derivative of f with respect to $x_k \in \mathbb{R}$, evaluated at the point $\mathbf{x} = \mathbf{x}_0$, are given by Eq. (4.17) below.

$$\left. \frac{\partial f}{\partial x_k} \right|_{\mathbf{x}=\mathbf{x}_0} \approx \begin{cases} \frac{f(\mathbf{x}_0 + \mathbf{e}_k \Delta x_k) - f(\mathbf{x}_0)}{\Delta x_k}, & \text{forward difference} \\ \frac{f(\mathbf{x}_0) - f(\mathbf{x}_0 - \mathbf{e}_k \Delta x_k)}{\Delta x_k}, & \text{backward difference} \\ \frac{f(\mathbf{x}_0 + \mathbf{e}_k \Delta x_k) - f(\mathbf{x}_0 - \mathbf{e}_k \Delta x_k)}{2\Delta x_k}, & \text{central difference} \\ \frac{\text{Im}[f(\mathbf{x}_0 + ih\mathbf{e}_k)]}{h}, & \text{complex-step} \end{cases} \quad (4.17)$$

Partial derivatives of multivariate, vector-valued functions.

Consider a multivariate, vector-valued function $\mathbf{f} : \mathbb{R}^n \rightarrow \mathbb{R}^m$. Approximations for the partial derivative of \mathbf{f} with respect to $x_k \in \mathbb{R}$, evaluated at the point $\mathbf{x} = \mathbf{x}_0$, are given by Eq. (4.18) below.

$$\left. \frac{\partial \mathbf{f}}{\partial x_k} \right|_{\mathbf{x}=\mathbf{x}_0} \approx \begin{cases} \frac{\mathbf{f}(\mathbf{x}_0 + \mathbf{e}_k \Delta x_k) - \mathbf{f}(\mathbf{x}_0)}{\Delta x_k}, & \text{forward difference} \\ \frac{\mathbf{f}(\mathbf{x}_0) - \mathbf{f}(\mathbf{x}_0 - \mathbf{e}_k \Delta x_k)}{\Delta x_k}, & \text{backward difference} \\ \frac{\mathbf{f}(\mathbf{x}_0 + \mathbf{e}_k \Delta x_k) - \mathbf{f}(\mathbf{x}_0 - \mathbf{e}_k \Delta x_k)}{2\Delta x_k}, & \text{central difference} \\ \frac{\text{Im}[\mathbf{f}(\mathbf{x}_0 + ih\mathbf{e}_k)]}{h}, & \text{complex-step} \end{cases} \quad (4.18)$$

PART II

Implementation

5

Numerical Differentiation Using the Forward Difference Approximation

5.1 Derivatives

Consider a univariate, vector-valued function $\mathbf{f} : \mathbb{R} \rightarrow \mathbb{R}^m$. Recall¹ that the forward difference approximation of the derivative of \mathbf{f} with respect to $x \in \mathbb{R}$, evaluated at the point $x = x_0$, is defined as

$$\left. \frac{d\mathbf{f}}{dx} \right|_{x=x_0} \approx \frac{\mathbf{f}(x_0 + \Delta x) - \mathbf{f}(x_0)}{\Delta x}$$

where the absolute step size is defined as

$$\Delta x = h(1 + |x_0|)$$

Algorithm 1: fderivative

Derivative of a univariate, vector-valued function using the forward difference approximation.

Given:

- $\mathbf{f}(x)$ - univariate, vector-valued function ($\mathbf{f} : \mathbb{R} \rightarrow \mathbb{R}^m$)
- $x_0 \in \mathbb{R}$ - evaluation point
- $h \in \mathbb{R}$ - (*OPTIONAL*) relative step size (defaults to $\sqrt{\varepsilon}$)

Procedure:

1. Default the relative step size to $h = \sqrt{\varepsilon}$ if not input.
2. Absolute step size.

$$\Delta x = h(1 + |x_0|)$$

¹ See Sections 2.1 and 4.4.

3. Evaluate the derivative.

$$\left. \frac{df}{dx} \right|_{x=x_0} = \frac{f(x_0 + \Delta x) - f(x_0)}{\Delta x}$$

Return:

- $\left. \frac{df}{dx} \right|_{x=x_0} \in \mathbb{R}^m$ - derivative of \mathbf{f} with respect to x , evaluated at $x = x_0$

Note:

- This algorithm requires 2 evaluations of $\mathbf{f}(x)$.
- If the function is scalar-valued, then $m = 1$.

5.2 Partial Derivatives

Consider a multivariate, vector-valued function $\mathbf{f} : \mathbb{R}^n \rightarrow \mathbb{R}^m$. Recall² that the forward difference approximation of the partial derivative of \mathbf{f} with respect to $x_k \in \mathbb{R}$, evaluated at the point $\mathbf{x} = \mathbf{x}_0$, is defined as

$$\left. \frac{\partial \mathbf{f}}{\partial x_k} \right|_{\mathbf{x}=\mathbf{x}_0} \approx \frac{\mathbf{f}(\mathbf{x}_0 + \mathbf{e}_k \Delta x_k) - \mathbf{f}(\mathbf{x}_0)}{\Delta x_k}$$

where the absolute step size is defined as

$$\Delta x_k = h(1 + |x_{0,k}|)$$

Algorithm 2: fpartial

Partial derivative of a multivariate, vector-valued function using the forward difference approximation.

Given:

- $\mathbf{f}(\mathbf{x})$ - multivariate, vector-valued function ($\mathbf{f} : \mathbb{R}^n \rightarrow \mathbb{R}^m$)
- $\mathbf{x}_0 \in \mathbb{R}^n$ - evaluation point
- $k \in \mathbb{Z}$ - element of \mathbf{x} to differentiate with respect to
- $h \in \mathbb{R}$ - (*OPTIONAL*) relative step size (defaults to $\sqrt{\varepsilon}$)

Procedure:

1. Default the relative step size to $h = \sqrt{\varepsilon}$ if not input.
2. Evaluate and store the value of $\mathbf{f}(\mathbf{x}_0)$.

$$\mathbf{f}_0 = \mathbf{f}(\mathbf{x}_0)$$

3. Absolute step size.

$$\Delta x_k = h(1 + |x_{0,k}|)$$

4. Step in the k th direction.

$$x_{0,k} = x_{0,k} + \Delta x_k$$

² See Sections 2.1 and 4.4.

5. Evaluate the partial derivative of $\mathbf{f}(\mathbf{x})$ with respect to x_k .

$$\left. \frac{\partial \mathbf{f}}{\partial x_k} \right|_{\mathbf{x}=\mathbf{x}_0} = \frac{\mathbf{f}(\mathbf{x}_0) - \mathbf{f}_0}{\Delta x_k}$$

Return:

- $\left. \frac{\partial \mathbf{f}}{\partial x_k} \right|_{\mathbf{x}=\mathbf{x}_0} \in \mathbb{R}^m$ - partial derivative of \mathbf{f} with respect to x_k , evaluated at $\mathbf{x} = \mathbf{x}_0$

Note:

- This algorithm requires 2 evaluations of $\mathbf{f}(\mathbf{x})$.
- If the function is scalar-valued, then $m = 1$.

5.3 Gradients

Consider a multivariate, scalar-valued function $f : \mathbb{R}^n \rightarrow \mathbb{R}$. Recall³ that the gradient of f with respect to $\mathbf{x} \in \mathbb{R}^n$, evaluated at the point $\mathbf{x} = \mathbf{x}_0$, is defined as

$$\mathbf{g}(\mathbf{x}_0) = \nabla f(\mathbf{x}_0) = \begin{bmatrix} \left. \frac{\partial f}{\partial x_1} \right|_{\mathbf{x}=\mathbf{x}_0} \\ \vdots \\ \left. \frac{\partial f}{\partial x_n} \right|_{\mathbf{x}=\mathbf{x}_0} \end{bmatrix}$$

The procedure for evaluating the individual partial derivatives in the equation above is detailed in Section 5.2. Algorithm 3 below is adapted from [10, p. 232].

Algorithm 3: fgradient

Gradient of a multivariate, scalar-valued function using the forward difference approximation.

