The construction of transfinite equivalence algorithms.

Possible numerical chaos in error terms of 2^{nd} order quotients.

J.F. Geurdes • Logica Plc Keesomlaan 43 Amstelveen Netherlands • han.geurdes@gmail.com

Structured Abstract

Context: Consistency of mathematical constructions in numerical analysis and the application of computerized proofs in the light of the occurrence of numerical chaos in simple systems. **Purpose:** To show that a computer in general and a numerical analysis in particular can add its own peculiarities to the subject under study. Hence the need of thorough theoretical studies on chaos in numerical simulation. Hence, a questioning of what e.g. a numerical disproof of a theorem in physics or a prediction in numerical economics could mean. **Method:** An algebraic simple model system is subjected to a deeper structure of underlying variables. With an algorithm simulating the steps in taking a limit of second order difference quotients the error terms are studied at the background of their algebraic expression. **Results:** With the algorithm that was applied to a simple quadratic polynomial system we found unstably amplified round-off errors. The possibility of numerical chaos is already known but not in such a simple system as used in our paper. The amplification of the errors implies that it is not possible with computer means to constructively show that the algebra and numerical analysis will 'on the long run' converge to each other and the error term will vanish. The algebraic vanishing of the error term cannot be demonstrated with the use of the computer because the round-off errors are amplified. In philosophical terms, the amplification of the round-off error is equivalent to the continuum hypothesis. This means that the requirement of (numerical) construction of mathematical objects is no safeguard against inference-only conclusions of qualities of (numerical) mathematical objects. Unstably amplified round-off errors are a same type of problem as the ordering in size of transfinite cardinal numbers. The difference is that the former problem is created within the requirements of constructive mathematics. This can be seen as the reward for working numerically constructive.

Key words: Constructive numerical analysis, numerical chaos and transfinite equivalence algorithms.

Introduction

In the philosophy of mathematics there is an ongoing debate about the nature of mathematical object(s). The question 'What is mathematics about' was asked many times (Dummet, 1994). Despite perhaps one's first impression, such a debate is not at all an esoteric and superfluous affair because it makes practical sense to know with what one is dealing: e.g. when venturing into the mathematical unknown.

No big surprise that it turns out we just do not have a clear view on mathematical objects. Moreover, to the practical applied mathematician, it simply seems not to matter very much what a mathematical object exactly is as long as it can be related to the phenomenon that must be described. Pure mathematics is, then, just a way to prepare transformations of the representation of basically unknown objects sometimes venturing in the mathematically unknown. From this perspective it looks as though numbers are 'intentions' to describe a 'picture' such as Wittgenstein's Tractarian facts (Wittgenstein, 1922): i.e. a thought is a logical picture of facts. The question 'what is(are) the object(s) of mathematics' is likely also related to the question 'what is a thought'. Let us also refer to Constable (Constable, 1986). In this paper the question is asked: 'On what grounds is the trust in the use of a computer based'.

Perhaps we could agree for the moment to let a mathematical object be a multifaceted 'something' that shows itself, like a chameleon, depending on its context. A chameleon remains a chameleon but simply changes its colour when the need arises. We know the chameleon because of its colour but we, perhaps, do not need to know the chameleon itself. One can ask the question what the real colour of a chameleon is but that question cannot be answered because we cannot look at a chameleon without a contextual environment. In the author's opinion this biological-like picture comes pretty close to Frege's view that mathematics is not about anything in particular.

In a very crude and 'cutting corners' sense we could state that Finitism and Constructivism can be seen as a kind of ' do not just talk about mathematical objects but try to show them' response to foundational set-theoretical 'miracles' like infinite cardinal numbers with their ordering of 'size'.

The reasoning that the chameleon must have a such and such colour is not the same as showing that there is a chameleon and that it has this colour. Indeed a chameleon can be overlooked because of its colour but there is a difference between the logical mental preparation to catch and to actually have one. It is by no means the intention to say that the 'infinite cardinal colour sequence' of the cardinal number chameleon is impossible. It is only not enough to say 'there must be a chameleon there because this is an outskirt of chameleon territory'. Perhaps in the real animal kingdom one can with a certain probability make this inference but with numbers we just do not know. Judging from the status of the philosophical discussion (Jaquette, 2002), in essence we do not have enough knowledge of mathematical objects to extrapolate into the unknown. Such reasoning seems to make sense. It remains to be seen, however, whether or not the finite also is not haunted by similar unexpected traits.

