This document pertians to the function `'trainBust'`

1 definitions

For the purposes to follow, `'trainBust'`has the following properties:

1. Performs 1 step of an implicit-qr algorithm (using householder projections) on a Hessenberg matrix H
2. extracts a series of m numbers upon completion, to be referred to as outputs.
3. takes an input of s shifts to use in the implicit-qr algorithm. to be referred to as inputs.
4. capable of taking an input of size $s \times d$ to represent d seeds for automatic differentiation
5. capable of returning the resulting $d \times 2m$ outputs of the automatic differentiation

For those interested, the extraction of the $2 \cdot m$ numbers is performed by performing a schur decomposition of a submatrix embedded within the larger matrix. This schur decomposition results in a spike (or 2 spikes if perfomed in the middle), the values of which are the m numbers.

In the framework considering the shifts as inputs and the spikes as outputs, we can view this algorith as performing the following transformation

$$T : \mathbb{C}^s \to \mathbb{C}^m$$

Let the shifts be $\vec{\sigma} \in \mathbb{C}^s$ and the spikes be $\vec{m} \in \mathbb{C}^m$. Then we can define the Jacobian as

$$J \in \mathbb{C}^{s \times m} J_{i,j} = \frac{\partial T_i}{\partial \vec{\sigma}_j}$$

Where T_i is the i^{th} output of the function T.

If we provide a seed $\vec{\delta} \in \mathbb{C}^s$ to the automatic differentiation provided in `'trainBust'`, the end result is equivalent to

$$\vec{\delta} \cdot J = [\vec{\delta} \cdot \bigtriangledown T_i]$$

Where the surrounding square brackets indicate that this is a vector over the free indices (in this case i).

2 Automatic Differentiation Performance

Note that

$$\vec{\delta} \cdot J \Big|_{\vec{\sigma}} \approx \left[\frac{T_i(\vec{\sigma} + \varepsilon\vec{\delta}) - T_i(\vec{\sigma} - \varepsilon\vec{\delta})}{2\varepsilon\|\vec{\delta}\|} \right]$$

for $\varepsilon > 0$.

The following results are based upon using $H \in \mathbb{R}^{100\times100}$, H hessenberg, chosen by running a hessenberg decomposition on a matrix chosen from randomly from a normal distribution, $\vec{\sigma}$ chosen from a random normal distribution (multiplied by 100), $\vec{\delta}$ was chosen from a uniform distribution and then normalized, and $s = 8$ (ε is specified below).

Before considering the function T, consider the following two processes:

1. 'trainBust'after it has pushed the bulges through the matrix, but before performing a schur decomposition to create spikes

2. the schur decomposition and its derivative.

Let B be the output of item (1), so $B = B(\vec{\sigma})$. Furthermore, let $\dot{B}$ represent the derivative of B with respect to the normalized direction $\vec{\delta}$. Then

$$\dot{B} \approx \frac{B(\vec{\sigma} + \varepsilon\vec{\delta}) - B(\vec{\sigma})}{\varepsilon}$$

The Schur decomposition is $A = QTQ'$, where the outputs are Q and T. For the direction Δ, we have that the schur decomposition is $A + \varepsilon\Delta = Q_\varepsilon T_\varepsilon Q_\varepsilon$. The `schurAD` fuction calculates $\dot{Q}$ and $\dot{T}$ given Δ. Thus we have that

$$A + \varepsilon\Delta \approx (Q + \dot{Q})(T + \dot{T})(Q + \dot{Q})'$$

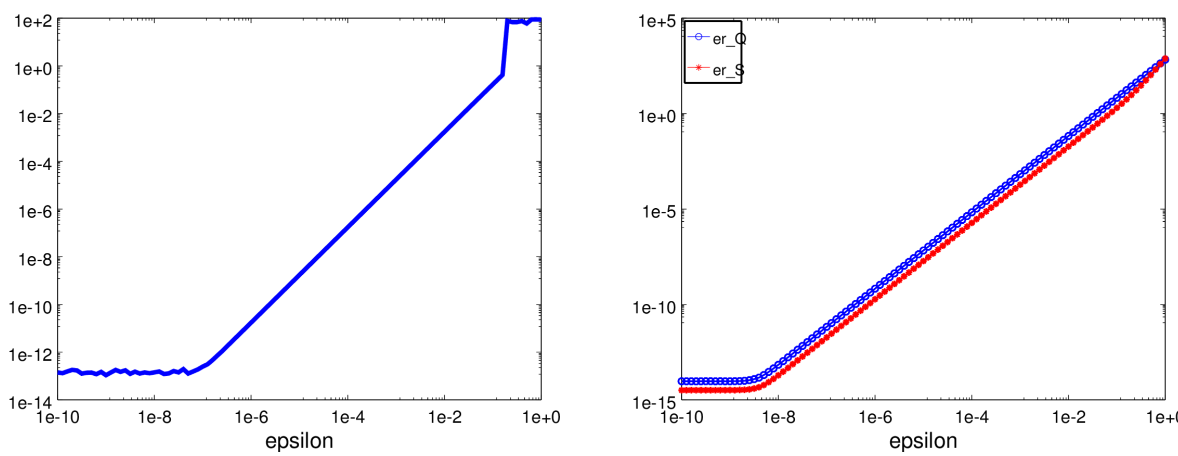
Thus treating the automatic differentiation result as the exact derivative, the above finite difference approximations should yield accuracy of $O(\varepsilon)$. The following equations are used to caculate the error:

$$er_F(\varepsilon) = \frac{\|B(\vec{\sigma} + \varepsilon\vec{\delta}) - (B(\vec{\sigma}) + \varepsilon \cdot \dot{B})\|_F}{\|B(\vec{\sigma})\|_F}$$

$$er_S(\varepsilon) = \frac{\|(Q + \varepsilon\dot{Q})(T + \varepsilon\dot{T})(Q + \varepsilon\dot{Q})' - (A + \varepsilon\Delta)\|_F}{\|A\|_F}$$

$$er_Q(\varepsilon) = \|(Q + \varepsilon\dot{Q})(Q + \varepsilon\dot{Q})' - I\|_F$$

Figure 1 shows that the errors are indeed of $O(\varepsilon)$, indicating that the AD approximation does indeed yield the derivative.