Given:

- $f(\mathbf{x})$ - multivariate, scalar-valued function ($f : \mathbb{R}^n \rightarrow \mathbb{R}$)
- $\mathbf{x}_0 \in \mathbb{R}^n$ - evaluation point
- $h \in \mathbb{R}$ - (OPTIONAL) relative step size (defaults to $\sqrt{\varepsilon}$)

Procedure:

1. Default the relative step size to $h = \sqrt{\varepsilon}$ if not input.
2. Determine n given that $\mathbf{x}_0 \in \mathbb{R}^n$.
3. Preallocate the vector $\mathbf{g} \in \mathbb{R}^n$ to store the gradient.
4. Evaluate and store the value of $f(\mathbf{x}_0)$.

$$f_0 = f(\mathbf{x}_0)$$

5. Evaluate the gradient.

³ See Section 1.5.

```

for  $k = 1$  to  $n$ 
    (a) Absolute step size.
        
$$\Delta x_k = h(1 + |x_{0,k}|)$$

    (b) Step in the  $k$ th direction.
        
$$x_{0,k} = x_{0,k} + \Delta x_k$$

    (c) Partial derivative of  $f$  with respect to  $x_k$ .
        
$$g_k = \frac{f(\mathbf{x}_0) - f_0}{\Delta x_k}$$

    (d) Reset  $\mathbf{x}_0$ .
        
$$x_{0,k} = x_{0,k} - \Delta x_k$$

end

```

Return:

- $\mathbf{g} = \mathbf{g}(\mathbf{x}_0) \in \mathbb{R}^n$ - gradient of f with respect to \mathbf{x} , evaluated at $\mathbf{x} = \mathbf{x}_0$

Note:

- This algorithm requires $n + 1$ evaluations of $f(\mathbf{x})$.

5.4 Directional Derivatives

Consider a multivariate, scalar-valued function $f : \mathbb{R}^n \rightarrow \mathbb{R}$. Recall⁴ that the directional derivative of f with respect to $\mathbf{x} \in \mathbb{R}^n$ in the direction of $\mathbf{v} \in \mathbb{R}^n$, evaluated at the point $\mathbf{x} = \mathbf{x}_0$, is defined as

$$\nabla_{\mathbf{v}} f(\mathbf{x}_0) = \nabla f(\mathbf{x}_0)^T \mathbf{v}$$

Therefore, we can simply add a single additional line to Algorithm 3 to develop Algorithm 4 for calculating directional derivatives.

Algorithm 4: fdirectional

Directional derivative of a multivariate, scalar-valued function using the forward difference approximation.

Given:

- $f(\mathbf{x})$ - multivariate, scalar-valued function ($f : \mathbb{R}^n \rightarrow \mathbb{R}$)
- $\mathbf{x}_0 \in \mathbb{R}^n$ - evaluation point
- $\mathbf{v} \in \mathbb{R}^n$ - vector defining direction of differentiation
- $h \in \mathbb{R}$ - (*OPTIONAL*) relative step size (defaults to $\sqrt{\varepsilon}$)

Procedure:

⁴ See Eq. (1.11) in Section 1.6.

1. Default the relative step size to $h = \sqrt{\varepsilon}$ if not input.
2. Determine n given that $\mathbf{x}_0 \in \mathbb{R}^n$.
3. Preallocate the vector $\mathbf{g} \in \mathbb{R}^n$ to store the gradient.
4. Evaluate and store the value of $f(\mathbf{x}_0)$.

$$f_0 = f(\mathbf{x}_0)$$

5. Evaluate the gradient.

for $k = 1$ **to** n

- (a) Absolute step size.

$$\Delta x_k = h(1 + |x_{0,k}|)$$

- (b) Step in the k th direction.

$$x_{0,k} = x_{0,k} + \Delta x_k$$

- (c) Partial derivative of f with respect to x_k .

$$g_k = \frac{f(\mathbf{x}_0) - f_0}{\Delta x_k}$$

- (d) Reset \mathbf{x}_0 .

$$x_{0,k} = x_{0,k} - \Delta x_k$$

end

6. Evaluate the directional derivative.

$$\nabla_{\mathbf{v}} f(\mathbf{x}_0) = \mathbf{g}^T \mathbf{v}$$

Return:

- $\nabla_{\mathbf{v}} f(\mathbf{x}_0) \in \mathbb{R}$ - directional derivative of f with respect to \mathbf{x} in the direction of \mathbf{v} , evaluated at $\mathbf{x} = \mathbf{x}_0$

Note:

- This algorithm requires $n + 1$ evaluations of $f(\mathbf{x})$.
- This implementation does *not* assume that \mathbf{v} is a unit vector.

5.5 Jacobians

Consider a multivariate, vector-valued function $\mathbf{f} : \mathbb{R}^n \rightarrow \mathbb{R}^m$. Recall⁵ that the Jacobian of \mathbf{f} with respect to $\mathbf{x} \in \mathbb{R}^n$, evaluated at the point $\mathbf{x} = \mathbf{x}_0$, is defined as

$$\mathbf{J}(\mathbf{x}_0) = \frac{\partial \mathbf{f}}{\partial \mathbf{x}} \bigg|_{\mathbf{x}=\mathbf{x}_0} = \begin{bmatrix} \frac{\partial \mathbf{f}}{\partial x_1} \bigg|_{\mathbf{x}=\mathbf{x}_0} & \cdots & \frac{\partial \mathbf{f}}{\partial x_n} \bigg|_{\mathbf{x}=\mathbf{x}_0} \end{bmatrix}$$

The procedure for evaluating the individual partial derivatives in the equation above is detailed in Section 5.2.

⁵ See Section 1.7.

Algorithm 5: fjacobian

Jacobian of a multivariate, vector-valued function using the forward difference approximation.

Given:

- $\mathbf{f}(\mathbf{x})$ - multivariate, vector-valued function ($\mathbf{f} : \mathbb{R}^n \rightarrow \mathbb{R}^m$)
- $\mathbf{x}_0 \in \mathbb{R}^n$ - evaluation point
- $h \in \mathbb{R}$ - (*OPTIONAL*) relative step size (defaults to $\sqrt{\varepsilon}$)

Procedure:

1. Default the relative step size to $h = \sqrt{\varepsilon}$ if not input.
2. Evaluate and store the value of $\mathbf{f}(\mathbf{x}_0)$.

$$\mathbf{f}_0 = \mathbf{f}(\mathbf{x}_0)$$

3. Determine m given that $\mathbf{f}_0 \in \mathbb{R}^m$.
4. Determine n given that $\mathbf{x}_0 \in \mathbb{R}^n$.
5. Preallocate the matrix $\mathbf{J} \in \mathbb{R}^{m \times n}$ to store the Jacobian.
6. Evaluate the Jacobian.

for $k = 1$ **to** n

(a) Absolute step size.

$$\Delta x_k = h(1 + |x_{0,k}|)$$

(b) Step in the k th direction.

$$x_{0,k} = x_{0,k} + \Delta x_k$$

(c) Partial derivative of \mathbf{f} with respect to x_k .

$$\mathbf{J}_{:,k} = \frac{\mathbf{f}(\mathbf{x}_0) - \mathbf{f}_0}{\Delta x_k}$$

(d) Reset \mathbf{x}_0 .

$$x_{0,k} = x_{0,k} - \Delta x_k$$

end

Return:

- $\mathbf{J} = \mathbf{J}(\mathbf{x}_0) \in \mathbb{R}^{m \times n}$ - Jacobian of \mathbf{f} with respect to \mathbf{x} , evaluated at $\mathbf{x} = \mathbf{x}_0$

Note:

- This algorithm requires $n + 1$ evaluations of $\mathbf{f}(\mathbf{x})$.