The first complete philosophical opposition to the implicit approach viz. the derived necessary existence of mathematical objects can be found in Brouwer's ' intuitionism'. For a review on

Brouwer’s position see Brouwer (Brouwer, 1999). He rejected logic as the basis of mathematics and questioned the omnipotence of the law of the excluded middle. A later approach came from E. Bishop who founded constructivism in mathematics. To Bishop a computer program was an ultimate expression of a mathematical truth. The idea ‘ do not just talk about objects but show them’ as a demand for the practical use of the existential quantor ∃ is translated in (extreme form) constructivism by ‘$\exists_x$ can be translated as… there is a computer program that can compute or store x‘. The latter is of course an extreme requirement, because I am also allowed to say $\exists_x$ when I can do the computations etc on a piece of paper. The general idea, however, is clear. Here we will accept the computer program metaphor for $\exists_x$.

Many set theoretical existence theorems turned out to be of great generality but devoid of much computational meaning. In this paper it is asked if all overgeneralization of concepts really vanishes if we employ a computer storage and processing as a metaphor for mathematical objects. Note that in the time of Bishop’s claim of constructivism the computer was still a relatively new phenomenon. Nowadays most people do not live their life without using a computer. Socially a computer brings its own peculiar problems. It is also time to try to find out if a computer also cannot be ‘just a new source of misconception’ in foundational mathematics. Why should the use of a numerical simulation with algorithms or a computer proof (Maietti, 2009) be without its own peculiarities.

Difference-differential quotients

Preliminaries

Finitism and Constructivism in a practical numerical mathematics setting builds from the integers 1,2,3,.... The set $\mathbb{Q}^+$ is created from the set $\mathbb{N}$ of positive integers 1,2,3…, excluding zero, by the elements $q=\left(\frac{n}{m}\right)$ with $n, m \epsilon \mathbb{N}$. The set $\mathbb{Q}^-$ contains the elements $q=-\left(\frac{n}{m}\right)$ with $n, m \epsilon \mathbb{N}$. The set $\mathbb{Q}$ of rational numbers is the union of both sets $\mathbb{Q}^+$ and $\mathbb{Q}^-$. In a computer on the other hand we only have a finite subset of the rational numbers $\mathbb{Q}_{fin}$ to our disposal. Hence, cutting some philosophical corners again, we can say that it looks like Finitism is to Constructivism as numerical analysis is to solving continuous partial differential equations.

Let us start with a simple quadratic function $H_\lambda(x;a)=2\lambda a x+(a^2-1)x^2$. Here, in accordance with the idea that existential and for-all quantors refer to constructible entities, let us suppose that $\lambda, a, x \in \mathbb{Q}_{fin}$ with $a>1$. The employed function somewhat resembles a logistic function where chaos in numerical terms are already being studied (Munkhammar, 2010).

For the function $H_\lambda(x;a)$ let us suppose that $\lambda=\lambda'=-\frac{1}{2a}\sqrt{a^2-1}$. Then it is straightforwardly found that $H_{\lambda'}(x;a)=-y(x)+y(x)^2$, which is $-Q_1(y)$ and $Q_1(y)$ the simple logistic function $y-y^2$. Here, however we do not study $H_\lambda(x;a)$ but the behaviour of its second order difference quotients. In addition, the algorithm does not follow iteration along a logistic map.

In e.g. physical chemistry there is another reference to numerical chaos (Sung, Moon, & Kim, 2001). Let us note beforehand that in the computer (or better in $\mathbb{Q}_{fin}$) there can be two types of numerical errors, truncation and round-off errors. Truncation arises from finite Taylor series. Round-off arises from finite numerical precision.

The constructability of the numbers and the employed functions reside in the fact that they can be created and stored in a computer. Concerning the constructability requirement, the computer is only a numerical aid. In theory, one could also write the numbers down and do the to be given algorithmic operations by hand. It would make our numerical experiments rather tedious but still not impossible.