(a) $er_F(\varepsilon)$

(b) $er_S(\varepsilon)$ and $er_Q(\varepsilon)$

Figure 1: Finite difference errors

3 Completing the process

Figure 2 shows er_F applied after the schur decomposition has been used to eliminate the blocks, for 2 different matrices. The matrices used 10 shifts and were size 100.

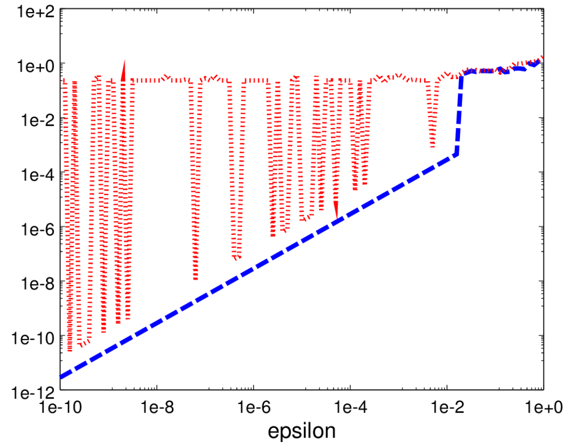


Figure 2: $er_F(\varepsilon)$ applied to completed process

Clearly some matrices break the automatic differentiation as applied/interpreted. The reason can be seen in the equation for er_S as follows. Automatic Differentiation was constructed to accurately represent how the complete schur decomposition changes when the input matrix is changed. But since this decomposition is used inside a larger matrix, the first and last rows form the spikes. Thus we need for the derivatives of each row of the schur decomposition to be 'stable', rather than just the formation of the complete schur decomposition. Note that if the matrix derivatives cannot be reliably predicted, then neither can the derivatives of T.

The following 2 obervations can help alleviate the issues presented by the spikes.

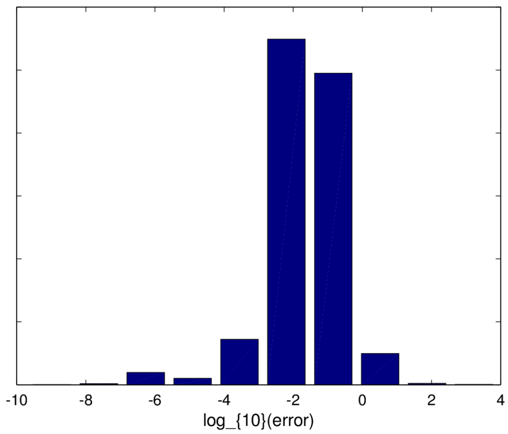
1. Reordering the matrix does not affect the derivatives
2. In the formation of the householder vectors, there is freedom to choose a sign. This is chosen to maximize the stability of the algorithm. The derivatives are 'sign blind' in that they can't account for the change of sign due to it's discontinuous nature.

In light of these two considerations, sort($|\vec{m}|$) should be considered the output of $T(\vec{\sigma})$ rather than $\vec{m}$ (where $\vec{m}$ were the values on the spikes), rather than the spikes themselves.

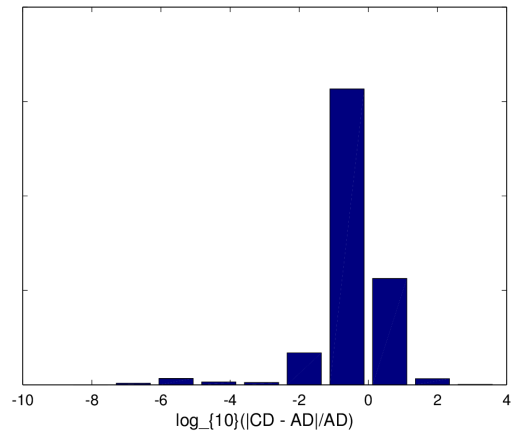
This yields clarity to figure 2. When the derivative drops to the low value expected, the ordering and signs of all the spike values are the same as at $B(\vec{\sigma})$. The poor performance would thus be when one or multiple entries on the spikes have swapped positions or switched signs.

Both of these are important considerations, as changing the input by $\varepsilon = 10^{-8}$ was observed to swap the position of two entries and/or switch an entries sign fairly often. This not only is problemsome in testing the output against central difference schemes, but without considering would yield a function T littered with discontinuities. By sorting and ignoring signs, the function becomes smoother and the derivative provides local information about the function. Furthermore, this has the added benifit that in our derivatives a negative value indicates that the magnitude of the spike entry shrinks, rather than the value itself decreases.

When this was run on 1000 samples, the base-10 logarithm of the error and relative error was examined (thus the number of digits of accuracy in the derivative). Where the accuracy was defined as the central difference approximation compared to the automatic differentiation results from `'trainBust'`, where the central difference was conducted over an interval fo length $2\varepsilon = 2 * 10^{-8}$. It was found that the mean accuracy was -2.2 while the standard deviation in that accuracy was 1.1. Below is included a histogram of the accuracy's observed. Note: the relative accuracy is relative to the automatic differentiation result.



(a) Histogram of the absolute accuracy of the a.d. derivative



(b) Histogram of the relative accuracy of the a.d. derivative

This distribution