5.6 Hessians

Consider a multivariate, scalar-valued function $\mathbf{f} : \mathbb{R}^n \rightarrow \mathbb{R}$. Recall⁶ that the (j, k) th element of the Hessian of f with respect to $\mathbf{x} \in \mathbb{R}^n$, evaluated at the point $\mathbf{x} = \mathbf{x}_0$, is defined as

$$[\mathbf{H}(\mathbf{x}_0)]_{j,k} = \left. \frac{\partial^2 f}{\partial x_j \partial x_k} \right|_{\mathbf{x}=\mathbf{x}_0}$$

Let's rewrite this equation in a slightly different form.

$$[\mathbf{H}(\mathbf{x}_0)]_{j,k} = \left. \frac{\partial}{\partial x_j} \right|_{\mathbf{x}=\mathbf{x}_0} \left(\left. \frac{\partial f}{\partial x_k} \right|_{\mathbf{x}=\mathbf{x}_0} \right)$$

Recall⁷ that the forward difference approximation for the partial derivative of f is

$$\left. \frac{\partial f}{\partial x_k} \right|_{\mathbf{x}=\mathbf{x}_0} \approx \frac{f(\mathbf{x}_0 + \mathbf{e}_k \Delta x_k) - f(\mathbf{x}_0)}{\Delta x_k}$$

Replacing the derivative in the parentheses with its forward difference approximation, we can write

$$[\mathbf{H}(\mathbf{x}_0)]_{j,k} \approx \left. \frac{\partial}{\partial x_j} \right|_{\mathbf{x}=\mathbf{x}_0} \left[\frac{f(\mathbf{x}_0 + \mathbf{e}_k \Delta x_k) - f(\mathbf{x}_0)}{\Delta x_k} \right]$$

Applying the forward difference approximation once more,

$$\begin{aligned} [\mathbf{H}(\mathbf{x}_0)]_{j,k} &\approx \frac{1}{\Delta x_j} \left[\frac{f(\mathbf{x}_0 + \mathbf{e}_k \Delta x_k + \mathbf{e}_j \Delta x_j) - f(\mathbf{x}_0 + \mathbf{e}_j \Delta x_j)}{\Delta x_k} \right] - \frac{1}{\Delta x_j} \left[\frac{f(\mathbf{x}_0 + \mathbf{e}_k \Delta x_k) - f(\mathbf{x}_0)}{\Delta x_k} \right] \\ &\approx \frac{f(\mathbf{x}_0 + \mathbf{e}_k \Delta x_k + \mathbf{e}_j \Delta x_j) - f(\mathbf{x}_0 + \mathbf{e}_j \Delta x_j) - f(\mathbf{x}_0 + \mathbf{e}_k \Delta x_k) + f(\mathbf{x}_0)}{\Delta x_j \Delta x_k} \end{aligned}$$

With these expression for the (j, k) th element of the Hessian, we can develop Algorithm 6 below. Recall that since the Hessian matrix is symmetric, we only have to evaluate the derivatives in the upper triangle of the matrix (see Section 1.8). Through experimentation, we find that a relative step size of $h = \varepsilon^{1/3}$ is suitable.

Algorithm 6: fhessian

Hessian of a multivariate, scalar-valued function using the forward difference approximation.

Given:

- $f(\mathbf{x})$ - multivariate, scalar-valued function ($f : \mathbb{R}^n \rightarrow \mathbb{R}$)
- $\mathbf{x}_0 \in \mathbb{R}^n$ - evaluation point
- $h \in \mathbb{R}$ - (OPTIONAL) relative step size (defaults to $\varepsilon^{1/3}$)

Procedure:

1. Default the relative step size to $h = \varepsilon^{1/3}$ if not input.
2. Determine n given that $\mathbf{x}_0 \in \mathbb{R}^n$.
3. Preallocate the matrix $\mathbf{H} \in \mathbb{R}^{n \times n}$ to store the Hessian.
4. Evaluate and store the value of $f(\mathbf{x}_0)$.

$$f_0 = f(\mathbf{x}_0)$$

⁶ See Section 1.8

⁷ See Sections 2.1 and 4.4.

5. Preallocate the vector $\mathbf{a} \in \mathbb{R}^n$ to store the absolute step size for each direction k .
6. Preallocate the vector $\mathbf{b} \in \mathbb{R}^n$ to store the evaluations of f with steps in each direction k .
7. Populate \mathbf{a} and \mathbf{b} .

```

for  $k = 1$  to  $n$ 
    (a) Absolute step size.

        
$$\Delta x_k = h(1 + |x_{0,k}|)$$


    (b) Step in the  $k$ th direction.

        
$$x_{0,k} = x_{0,k} + \Delta x_k$$


    (c) Function evaluation.

        
$$b_k = f(\mathbf{x}_0)$$


    (d) Reset  $\mathbf{x}_0$ .

        
$$x_{0,k} = x_{0,k} - \Delta x_k$$


    (e) Store  $\Delta x_k$  in  $\mathbf{a}$ .

        
$$a_k = \Delta x_k$$

end

```

8. Evaluate the Hessian, looping through the upper triangular elements.

```

for  $k = 1$  to  $n$ 
    for  $j = k$  to  $n$ 
        (a) Step in the  $j$ th and  $k$ th directions.

            
$$x_{0,j} = x_{0,j} + a_j$$

            
$$x_{0,k} = x_{0,k} + a_k$$


        (b) Evaluate the  $(j, k)$ th element of the Hessian.

            
$$H_{j,k} = \frac{f(\mathbf{x}_0) - b_j - b_k + f_0}{a_j a_k}$$


        (c) Evaluate the  $(k, j)$ th element of the Hessian using symmetry.

            
$$H_{k,j} = H_{j,k}$$


        (d) Reset  $\mathbf{x}_0$ .

            
$$x_{0,j} = x_{0,j} - a_j$$

            
$$x_{0,k} = x_{0,k} - a_k$$

    end
end

```

Return:

- $\mathbf{H} = \mathbf{H}(\mathbf{x}_0) \in \mathbb{R}^{n \times n}$ - Hessian of f with respect to \mathbf{x} , evaluated at $\mathbf{x} = \mathbf{x}_0$

Note:

- This algorithm requires $n + 1$ evaluations of $f(\mathbf{x})$.

6

Numerical Differentiation Using the Central Difference Approximation

6.1 Derivatives

Consider a univariate, vector-valued function $\mathbf{f} : \mathbb{R} \rightarrow \mathbb{R}^m$. Recall¹ that the central difference approximation of the derivative of \mathbf{f} with respect to $x \in \mathbb{R}$, evaluated at the point $x = x_0$, is defined as

$$\left. \frac{d\mathbf{f}}{dx} \right|_{x=x_0} \approx \frac{\mathbf{f}(x_0 + \Delta x) - \mathbf{f}(x_0 - \Delta x)}{2\Delta x}$$

where the absolute step size is defined as

$$\Delta x = h(1 + |x_0|)$$

Algorithm 7: cderivative

Derivative of a univariate, vector-valued function using the central difference approximation.

Given:

- $\mathbf{f}(x)$ - univariate, vector-valued function ($\mathbf{f} : \mathbb{R} \rightarrow \mathbb{R}^m$)
- $x_0 \in \mathbb{R}$ - evaluation point
- $h \in \mathbb{R}$ - (*OPTIONAL*) relative step size (defaults to $\varepsilon^{1/3}$)

Procedure:

1. Default the relative step size to $h = \varepsilon^{1/3}$ if not input.
2. Absolute step size.

$$\Delta x = h(1 + |x_0|)$$

¹ See Sections 2.3 and 4.4.

3. Evaluate the derivative.

$$\left. \frac{df}{dx} \right|_{x=x_0} = \frac{f(x_0 + \Delta x) - f(x_0 - \Delta x)}{2\Delta x}$$

Return:

- $\left. \frac{df}{dx} \right|_{x=x_0} \in \mathbb{R}^m$ - derivative of \mathbf{f} with respect to x , evaluated at $x = x_0$

Note:

- This algorithm requires 2 evaluations of $\mathbf{f}(x)$.
- If the function is scalar-valued, then $m = 1$.