Mathematical definitions and concepts

For a specific selection of λ this quadratic polynomial can be written in an even more simpler form, namely, when $\lambda = \frac{\sqrt{a^2-1}}{2a}$ we have for $g_\lambda(x;a) = 2\lambda ax = x\sqrt{a^2-1}$ the polynomial $H_\lambda(g) = g + g^2$. Here, $g = g_\lambda(x;a)$. As is well known the differential quotient of $H_\lambda(g)$ depends on the difference quotient

$$\frac{\delta H}{\delta g} = \frac{H(g+\delta g) - H(g)}{\delta g}$$

From differential calculus it is then easy to see that, when $\delta g \to 0$, the first derivative of $H_\lambda(g)$ is $\frac{\mathrm{dH}}{\mathrm{dg}} = \lim_{\delta g \to 0}\left(\frac{\delta H}{\delta g}\right)$ is $1 + 2g$.

As we can see from comparing $H_\lambda(g)$ with $H_\lambda(x;a)$ both δH as well as δg depend on $a, x \in \mathbb{Q}$ when $\lambda = \frac{\sqrt{a^2-1}}{2a}$. In order to make this obvious let us define requirements for to be used $(x;a)$ pairs.

In the first place we are randomly given two rational numbers α and β with $\alpha < \beta$ and select the x such that $\alpha < x < \beta$. Denote this with $x \in (\alpha, \beta)$. In the second place two neighborhood sets $B_n^+(y)$ and $B_n^-(y)$ are defined for random $y \in \mathbb{Q}_{fin}$ as

$$B_n^+(y) = \left\{x \in \mathbb{Q}_{fin} \,\middle|\, 0 < x - y < \frac{1}{n},\ \ n > n_0,\ \ n, n_0 \epsilon \mathbb{N}\ \right\}$$

and

$$B_n^-(y) = \left\{x \in \mathbb{Q}_{fin} \,\middle|\, 0 < y - x < \frac{1}{n},\ \ n > n_0,\ \ n, n_0 \epsilon \mathbb{N}\ \right\}$$

with $B_n(y) = B_n^+(y) \cup B_n^-(y)$.

The previous two definitions allow us subsequently to randomly draw $(x_1;a_1)$ and $(x_2;a_2)$ such that $x_2 \epsilon B_n(x_1)$ and $a_i \epsilon B_n^+(1)$ for $i = 1{,}2$. Based on the two pairs $(x_1;a_1)$ and $(x_2;a_2)$ the difference

$$\delta g(\underline{x};\underline{a}) = g(x_1;a_1) - \ g(x_2;a_2)$$

can be constructed. Here, for brevity, $\underline{x} = (x_1, x_2) \in (\alpha, \beta) \times (\alpha, \beta)$ and of course, $\underline{a} = (a_1, a_2) \in B_m^+(1) \times B_m^+(1)$. Denoting the use of $(x_i; a_i)$ with (i=1,2) in the g functions as index the difference δH is

$$\delta H = (g_1 + g_1^2) - (g_2 + g_2^2)$$

Such that, for $g_2 = g, g_1 = g + \delta g,$ the obvious

$$\frac{\delta H}{\delta g} = 1 + 2g + \delta g$$

follows. Hence, if $(x_1; a_1) \to (x_2; a_2)$ then the same differential quotient would arise from $\left(\frac{\delta H}{\delta g}\right)$.

If we return to the form $H_\lambda(x; a)$ and replace λ with the arithmetic average of the two forms that arise from a_1 and a_2, i.e. $\overline{\lambda} = (\lambda_1 + \lambda_2)/2$, when, $\lambda_i = \frac{\sqrt{a_i^2 - 1}}{2a_i}$, i=1,2, then the difference quotient can be written as

$$\frac{\delta H(\underline{x}; \underline{a})}{\delta g(\underline{x}; \underline{a})} = \frac{\delta H}{\delta g}(\underline{x}; \underline{a}) = 1 + \frac{(a_1^2 - 1)x_1^2 - (a_2^2 - 1)x_2^2}{2\overline{\lambda}(a_1 x_1 - a_2 x_2)}$$

When $a_i \epsilon B_n^+(1)$ for sufficiently large n, the $\overline{\lambda}$ will not vary too much and no great numerical differences for relatively large α and β, are to be expected with: $\frac{\delta H}{\delta g} = 1 + 2g + \delta g$.

