Halley Chebyshev Challenge!

Robert Mansel Gower*

November 26, 2014

This is two part challenge surrounding the Halley-Chebyshev methods and a new approach for implementing these methods found in HalleyChebyAD (http://www.maths.ed.ac.uk/~s1065527/). Complete either part, and I will offer you co-authorship on a related article!

1 Challenge I: Empirical evidence using Automatic Derivatives

Halley’s method has been around, and repeatedly rediscovered, since the astronomer Edmond Halley (the same as the infamous comet) conceived it in the 17th century [3]. Despite the methods long history and continued research, it lacks a fundamental credential: Empirical evidence of its effectiveness. With the package HalleyChebyAD it is now possible to run large sets of tests. The challenge is,

Through theory (to follow) and empirical tests (using HalleyChebyAD), encounter a class of function for which the Halley-Chebyshev methods can be tuned to outperform Newton’s method.

Alternatively, show that the Halley-Chebyshev method does not significantly outperform Newton’s method.

It’s a win-win situation! Either a sufficiently negative or positive result is interesting. Now for the theoretical details. From a current iterate \( x \in \mathbb{R}^n \), and a sufficiently differentiable objective function \( f : \mathbb{R}^n \rightarrow \mathbb{R} \), the Halley’s method encounters a next point \( x^* \in \mathbb{R}^n \) by solving \( HA(x^*, x) = 0 \) where

\[
HA(x^*, x) := \nabla f(x) + \left( \nabla^2 f(x) + \frac{1}{2} \nabla^3 f(x) \cdot n(x) \right) \cdot (x^* - x),
\]

where \( n(x) = -\nabla^2 f(x)^{-1} \nabla f(x) \) is the Newton direction. The matrix \( \nabla^3 f(x) \cdot n(x) \) is a contraction of a third-order tensor with a vector. Previously, this has been calculated by obtaining the entire three dimensional tensor, then efficiently contracting with \( n(x) \). In the HalleyChebyAD package, instead, just the directional derivative

\[
\frac{d}{dt} \nabla^2 f(x + tn(x)) \big|_{t=0} = \nabla^3 f(x) \cdot n(x)
\]

is calculated. This can be calculated as efficiently as the derivative \( \nabla^2 f(x) \) [1].

In the 90’s, Gutierrez and Hernandez extend Halley’s method to a one parameter family of methods known as Halley-Chebyshev methods [2]. This is done by a convex combination of Halley’s method and the

---

*contact: University of Edinburgh gowerrobert@gmail.com
Chebyshev step implicitly defined by $CH(x^*, x) = 0$ where

$$CH(x^*, x) := \nabla f(x) + \frac{1}{2} \nabla^3 f(x) \cdot (n(x), n(x)) + \left( \nabla^2 f(x) \cdot (x^* - x) \right).$$

The resulting combined method, for $\lambda(x) \in [0, 1]$ is

$$\lambda(x)CH(x^*, x) + (1 - \lambda(x))HA(x^*, x) = 0.$$

Note that $\lambda(x)$ is a function of $x$. Having a iteration dependent free parameter can be an advantage, but also a pain when you don’t know what to do with it. Iteration dependent parameters have been used to great advantage by tuning a Halley type method for calculating the polar decomposition of a matrix [4]. So much so, this is now the preferred method for calculating the polar decompositions!

The free parameter is probably the key! There is function `void HalleyCheby::set_lambda(const double lambda_in);` ready to be altered within the HalleyChebyshev class in HalleyChebyAD.

## 2 Challenge II: Extending the Halley-Chebyshev family

This extension to allow $\lambda(x)$ to be iterate dependent is only the beginning. In a private article, we have also extended the family to include operator combinations. Let $\Theta \in C^3(\mathbb{R}^n, \mathbb{R}^{n \times n})$, then the following defines an order-3 locally convergent method

$$(I - \Theta(x)) \cdot CH(x^*, x) + \Theta(x) \cdot HA(x^*, x)$$

$$= \nabla f(x) + \left( \nabla^2 f(x) + \frac{\Theta(x)}{2} \nabla^3 f(x) \cdot n(x) \right) \cdot (x^* - x) + \frac{I - \Theta(x)}{2} \nabla^3 f(x) \cdot (n(x), n(x)) = 0. \tag{1}$$

The second challenge is simply

> What criteria should be used for selecting the operator $\Theta(x)$?

Preconditioning? Global progression? Trust region inspired? Many ways to approach the problem, but which one yields an efficient method. Efficient for a class of function or as a general purpose method.

### References