6.2 Partial Derivatives

Consider a multivariate, vector-valued function $\mathbf{f} : \mathbb{R}^n \rightarrow \mathbb{R}^m$. Recall² that the central difference approximation of the partial derivative of \mathbf{f} with respect to $x_k \in \mathbb{R}$, evaluated at the point $\mathbf{x} = \mathbf{x}_0$, is defined as

$$\left. \frac{\partial \mathbf{f}}{\partial x_k} \right|_{\mathbf{x}=\mathbf{x}_0} \approx \frac{\mathbf{f}(\mathbf{x}_0 + \mathbf{e}_k \Delta x_k) - \mathbf{f}(\mathbf{x}_0 - \mathbf{e}_k \Delta x_k)}{2\Delta x_k}$$

where the absolute step size is defined as

$$\Delta x_k = h(1 + |x_{0,k}|)$$

Algorithm 8: cpartial

Partial derivative of a multivariate, vector-valued function using the central difference approximation.

Given:

- $\mathbf{f}(\mathbf{x})$ - multivariate, vector-valued function ($\mathbf{f} : \mathbb{R}^n \rightarrow \mathbb{R}^m$)
- $\mathbf{x}_0 \in \mathbb{R}^n$ - evaluation point
- $k \in \mathbb{Z}$ - element of \mathbf{x} to differentiate with respect to
- $h \in \mathbb{R}$ - (OPTIONAL) relative step size (defaults to $\varepsilon^{1/3}$)

Procedure:

1. Default the relative step size to $h = \varepsilon^{1/3}$ if not input.
2. Absolute step size.

$$\Delta x_k = h(1 + |x_{0,k}|)$$

3. Step forward in the k th direction.

$$x_{0,k} = x_{0,k} + \Delta x_k$$

$$\mathbf{f}_1 = \mathbf{f}(\mathbf{x}_0)$$

4. Step backward in the k th direction.

$$x_{0,k} = x_{0,k} - 2\Delta x_k$$

$$\mathbf{f}_2 = \mathbf{f}(\mathbf{x}_0)$$

² See Sections 2.3 and 4.4.

5. Evaluate the partial derivative of $\mathbf{f}(\mathbf{x})$ with respect to x_k .

$$\left. \frac{\partial \mathbf{f}}{\partial x_k} \right|_{\mathbf{x}=\mathbf{x}_0} = \frac{\mathbf{f}_1 - \mathbf{f}_2}{2\Delta x_k}$$

Return:

- $\left. \frac{\partial \mathbf{f}}{\partial x_k} \right|_{\mathbf{x}=\mathbf{x}_0} \in \mathbb{R}^m$ - partial derivative of \mathbf{f} with respect to x_k , evaluated at $\mathbf{x} = \mathbf{x}_0$

Note:

- This algorithm requires 2 evaluations of $\mathbf{f}(\mathbf{x})$.
- If the function is scalar-valued, then $m = 1$.

6.3 Gradients

Consider a multivariate, scalar-valued function $f : \mathbb{R}^n \rightarrow \mathbb{R}$. Recall³ that the gradient of f with respect to $\mathbf{x} \in \mathbb{R}^n$, evaluated at the point $\mathbf{x} = \mathbf{x}_0$, is defined as

$$\mathbf{g}(\mathbf{x}_0) = \nabla f(\mathbf{x}_0) = \begin{bmatrix} \left. \frac{\partial f}{\partial x_1} \right|_{\mathbf{x}=\mathbf{x}_0} \\ \vdots \\ \left. \frac{\partial f}{\partial x_n} \right|_{\mathbf{x}=\mathbf{x}_0} \end{bmatrix}$$

The procedure for evaluating the individual partial derivatives in the equation above is detailed in Section 6.2.

Algorithm 9: cgradient

Gradient of a multivariate, scalar-valued function using the central difference approximation.

Given:

- $f(\mathbf{x})$ - multivariate, scalar-valued function ($f : \mathbb{R}^n \rightarrow \mathbb{R}$)
- $\mathbf{x}_0 \in \mathbb{R}^n$ - evaluation point
- $h \in \mathbb{R}$ - (*OPTIONAL*) relative step size (defaults to $\varepsilon^{1/3}$)

Procedure:

1. Default the relative step size to $h = \varepsilon^{1/3}$ if not input.
2. Determine n given that $\mathbf{x}_0 \in \mathbb{R}^n$.
3. Preallocate the vector $\mathbf{g} \in \mathbb{R}^n$ to store the gradient.
4. Evaluate the gradient.

for $k = 1$ **to** n

³ See Section 1.5.

```

(a) Absolute step size.
    
$$\Delta x_k = h(1 + |x_{0,k}|)$$

(b) Step forward in the  $k$ th direction.
    
$$x_{0,k} = x_{0,k} + \Delta x_k$$

    
$$f_1 = f(\mathbf{x}_0)$$

(c) Step backward in the  $k$ th direction.
    
$$x_{0,k} = x_{0,k} - 2\Delta x_k$$

    
$$f_2 = f(\mathbf{x}_0)$$

(d) Partial derivative of  $f$  with respect to  $x_k$ .
    
$$g_k = \frac{f_1 - f_2}{2\Delta x_k}$$

(e) Reset  $\mathbf{x}_0$ .
    
$$x_{0,k} = x_{0,k} + \Delta x_k$$

end

```

Return:

- $\mathbf{g} = \mathbf{g}(\mathbf{x}_0) \in \mathbb{R}^n$ - gradient of f with respect to \mathbf{x} , evaluated at $\mathbf{x} = \mathbf{x}_0$

Note:

- This algorithm requires $2n$ evaluations of $f(\mathbf{x})$.

6.4 Directional Derivatives

Consider a multivariate, scalar-valued function $f : \mathbb{R}^n \rightarrow \mathbb{R}$. Recall⁴ that the directional derivative of f with respect to $\mathbf{x} \in \mathbb{R}^n$ in the direction of $\mathbf{v} \in \mathbb{R}^n$, evaluated at the point $\mathbf{x} = \mathbf{x}_0$, is defined as

$$\nabla_{\mathbf{v}} f(\mathbf{x}_0) = \nabla f(\mathbf{x}_0)^T \mathbf{v}$$

Therefore, we can simply add a single additional line to Algorithm 3 to develop Algorithm 10 for calculating directional derivatives.

⁴ See Eq. (1.11) in Section 1.6.

Algorithm 10: cdirectional

Directional derivative of a multivariate, scalar-valued function using the central difference approximation.

Given:

- $f(\mathbf{x})$ - multivariate, scalar-valued function ($f : \mathbb{R}^n \rightarrow \mathbb{R}$)
- $\mathbf{x}_0 \in \mathbb{R}^n$ - evaluation point
- $\mathbf{v} \in \mathbb{R}^n$ - vector defining direction of differentiation
- $h \in \mathbb{R}$ - (*OPTIONAL*) relative step size (defaults to $\varepsilon^{1/3}$)

Procedure:

1. Default the relative step size to $h = \varepsilon^{1/3}$ if not input.
2. Determine n given that $\mathbf{x}_0 \in \mathbb{R}^n$.
3. Preallocate the vector $\mathbf{g} \in \mathbb{R}^n$ to store the gradient.
4. Evaluate the gradient.

for $k = 1$ to n

(a) Absolute step size.

$$\Delta x_k = h(1 + |x_{0,k}|)$$

(b) Step forward in the k th direction.

$$x_{0,k} = x_{0,k} + \Delta x_k$$

$$f_1 = f(\mathbf{x}_0)$$

(c) Step backward in the k th direction.

$$x_{0,k} = x_{0,k} - 2\Delta x_k$$

$$f_2 = f(\mathbf{x}_0)$$

(d) Partial derivative of f with respect to x_k .

$$g_k = \frac{f_1 - f_2}{2\Delta x_k}$$

(e) Reset \mathbf{x}_0 .

$$x_{0,k} = x_{0,k} + \Delta x_k$$

end

5. Evaluate the directional derivative.

$$\nabla_{\mathbf{v}} f(\mathbf{x}_0) = \mathbf{g}^T \mathbf{v}$$

Return:

- $\nabla_{\mathbf{v}} f(\mathbf{x}_0) \in \mathbb{R}$ - directional derivative of f with respect to \mathbf{x} in the direction of \mathbf{v} , evaluated at $\mathbf{x} = \mathbf{x}_0$

Note:

- This algorithm requires $2n$ evaluations of $f(\mathbf{x})$.
- This implementation does *not* assume that \mathbf{v} is a unit vector.