Second order differences

In order to obtain a second order difference coefficient it is necessary to select $(y_1; b_1)$ and $(y_2; b_2)$ in addition to the two pairs $(x_1; a_1)$ and $(x_2; a_2)$. In vector notation the following difference can be defined

$$\delta\left(\frac{\delta H}{\delta g}\right)\left(\underline{x}, \underline{y}; \underline{a}, \underline{b}\right) = \frac{\delta H}{\delta g}\left(\underline{y}; \underline{b}\right) - \frac{\delta H}{\delta g}\left(\underline{x}; \underline{a}\right)$$

Similar to $x_2 \in B_n(x_1)$ and $a_i \in B_m^+(1)$ for $i = 1{,}2$ and $n, m \in \mathbb{N}$ we also must have $y_2 \in B_k(y_1)$ and $b_i \in B_l^+(1)$ for $i = 1{,}2$ and $k, l \in \mathbb{N}$.

Subsequently, let us define

$$\delta\overline{g}\left(\underline{x}, \underline{y}; \underline{a}, \underline{b}\right) = \frac{1}{2}\delta g\left(\underline{y}; \underline{b}\right) + \frac{1}{2}\delta g\left(\underline{x}; \underline{a}\right)$$

From the previous two definitions it follows that a difference quotient

$$\frac{\delta\left(\frac{\delta H}{\delta g}\right)}{\delta\overline{g}} = \frac{\delta}{\delta\overline{g}}\left(\frac{\delta H}{\delta g}\right)$$

can be computed. From the definition of $\delta\overline{g}\left(\underline{x},\underline{y};\underline{a},\underline{b}\right)$ we can subsequently derive that

$$\frac{\delta\overline{g}\left(\underline{x},\underline{y};\underline{a},\underline{b}\right)}{\delta g\left(\underline{x};\underline{a}\right)} = \frac{1}{2}\left(1+\frac{1}{\zeta}\right)$$

with ζequal to $\delta g\left(\underline{x};\underline{a}\right)/\delta g\left(\underline{y};\underline{b}\right)$.

Note that, when $\delta\left(\frac{\delta H}{\delta g}\right)$ is $\left(\frac{\delta H}{\delta g}\right)(g+\delta\overline{g}) - \left(\frac{\delta H}{\delta g}\right)(g)$ it follows that $\frac{\delta\left(\frac{\delta H}{\delta g}\right)}{\delta\overline{g}} = 2$.

Multiplying $\frac{\delta}{\delta\overline{g}}\left(\frac{\delta H}{\delta g}\right)$ with $\frac{\delta\overline{g}\left(\underline{x},\underline{y};\underline{a},\underline{b}\right)}{\delta g\left(\underline{x};\underline{a}\right)}$ then gives the following expression

$$\frac{\delta\overline{g}\left(\underline{x},\underline{y};\underline{a},\underline{b}\right)}{\delta g\left(\underline{x};\underline{a}\right)}\frac{\delta}{\delta\overline{g}}\left(\frac{\delta H}{\delta g}\right) = \frac{1}{2}\left(1+\frac{1}{\zeta}\right)\frac{\delta}{\delta\overline{g}}\left(\frac{\delta H}{\delta g}\right) \qquad (*)$$

From equation (*) the left hand side is simply the replacing of $\delta\overline{g}$ in the difference denominator by $\delta g_a = \delta g\left(\underline{x};\underline{a}\right)$. On the right hand side of (*) we have

$$\frac{1}{2\zeta\delta\overline{g}} = \frac{1}{2\delta g\left(\underline{x};\underline{a}\right)\frac{\delta\overline{g}\left(\underline{x},\underline{y};\underline{a},\underline{b}\right)}{\delta g\left(\underline{y};\underline{b}\right)}} = \frac{1}{2\delta g\left(\underline{x};\underline{a}\right)\frac{1}{2}(1+\zeta)} = \frac{1}{(1+\zeta)\delta g\left(\underline{x};\underline{a}\right)}$$

Hence, (*) becomes

$$\frac{\delta}{\delta g_a}\left(\frac{\delta H}{\delta g}\right) = \frac{1}{2}\frac{\delta}{\delta\overline{g}}\left(\frac{\delta H}{\delta g}\right) + \left(1-\frac{\zeta}{1+\zeta}\right)\frac{\delta}{\delta g_a}\left(\frac{\delta H}{\delta g}\right)$$

When, $S = -\zeta/(1+\zeta)$ then the following important equation arises

$$\Delta = \frac{1}{2}\frac{\delta}{\delta\overline{g}}\left(\frac{\delta H}{\delta g}\right) + S(\zeta)\frac{\delta}{\delta g_a}\left(\frac{\delta H}{\delta g}\right) \qquad (**)$$

And algebraically $\Delta = 0$.