6.5 Jacobians

Consider a multivariate, vector-valued function $\mathbf{f} : \mathbb{R}^n \rightarrow \mathbb{R}^m$. Recall⁵ that the Jacobian of \mathbf{f} with respect to $\mathbf{x} \in \mathbb{R}^n$, evaluated at the point $\mathbf{x} = \mathbf{x}_0$, is defined as

$$\mathbf{J}(\mathbf{x}_0) = \frac{\partial \mathbf{f}}{\partial \mathbf{x}} \bigg|_{\mathbf{x}=\mathbf{x}_0} = \begin{bmatrix} \frac{\partial \mathbf{f}}{\partial x_1} \big|_{\mathbf{x}=\mathbf{x}_0} & \cdots & \frac{\partial \mathbf{f}}{\partial x_n} \big|_{\mathbf{x}=\mathbf{x}_0} \end{bmatrix}$$

The procedure for evaluating the individual partial derivatives in the equation above is detailed in Section 6.2.

Algorithm 11: cjacobian

Jacobian of a multivariate, vector-valued function using the central difference approximation.

Given:

- $\mathbf{f}(\mathbf{x})$ - multivariate, vector-valued function ($\mathbf{f} : \mathbb{R}^n \rightarrow \mathbb{R}^m$)
- $\mathbf{x}_0 \in \mathbb{R}^n$ - evaluation point
- $h \in \mathbb{R}$ - (*OPTIONAL*) relative step size (defaults to $\varepsilon^{1/3}$)

Procedure:

1. Default the relative step size to $h = \varepsilon^{1/3}$ if not input.
2. Determine m given that $\mathbf{f}(\mathbf{x}_0) \in \mathbb{R}^m$.
3. Determine n given that $\mathbf{x}_0 \in \mathbb{R}^n$.
4. Preallocate the matrix $\mathbf{J} \in \mathbb{R}^{m \times n}$ to store the Jacobian.
5. Evaluate the Jacobian.

for $k = 1$ **to** n

(a) Absolute step size.

$$\Delta x_k = h(1 + |x_{0,k}|)$$

(b) Step forward in the k th direction.

$$x_{0,k} = x_{0,k} + \Delta x_k$$

$$\mathbf{f}_1 = \mathbf{f}(\mathbf{x}_0)$$

(c) Step backward in the k th direction.

$$x_{0,k} = x_{0,k} - 2\Delta x_k$$

$$\mathbf{f}_2 = \mathbf{f}(\mathbf{x}_0)$$

(d) Partial derivative of \mathbf{f} with respect to x_k .

$$\mathbf{J}_{:,k} = \frac{\mathbf{f}_1 - \mathbf{f}_2}{2\Delta x_k}$$

(e) Reset \mathbf{x}_0 .

$$x_{0,k} = x_{0,k} + \Delta x_k$$

end

⁵ See Section 1.7.

Return:

- $\mathbf{J} = \mathbf{J}(\mathbf{x}_0) \in \mathbb{R}^{m \times n}$ - Jacobian of \mathbf{f} with respect to \mathbf{x} , evaluated at $\mathbf{x} = \mathbf{x}_0$

Note:

- This algorithm requires $2n + 1$ evaluations of $\mathbf{f}(\mathbf{x})$.

6.6 Hessians

Consider a multivariate, scalar-valued function $\mathbf{f} : \mathbb{R}^n \rightarrow \mathbb{R}$. Recall⁶ that the (j, k) th element of the Hessian of f with respect to $\mathbf{x} \in \mathbb{R}^n$, evaluated at the point $\mathbf{x} = \mathbf{x}_0$, is defined as

$$[\mathbf{H}(\mathbf{x}_0)]_{j,k} = \left. \frac{\partial^2 f}{\partial x_j \partial x_k} \right|_{\mathbf{x}=\mathbf{x}_0}$$

Let's rewrite this equation in a slightly different form.

$$[\mathbf{H}(\mathbf{x}_0)]_{j,k} = \left. \frac{\partial}{\partial x_j} \right|_{\mathbf{x}=\mathbf{x}_0} \left(\left. \frac{\partial f}{\partial x_k} \right|_{\mathbf{x}=\mathbf{x}_0} \right)$$

Recall⁷ that the central difference approximation for the partial derivative of f is

$$\left. \frac{\partial f}{\partial x_k} \right|_{\mathbf{x}=\mathbf{x}_0} \approx \frac{f(\mathbf{x}_0 + \mathbf{e}_k \Delta x_k) - f(\mathbf{x}_0 - \mathbf{e}_k \Delta x_k)}{2\Delta x_k}$$

Replacing the derivative in the parentheses with its central difference approximation, we can write

$$[\mathbf{H}(\mathbf{x}_0)]_{j,k} \approx \left. \frac{\partial}{\partial x_j} \right|_{\mathbf{x}=\mathbf{x}_0} \left[\frac{f(\mathbf{x}_0 + \mathbf{e}_k \Delta x_k) - f(\mathbf{x}_0 - \mathbf{e}_k \Delta x_k)}{2\Delta x_k} \right]$$

Applying the central difference approximation once more,

$$\begin{aligned} [\mathbf{H}(\mathbf{x}_0)]_{j,k} \approx & \frac{1}{2\Delta x_j} \left[\frac{f(\mathbf{x}_0 + \mathbf{e}_k \Delta x_k + \mathbf{e}_j \Delta x_j) - f(\mathbf{x}_0 - \mathbf{e}_k \Delta x_k + \mathbf{e}_j \Delta x_j)}{2\Delta x_k} \right] \\ & - \frac{1}{2\Delta x_j} \left[\frac{f(\mathbf{x}_0 + \mathbf{e}_k \Delta x_k - \mathbf{e}_j \Delta x_j) - f(\mathbf{x}_0 - \mathbf{e}_k \Delta x_k - \mathbf{e}_j \Delta x_j)}{2\Delta x_k} \right] \end{aligned}$$

$$[\mathbf{H}(\mathbf{x}_0)]_{j,k} \approx \frac{f(\mathbf{x}_0 + \mathbf{e}_k \Delta x_k + \mathbf{e}_j \Delta x_j) - f(\mathbf{x}_0 - \mathbf{e}_k \Delta x_k + \mathbf{e}_j \Delta x_j) - f(\mathbf{x}_0 + \mathbf{e}_k \Delta x_k - \mathbf{e}_j \Delta x_j) + f(\mathbf{x}_0 - \mathbf{e}_k \Delta x_k - \mathbf{e}_j \Delta x_j)}{4\Delta x_j \Delta x_k}$$

With these expression for the (j, k) th element of the Hessian, we can develop Algorithm 12 below⁸. Recall that since the Hessian matrix is symmetric, we only have to evaluate the derivatives in the upper triangle of the matrix (see Section 1.8). Through experimentation, we find that a relative step size of $h = \varepsilon^{1/3}$ is suitable.

⁶ See Section 1.8.

⁷ See Sections 2.3 and 4.4

⁸ This algorithm was inspired by [1] and [4].

Algorithm 12: chessian

Hessian of a multivariate, scalar-valued function using the central difference approximation.

Given:

- $f(\mathbf{x})$ - multivariate, scalar-valued function ($f : \mathbb{R}^n \rightarrow \mathbb{R}$)
- $\mathbf{x}_0 \in \mathbb{R}^n$ - evaluation point
- $h \in \mathbb{R}$ - (*OPTIONAL*) relative step size (defaults to $\varepsilon^{1/3}$)

Procedure:

1. Default the relative step size to $h = \varepsilon^{1/3}$ if not input.
2. Determine n given that $\mathbf{x}_0 \in \mathbb{R}^n$.
3. Preallocate the matrix $\mathbf{H} \in \mathbb{R}^{n \times n}$ to store the Hessian.
4. Preallocate the vector $\mathbf{a} \in \mathbb{R}^n$ to store the absolute step size for each direction k .
5. Populate \mathbf{a} .

```

for  $k = 1$  to  $n$ 
|    $a_k = h(1 + |x_{0,k}|)$ 
end

```

6. Evaluate the Hessian, looping through the upper triangular elements.

```

for  $k = 1$  to  $n$ 

```

```

    for  $j = k$  to  $n$ 
        (a) Step forward in the  $j$ th and  $k$ th directions.

             $x_{0,j} = x_{0,j} + a_j$ 
             $x_{0,k} = x_{0,k} + a_k$ 
             $b = f(\mathbf{x}_0)$ 
             $x_{0,j} = x_{0,j} - a_j$ 
             $x_{0,k} = x_{0,k} - a_k$ 

        (b) Step forward in the  $j$ th direction and backward in the  $k$ th
            direction.

             $x_{0,j} = x_{0,j} + a_j$ 
             $x_{0,k} = x_{0,k} - a_k$ 
             $c = f(\mathbf{x}_0)$ 
             $x_{0,j} = x_{0,j} - a_j$ 
             $x_{0,k} = x_{0,k} + a_k$ 

        (c) Step backward in the  $j$ th direction and forward in the  $k$ th
            direction.

             $x_{0,j} = x_{0,j} - a_j$ 
             $x_{0,k} = x_{0,k} + a_k$ 
             $d = f(\mathbf{x}_0)$ 
             $x_{0,j} = x_{0,j} + a_j$ 
             $x_{0,k} = x_{0,k} - a_k$ 

        (d) Step backward in the  $j$ th and  $k$ th directions.

             $x_{0,j} = x_{0,j} - a_j$ 
             $x_{0,k} = x_{0,k} - a_k$ 
             $e = f(\mathbf{x}_0)$ 
             $x_{0,j} = x_{0,j} + a_j$ 
             $x_{0,k} = x_{0,k} + a_k$ 

        (e) Evaluate the  $(j, k)$ th element of the Hessian.

            
$$H_{j,k} = \frac{b - c - d + e}{4a_j a_k}$$


        (f) Evaluate the  $(k, j)$ th element of the Hessian using sym-
            metry.

            
$$H_{k,j} = H_{j,k}$$


    end
end

```

Return:

- $\mathbf{H} = \mathbf{H}(\mathbf{x}_0) \in \mathbb{R}^{n \times n}$ - Hessian of f with respect to \mathbf{x} , evaluated at $\mathbf{x} = \mathbf{x}_0$

Note:

- This algorithm requires $2n(n+1)$ evaluations of $f(\mathbf{x})$ (the upper triangular matrix entries (including the diagonal) consist of $n(n+1)/2$ entries [6], and each entry requires 4 evaluations of $f(\mathbf{x})$).

Numerical Differentiation Using the Complex-Step Approximation

7.1 Derivatives

Consider a univariate, vector-valued function $\mathbf{f} : \mathbb{R} \rightarrow \mathbb{R}^m$. Recall¹ that the complex-step approximation of the derivative of \mathbf{f} with respect to $x \in \mathbb{R}$, evaluated at the point $x = x_0$, is defined as

$$\left. \frac{d\mathbf{f}}{dx} \right|_{x=x_0} \approx \frac{\text{Im} [\mathbf{f}(x_0 + ih)]}{h}$$

Algorithm 13: derivative

Derivative of a univariate, vector-valued function using the complex-step approximation.

Given:

- $\mathbf{f}(x)$ - univariate, vector-valued function ($\mathbf{f} : \mathbb{R} \rightarrow \mathbb{R}^m$)
- $x_0 \in \mathbb{R}$ - evaluation point
- $h \in \mathbb{R}$ - (OPTIONAL) step size (defaults to 10^{-200})

Procedure:

1. Default the step size to $h = 10^{-200}$ if not input.
2. Evaluate the derivative.

$$\left. \frac{d\mathbf{f}}{dx} \right|_{x=x_0} = \frac{\text{Im} [\mathbf{f}(x_0 + ih)]}{h}$$

Return:

- $\left. \frac{d\mathbf{f}}{dx} \right|_{x=x_0} \in \mathbb{R}^m$ - derivative of \mathbf{f} with respect to x , evaluated at $x = x_0$

¹ See Sections 3.1 and 4.4.

Note:

- This algorithm requires 1 evaluations of $\mathbf{f}(x)$.
- If the function is scalar-valued, then $m = 1$.

7.2 Partial Derivatives

Consider a multivariate, vector-valued function $\mathbf{f} : \mathbb{R}^n \rightarrow \mathbb{R}^m$. Recall² that the complex-step approximation of the partial derivative of \mathbf{f} with respect to $x_k \in \mathbb{R}$, evaluated at the point $\mathbf{x} = \mathbf{x}_0$, is defined as

$$\left. \frac{\partial \mathbf{f}}{\partial x_k} \right|_{\mathbf{x}=\mathbf{x}_0} \approx \frac{\text{Im} [\mathbf{f}(\mathbf{x}_0 + ih\mathbf{e}_k)]}{h}$$

Algorithm 14: ipartial

Partial derivative of a multivariate, vector-valued function using the complex-step approximation.

Given:

- $\mathbf{f}(\mathbf{x})$ - multivariate, vector-valued function ($\mathbf{f} : \mathbb{R}^n \rightarrow \mathbb{R}^m$)
- $\mathbf{x}_0 \in \mathbb{R}^n$ - evaluation point
- $k \in \mathbb{Z}$ - element of \mathbf{x} to differentiate with respect to
- $h \in \mathbb{R}$ - (*OPTIONAL*) step size (defaults to 10^{-200})

Procedure:

1. Default the step size to $h = 10^{-200}$ if not input.
2. Step in the k th direction.

$$x_{0,k} = x_{0,k} + \Delta x_k$$

3. Evaluate the partial derivative of $\mathbf{f}(\mathbf{x})$ with respect to x_k .

$$\left. \frac{\partial \mathbf{f}}{\partial x_k} \right|_{\mathbf{x}=\mathbf{x}_0} = \frac{\text{Im} [\mathbf{f}(\mathbf{x}_0)]}{h}$$

Return:

- $\left. \frac{\partial \mathbf{f}}{\partial x_k} \right|_{\mathbf{x}=\mathbf{x}_0} \in \mathbb{R}^m$ - partial derivative of \mathbf{f} with respect to x_k , evaluated at $\mathbf{x} = \mathbf{x}_0$

Note:

- This algorithm requires 1 evaluations of $\mathbf{f}(\mathbf{x})$.
- If the function is scalar-valued, then $m = 1$.

² See Sections 3.1 and 4.4.

7.3 Gradients

Consider a multivariate, scalar-valued function $f : \mathbb{R}^n \rightarrow \mathbb{R}$. Recall³ that the gradient of f with respect to $\mathbf{x} \in \mathbb{R}^n$, evaluated at the point $\mathbf{x} = \mathbf{x}_0$, is defined as

$$\mathbf{g}(\mathbf{x}_0) = \nabla f(\mathbf{x}_0) = \begin{bmatrix} \left. \frac{\partial f}{\partial x_1} \right|_{\mathbf{x}=\mathbf{x}_0} \\ \vdots \\ \left. \frac{\partial f}{\partial x_n} \right|_{\mathbf{x}=\mathbf{x}_0} \end{bmatrix}$$

The procedure for evaluating the individual partial derivatives in the equation above is detailed in Section 7.2. Algorithm 15 below is adapted from [10, p. 235].

Algorithm 15: igradient

Gradient of a multivariate, scalar-valued function using the complex-step approximation.

Given:

- $f(\mathbf{x})$ - multivariate, scalar-valued function ($f : \mathbb{R}^n \rightarrow \mathbb{R}$)
- $\mathbf{x}_0 \in \mathbb{R}^n$ - evaluation point
- $h \in \mathbb{R}$ - (*OPTIONAL*) step size (defaults to 10^{-200})

Procedure:

1. Default the step size to $h = 10^{-200}$ if not input.
2. Determine n given that $\mathbf{x}_0 \in \mathbb{R}^n$.
3. Preallocate the vector $\mathbf{g} \in \mathbb{R}^n$ to store the gradient.
4. Evaluate the gradient.

for $k = 1$ **to** n

(a) Step in the k th direction.

$$x_{0,k} = x_{0,k} + \Delta x_k$$

(b) Partial derivative of f with respect to x_k .

$$g_k = \frac{\text{Im}[f(\mathbf{x}_0)]}{h}$$

(c) Reset \mathbf{x}_0 .

$$x_{0,k} = x_{0,k} - \Delta x_k$$

end

Return:

- $\mathbf{g} = \mathbf{g}(\mathbf{x}_0) \in \mathbb{R}^n$ - gradient of f with respect to \mathbf{x} , evaluated at $\mathbf{x} = \mathbf{x}_0$

Note:

³ See Section 1.5.