Let us, for completeness, tally the requirements we used to obtain equation (**). There are three:

i. $0 < \zeta < 1$, with, $\zeta = \delta g(\underline{x}; \underline{a}) / \delta g\left(\underline{y}; \underline{b}\right)$.

ii. We randomly select $\underline{x}$ and $\underline{y}$ in $(\alpha, \beta) \times (\alpha, \beta)$ such that $\exists_{n \in \mathbb{N}}\ x_2 \in B_n(x_1)$ and $\exists_{m \in \mathbb{N}}\ y_2 \in B_m(y_1)$.

iii. We randomly select the parameters $\underline{a}$ and $\underline{b}$ in $\mathbb{Q}_{fin}{}^{+} \times \mathbb{Q}_{fin}{}^{+}$ such that $\exists_{k \in \mathbb{N}}\ a_i \in B_k^{+}(1)$ and $\exists_{l \in \mathbb{N}}\ b_i \in B_l^{+}(1)$ for $i = 1,2$.

From the definition of the first difference quotient it follows that, $\frac{1}{2} \frac{\delta}{\delta \overline{g}} \left(\frac{\delta H}{\delta \overline{g}} \right) = 1$. This implies, observing equation (**), that for $\delta g_b = \delta g\left(\underline{y}; \underline{b}\right) \to 0$ there will by necessity be $\delta \overline{g} \to 0$ when $0 < \zeta < 1$. Hence from requirement i. It follows, $\delta g_a = \delta g(\underline{x}; \underline{a}) \to 0$. Looking at (**) we see that it can be expected to obtain under this limit condition, $0 < \zeta < 1$ and $\zeta \to 0$, the result $\frac{1}{2} \frac{\delta}{\delta \overline{g}} \left(\frac{\delta H}{\delta g} \right) \to 1$. The $\zeta \to 0$ occurs when $\delta g_a = \delta g(\underline{x}; \underline{a}) \to 0$ faster than $\delta g_b = \delta g\left(\underline{y}; \underline{b}\right) \to 0$.

Algorithm

The convergence of Δ to zero was studied using the following generating algorithm.

i. Initially, n=0, we have $x_1^{(0)} = rnd(0)\ 2^5$ and $x_2^{(0)} = x_1^{(0)} + rnd(0)\varepsilon$, where e.g. $\varepsilon = 0.0001$. The pair $\underline{y}^{(0)} = (y_1^{(0)}, y_2^{(0)})$ was initialized similarly. Note $rnd(0)$ is a random number between 0 and 1.

ii. Initially, n=0, we selected $\underline{a}^{(0)}$ and $\underline{b}^{(0)}$ randomly in $B_m(y_1)$ with e.g. m=1000.

iii. Based on $\underline{a}^{(0)}$ and $\underline{b}^{(0)}$ the $\overline{\lambda}^{(0)} = \frac{1}{2} \sum_{i=1}^{2} \left[\lambda\left(a_i^{(0)}\right) + \lambda\left(b_i^{(0)}\right) \right]$ is computed. Here, $\lambda(z) = \frac{1}{2z} \sqrt{z^2 - 1}$ and the $\overline{\lambda}^{(n)}$ only is computed when $\forall_{z \in \{a_1^{(n)}, a_2^{(n)}, b_1^{(n)}, b_2^{(n)}\}} (\lambda(z) \in \mathbb{R})$ We study real λ. If not then $\overline{\lambda}^{(n+1)} = \overline{\lambda}^{(n)}$.

iv. Hence, $\delta g_a^{(n)} = \delta g(\underline{x}^{(n)}; \underline{a}^{(n)}) = 2\ \overline{\lambda}^{(n)} \left(a_1^{(n)} x_1^{(n)} - a_2^{(n)} x_2^{(n)} \right)$ and $\delta g_b^{(n)} = \delta g\left(\underline{y}^{(n)}; \underline{b}^{(n)}\right)$, similar, can be obtained. When

v. $\delta g_a^{(n)} < 0$, the $x_1^{(n)}$ and $x_2^{(n)}$ are interchanged together with interchanging $a_1^{(n)}$ and $a_2^{(n)}$. For $\delta g_b^{(n)} < 0$, a similar interchange occurs. In addition, if $\delta g_a^{(n)} > \delta g_b^{(n)}$ the $\underline{x}^{(n)}$ and $\underline{y}^{(n)}$ and the $\underline{a}^{(n)}$ and $\underline{b}^{(n)}$ are interchanged.

vi. Subsequently $\delta \overline{g}^{(n)} = \delta \overline{g}\left(\underline{x}^{(n)}, \underline{y}^{(n)}; \underline{a}^{(n)}, \underline{b}^{(n)} \right) = \frac{1}{2} \delta g_a^{(n)} + \frac{1}{2} \delta g_b^{(n)}$.

vii. Together with $\zeta^{(n)} = \frac{\delta g_a^{(n)}}{\delta g_b^{(n)}}$

viii. With the random starting position in$(\underline{x}^{(0)}; \underline{a}^{(0)})$, $\left(\underline{y}^{(0)}; \underline{b}^{(0)}\right)$ the following n=0,1,...N sequence is started for n:

ix. $\begin{cases} x_1^{(n+1)} = x_1^{(n)}/1.5 \\ x_2^{(n+1)} = \frac{1}{a_2}\left(\varepsilon_1^{(n+1)} + \left(a_1 x_1^{(n+1)}\right)\right) \end{cases}$ together with$\varepsilon_1^{(n+1)} = \varepsilon_1^{(n)}/1.001$ and $\varepsilon_1^{(0)} = \varepsilon$

x. $\begin{cases} y_1^{(n+1)} = x_1^{(n+1)} - \delta\overline{g}^{(n)} \\ y_2^{(n+1)} = \frac{1}{b_2^{(n)}}\left(\sqrt{\varepsilon_1^{(n+1)}} + \left(b_1^{(n)} y_1^{(n+1)}\right)\right) \end{cases}$ where the next b vector entries are obtained from

xi. $\begin{cases} b_1^{(n+1)} = a_1^{(n)} - \delta\overline{g}^{(n)} \\ b_2^{(n+1)} = a_2^{(n)} - \delta\overline{g}^{(n)} \end{cases}$ and the $\underline{a}^{(n)} = \underline{a}^{(0)}$ for n=1,2,...N.

xii. The functional form is: $H_{\overline{\lambda}}(x;a) = 2\overline{\lambda}ax + (a^2-1)x^2$ dropping the sub- and superscripts for the moment. Note that $\delta g_a{}^{(n+1)}$, $\delta g_b{}^{(n+1)}$ and $\delta\overline{g}^{(n+1)}$ are computed from ix. , x. and xi. but using the conditions already mentioned in iii., iv. and v.

xiii. Subsequently, we compute $\left(\frac{\delta H}{\delta g}\right)\left(\underline{x}^{(n+1)};\underline{a}^{(n+1)}\right) = \frac{H\left(x_1^{(n+1)};a_1^{(n+1)}\right) - H\left(x_2^{(n+1)};a_2^{(n+1)}\right)}{\delta g_a{}^{(n+1)}}$ and

xiv. $\left(\frac{\delta H}{\delta g}\right)\left(\underline{y}^{(n+1)};\underline{b}^{(n+1)}\right) = \frac{H\left(y_1^{(n+1)};b_1^{(n+1)}\right) - H\left(y_2^{(n+1)};b_2^{(n+1)}\right)}{\delta g_b{}^{(n+1)}}$ and with the use of the two first order difference quotients the necessary second order quotients from equation (*) are obtained .