- This algorithm requires n evaluations of $f(\mathbf{x})$.

7.4 Directional Derivatives

Consider a multivariate, scalar-valued function $f : \mathbb{R}^n \rightarrow \mathbb{R}$. Recall⁴ that the directional derivative of f with respect to $\mathbf{x} \in \mathbb{R}^n$ in the direction of $\mathbf{v} \in \mathbb{R}^n$, evaluated at the point $\mathbf{x} = \mathbf{x}_0$, is defined as

$$\nabla_{\mathbf{v}} f(\mathbf{x}_0) = \nabla f(\mathbf{x}_0)^T \mathbf{v}$$

Therefore, we can simply add a single additional line to Algorithm 3 to develop Algorithm 16 for calculating directional derivatives.

Algorithm 16: idirectional

Directional derivative of a multivariate, scalar-valued function using the complex-step approximation.

Given:

- $f(\mathbf{x})$ - multivariate, scalar-valued function ($f : \mathbb{R}^n \rightarrow \mathbb{R}$)
- $\mathbf{x}_0 \in \mathbb{R}^n$ - evaluation point
- $\mathbf{v} \in \mathbb{R}^n$ - vector defining direction of differentiation
- $h \in \mathbb{R}$ - (*OPTIONAL*) step size (defaults to 10^{-200})

Procedure:

1. Default the step size to $h = 10^{-200}$ if not input.
2. Determine n given that $\mathbf{x}_0 \in \mathbb{R}^n$.
3. Preallocate the vector $\mathbf{g} \in \mathbb{R}^n$ to store the gradient.
4. Evaluate the gradient.

```

for  $k = 1$  to  $n$ 
    (a) Step in the  $k$ th direction.

        
$$x_{0,k} = x_{0,k} + \Delta x_k$$


    (b) Partial derivative of  $f$  with respect to  $x_k$ .

        
$$g_k = \frac{\text{Im}[f(\mathbf{x}_0)]}{h}$$


    (c) Reset  $\mathbf{x}_0$ .

        
$$x_{0,k} = x_{0,k} - \Delta x_k$$

end

```

5. Evaluate the directional derivative.

$$\nabla_{\mathbf{v}} f(\mathbf{x}_0) = \mathbf{g}^T \mathbf{v}$$

Return:

⁴ See Eq. (1.11) in Section 1.6.

- $\nabla_{\mathbf{v}} f(\mathbf{x}_0) \in \mathbb{R}$ - directional derivative of f with respect to \mathbf{x} in the direction of \mathbf{v} , evaluated at $\mathbf{x} = \mathbf{x}_0$

Note:

- This algorithm requires n evaluations of $f(\mathbf{x})$.
- This implementation does *not* assume that \mathbf{v} is a unit vector.

7.5 Jacobians

Consider a multivariate, vector-valued function $\mathbf{f} : \mathbb{R}^n \rightarrow \mathbb{R}^m$. Recall⁵ that the Jacobian of \mathbf{f} with respect to $\mathbf{x} \in \mathbb{R}^n$, evaluated at the point $\mathbf{x} = \mathbf{x}_0$, is defined as

$$\mathbf{J}(\mathbf{x}_0) = \frac{\partial \mathbf{f}}{\partial \mathbf{x}} \bigg|_{\mathbf{x}=\mathbf{x}_0} = \left[\frac{\partial \mathbf{f}}{\partial x_1} \bigg|_{\mathbf{x}=\mathbf{x}_0} \quad \cdots \quad \frac{\partial \mathbf{f}}{\partial x_n} \bigg|_{\mathbf{x}=\mathbf{x}_0} \right]$$

The procedure for evaluating the individual partial derivatives in the equation above is detailed in Section 7.2.

Algorithm 17: ijacobian

Jacobian of a multivariate, vector-valued function using the complex-step approximation.

Given:

- $\mathbf{f}(\mathbf{x})$ - multivariate, vector-valued function ($\mathbf{f} : \mathbb{R}^n \rightarrow \mathbb{R}^m$)
- $\mathbf{x}_0 \in \mathbb{R}^n$ - evaluation point
- $h \in \mathbb{R}$ - (*OPTIONAL*) step size (defaults to 10^{-200})

Procedure:

1. Default the step size to $h = 10^{-200}$ if not input.
2. Determine m given that $\mathbf{f}(\mathbf{x}_0) \in \mathbb{R}^m$.
3. Determine n given that $\mathbf{x}_0 \in \mathbb{R}^n$.
4. Preallocate the matrix $\mathbf{J} \in \mathbb{R}^{m \times n}$ to store the Jacobian.
5. Evaluate the Jacobian.

for $k = 1$ **to** n

(a) Step in the k th direction.

$$x_{0,k} = x_{0,k} + \Delta x_k$$

(b) Partial derivative of \mathbf{f} with respect to x_k .

$$\mathbf{J}_{:,k} = \frac{\text{Im}[\mathbf{f}(\mathbf{x}_0)]}{h}$$

(c) Reset \mathbf{x}_0 .

$$x_{0,k} = x_{0,k} - \Delta x_k$$

end

⁵ See Section 1.7.

Return:

- $\mathbf{J} = \mathbf{J}(\mathbf{x}_0) \in \mathbb{R}^{m \times n}$ - Jacobian of \mathbf{f} with respect to \mathbf{x} , evaluated at $\mathbf{x} = \mathbf{x}_0$

Note:

- This algorithm requires $n + 1$ evaluations of $\mathbf{f}(\mathbf{x})$.

7.6 Hessians

Consider a multivariate, scalar-valued function $\mathbf{f} : \mathbb{R}^n \rightarrow \mathbb{R}$. Recall⁶ that the (j, k) th element of the Hessian of f with respect to $\mathbf{x} \in \mathbb{R}^n$, evaluated at the point $\mathbf{x} = \mathbf{x}_0$, is defined as

$$[\mathbf{H}(\mathbf{x}_0)]_{j,k} = \left. \frac{\partial^2 f}{\partial x_j \partial x_k} \right|_{\mathbf{x}=\mathbf{x}_0}$$

Let's rewrite this equation in a slightly different form.

$$[\mathbf{H}(\mathbf{x}_0)]_{j,k} = \left. \frac{\partial}{\partial x_j} \right|_{\mathbf{x}=\mathbf{x}_0} \left(\left. \frac{\partial f}{\partial x_k} \right|_{\mathbf{x}=\mathbf{x}_0} \right) \quad (7.1)$$

As mentioned in Section 3.4.1, we will not be using a true complex-step approximation for the second derivative in this section. Instead, we will evaluate the first derivative using a complex-step approximation, and the second derivative using a central difference approximation. Recall⁷ that the complex-step approximation for the partial derivative of f is

$$\left. \frac{\partial f}{\partial x_k} \right|_{\mathbf{x}=\mathbf{x}_0} \approx \frac{\text{Im}[f(\mathbf{x}_0 + ih\mathbf{e}_k)]}{h}$$

Replacing the derivative in the parentheses in Eq. (7.1) with its complex-step approximation, we can write

$$[\mathbf{H}(\mathbf{x}_0)]_{j,k} \approx \left. \frac{\partial}{\partial x_j} \right|_{\mathbf{x}=\mathbf{x}_0} \left(\frac{\text{Im}[f(\mathbf{x}_0 + ih\mathbf{e}_k)]}{h} \right) \quad (7.2)$$

Next, recall⁸ that the central difference approximation for the partial derivative of f is

$$\left. \frac{\partial f}{\partial x_k} \right|_{\mathbf{x}=\mathbf{x}_0} \approx \frac{f(\mathbf{x}_0 + \mathbf{e}_k \Delta x_k) - f(\mathbf{x}_0 - \mathbf{e}_k \Delta x_k)}{2\Delta x_k}$$

where

$$\Delta x_k = h_c(1 + |x_{0,k}|)$$

is the absolute step size, and where h_c is the relative step size for the central difference approximation. Applying this approximation to Eq. (7.2),

$$\begin{aligned} [\mathbf{H}(\mathbf{x}_0)]_{j,k} &\approx \frac{1}{2\Delta x_j} \left[\left(\frac{\text{Im}[f(\mathbf{x}_0 + ih_i\mathbf{e}_k + \mathbf{e}_j \Delta x_j)]}{h_i} \right) - \left(\frac{\text{Im}[f(\mathbf{x}_0 + ih_i\mathbf{e}_k - \mathbf{e}_j \Delta x_j)]}{h_i} \right) \right] \\ &\approx \frac{\text{Im}[f(\mathbf{x}_0 + ih_i\mathbf{e}_k + \mathbf{e}_j \Delta x_j)] - \text{Im}[f(\mathbf{x}_0 + ih_i\mathbf{e}_k - \mathbf{e}_j \Delta x_j)]}{2h_i \Delta x_j} \end{aligned}$$

⁶ See Section 1.8.

⁷ See Sections 3.1 and 4.4

⁸ See Sections 2.3 and 4.4

Algorithm 18: ihessian

Hessian of a multivariate, scalar-valued function using the complex-step and central difference approximations.

Given:

- $f(\mathbf{x})$ - multivariate, scalar-valued function ($f : \mathbb{R}^n \rightarrow \mathbb{R}$)
- $\mathbf{x}_0 \in \mathbb{R}^n$ - evaluation point
- $h_i \in \mathbb{R}$ - (*OPTIONAL*) relative step size for the complex-step approximation (defaults to 10^{-200})
- $h_c \in \mathbb{R}$ - (*OPTIONAL*) relative step size for the central difference approximation (defaults to $\varepsilon^{1/3}$)

Procedure:

1. Default the relative step size for the complex-step approximation to $h_i = 10^{-200}$ if not input.
2. Default the relative step size for the central difference approximation to $h_c = \varepsilon^{1/3}$ if not input.
3. Determine n given that $\mathbf{x}_0 \in \mathbb{R}^n$.
4. Preallocate the matrix $\mathbf{H} \in \mathbb{R}^{n \times n}$ to store the Hessian.
5. Preallocate the vector $\mathbf{a} \in \mathbb{R}^n$ to store the absolute step size for each direction k .
6. Populate \mathbf{a} .

```

for  $k = 1$  to  $n$ 
     $a_k = h_c(1 + |x_{0,k}|)$ 
end

```

7. Loop through columns.

```

for  $k = 1$  to  $n$ 

```

```

(a) Imaginary step forward in  $k$ th direction.

$$x_{0,k} = x_{0,k} + ih_i$$

(b) Loop through rows.
    for  $j = k$  to  $n$ 
        i. Real step forward in  $j$ th direction.

$$x_{0,j} = x_{0,j} + a_j$$


$$b = f(\mathbf{x}_0)$$

        ii. Real step backward in  $j$ th direction.

$$x_{0,j} = x_{0,j} - 2a_j$$


$$c = f(\mathbf{x}_0)$$

        iii. Reset  $\mathbf{x}_0$ .

$$x_{0,j} = x_{0,j} + a_j$$

        iv. Evaluate the  $(j, k)$ th element of the Hessian.

$$H_{j,k} \approx \frac{\text{Im}(b - c)}{2h_i a_j}$$

        v. Evaluate the  $(k, j)$ th element of the Hessian
            using symmetry.

$$H_{k,j} = H_{j,k}$$

    end
(c) Reset  $\mathbf{x}_0$ .

$$x_{0,k} = x_{0,k} - \Delta x_k$$

end

```

Return:

- $\mathbf{H} = \mathbf{H}(\mathbf{x}_0) \in \mathbb{R}^{n \times n}$ - Hessian of f with respect to \mathbf{x} , evaluated at $\mathbf{x} = \mathbf{x}_0$

Note:

- This algorithm requires $n(n+1)$ evaluations of $f(\mathbf{x})$ (the upper triangular matrix entries (including the diagonal) consist of $n(n+1)/2$ entries [6], and each entry requires 2 evaluations of $f(\mathbf{x})$).

References

- [1] Yi Cao. *Complex step Hessian*. MATLAB Central File Exchange. Accessed: August 3, 2021. URL: https://www.mathworks.com/matlabcentral/fileexchange/18177-complex-step-hessian?s_tid=srchtitle.
- [2] *complexify.f90*. MDO Lab. Accessed: August 3, 2021. URL: <https://mdolab.engin.umich.edu/misc/files/complexify.f90>.
- [3] *Gradient*. Wikipedia. Accessed: August 3, 2021. URL: <https://en.wikipedia.org/wiki/Gradient>.
- [4] Daniel R. Herberg. *DTQP_hessian_complex_step from DT QP Project*. MATLAB Central File Exchange. Accessed: August 3, 2021. URL: https://www.mathworks.com/matlabcentral/fileexchange/65434-dt-qp-project?s_tid=srchtitle.
- [5] *Hessian matrix*. Wikipedia. Accessed: August 3, 2021. URL: https://en.wikipedia.org/wiki/Hessian_matrix.
- [6] *Is there an analytic expression for a number of elements inside a triangular matrix (with and without items on diagonal)*. Stack Exchange. Accessed: December 26, 2021. URL: <https://math.stackexchange.com/questions/2388887/is-there-an-analytic-expression-for-a-number-of-elements-inside-a-triangular-mat/2388889>.
- [7] *Jacobian matrix and determinant*. Wikipedia. Accessed: August 3, 2021. URL: https://en.wikipedia.org/wiki/Jacobian_matrix_and_determinant.
- [8] Mykel J. Kochenderfer and Tim A. Wheeler. *Algorithms for Optimization*. Cambridge, MA: The MIT Press, 2019. URL: <https://algorithmsbook.com/optimization/>.
- [9] Kok-Lam Lai and John L. Crassidis. “Generalizations of the Complex-Step Derivative Approximation”. In: *AIAA Guidance, Navigation, and Control Conference and Exhibit* (2006). DOI: [10.2514/6.2006-6348](https://doi.org/10.2514/6.2006-6348).
- [10] Joaquim R. R. A. Martins and Andrew Ning. *Engineering Design Optimization*. Cambridge University Press, 2021. URL: <https://mdobook.github.io/>.
- [11] Joaquim R. R. A. Martins, Peter Sturdza, and Juan J. Alonso. “The Complex-Step Derivative Approximation”. In: *ACM Transactions on Mathematical Software* 29.3 (Sept. 2003), pp. 245–262. DOI: [10.1145/838250.838251](https://doi.org/10.1145/838250.838251).
- [12] *Matlab Implementation*. MDO Lab. Accessed: August 3, 2021. URL: <https://mdolab.engin.umich.edu/misc/complex-step-guide-matlab>.
- [13] Cleve Moler. *Complex Step Differentiation*. MathWorks. Accessed: August 3, 2021. URL: <https://blogs.mathworks.com/cleve/2013/10/14/complex-step-differentiation/>.
- [14] *Partial derivative*. Wikipedia. Accessed: April 13, 2022. URL: https://en.wikipedia.org/wiki/Partial_derivative.
- [15] *Partial Derivatives of Vector Valued Functions*. Stack Exchange. Accessed: April 13, 2022. URL: <https://math.stackexchange.com/questions/2823300/partial-derivatives-of-vector-valued-functions>.
- [16] William Squire and George Trapp. “Using Complex Variables to Estimate Derivatives of Real Functions”. In: *SIAM Review* 40.1 (Mar. 1998), pp. 110–112. DOI: [10.1137/S003614459631241X](https://doi.org/10.1137/S003614459631241X).

-
- [17] James Stewart. *Calculus*. 8th. Boston, MA: Cengage Learning, 2015.
- [18] *Vector-valued function (Wikipedia)*. Wikipedia. Accessed: April 13, 2022. URL: https://en.wikipedia.org/wiki/Vector-valued_function.