xv. $\frac{\delta}{\delta\overline{g}}\left(\frac{\delta H}{\delta g}\right)\left(\underline{x}^{(n+1)},\underline{y}^{(n+1)};\underline{a}^{(n+1)},\underline{b}^{(n+1)}\right) = \frac{\left(\frac{\delta H}{\delta g}\right)\left(\underline{y}^{(n+1)};\underline{b}^{(n+1)}\right) - \left(\frac{\delta H}{\delta g}\right)\left(\underline{x}^{(n+1)};\underline{a}^{(n+1)}\right)}{\delta\overline{g}^{(n+1)}}$ together with

xvi. $S\left(\zeta^{(n+1)}\right) = \frac{-\zeta^{(n+1)}}{1+\zeta^{(n+1)}}$ with

xvii. $S\frac{\delta}{\delta g_a}\left(\frac{\delta H}{\delta g}\right)\left(\underline{x}^{(n+1)},\underline{y}^{(n+1)};\underline{a}^{(n+1)},\underline{b}^{(n+1)}\right) = S\left(\zeta^{(n+1)}\right)\frac{\left(\frac{\delta H}{\delta g}\right)\left(\underline{y}^{(n+1)};\underline{b}^{(n+1)}\right) - \left(\frac{\delta H}{\delta g}\right)\left(\underline{x}^{(n+1)};\underline{a}^{(n+1)}\right)}{\delta g_a{}^{(n+1)}}.$

xviii. Then $n \to n+1$ and go to iii. until a sufficient number of iterations has been performed that simulate taking the limit: $\delta g_a = \delta g(\underline{x};\underline{a}) \to 0$ faster than $\delta g_b = \delta g\left(\underline{y};\underline{b}\right) \to 0$.

xix. Only the data points that produces $\frac{\delta}{\delta\overline{g}}\left(\frac{\delta H}{\delta g}\right)\left(\underline{x}^{(n+1)},\underline{y}^{(n+1)};\underline{a}^{(n+1)},\underline{b}^{(n+1)}\right) \approx 2$ are displayed.

The display condition xix can be justified noting that theoretically we expect the second order differential to be equal to 2.

Results

Here the results of the algorithm of the previous section are presented. In the first place it is important to note that the requirements belonging to equation (**) are obeyed in the numerical computations. We found that in a typical run the δg_a, δg_b and $\delta\overline{g}$ behaved as expected (see the figure below). Here , $\delta g_b \to 0$ is the driver of the limit. Note that the x-axis shows the iteration step whereas the y axis in the figures below represent the to-be investigated entities from the algorithm.

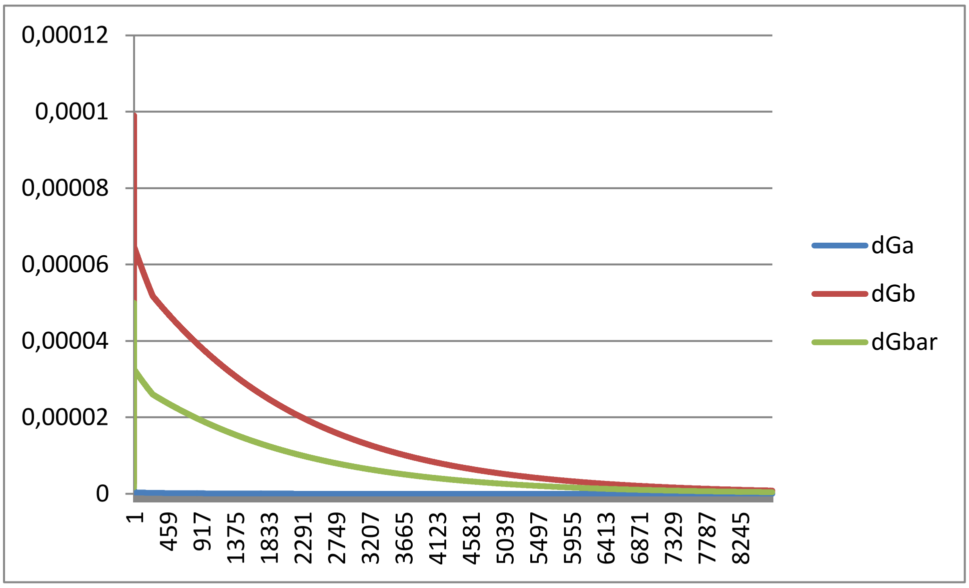

Figure 1 Check on the required behaviour of the δg_a, δg_b and $\delta \overline{g}$

Concerning the ζ its behaviour in the computations confirms the expectations from the figure above and with this we think the algorithm complied to requirement i.

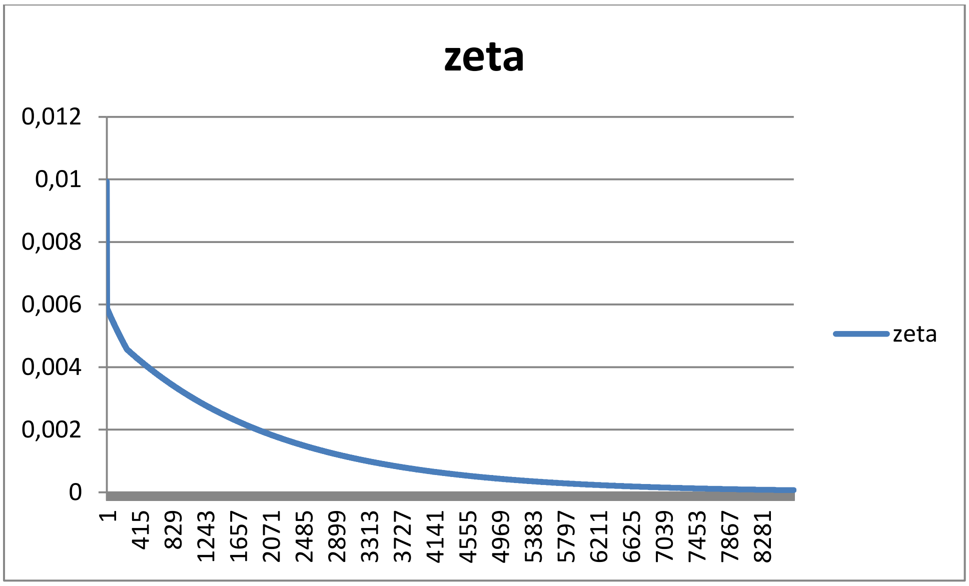

Figure 2 The behaviour of the ζ in the numerical computations.

Concerning the second requirement, i.e. $\underline{x}$ and $\underline{y}$ in $(\alpha, \beta) \times (\alpha, \beta)$ such that $\exists_{n \in \mathbb{N}}\ x_2 \in B_n(x_1)$ and $\exists_{m \in \mathbb{N}}\ y_2 \in B_m(y_1)$ from the following to figures it is clear that the algorithm did not violate ii. For the x we see the following figure.

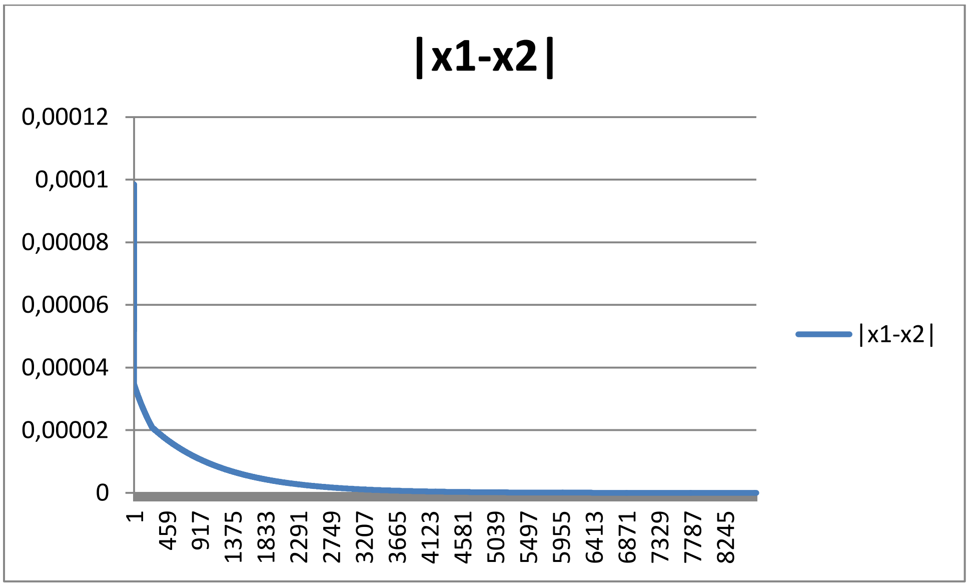

Figure 3 Behaviour of the x variables in the algorithm.

For the y variables we obtain in a typical run .

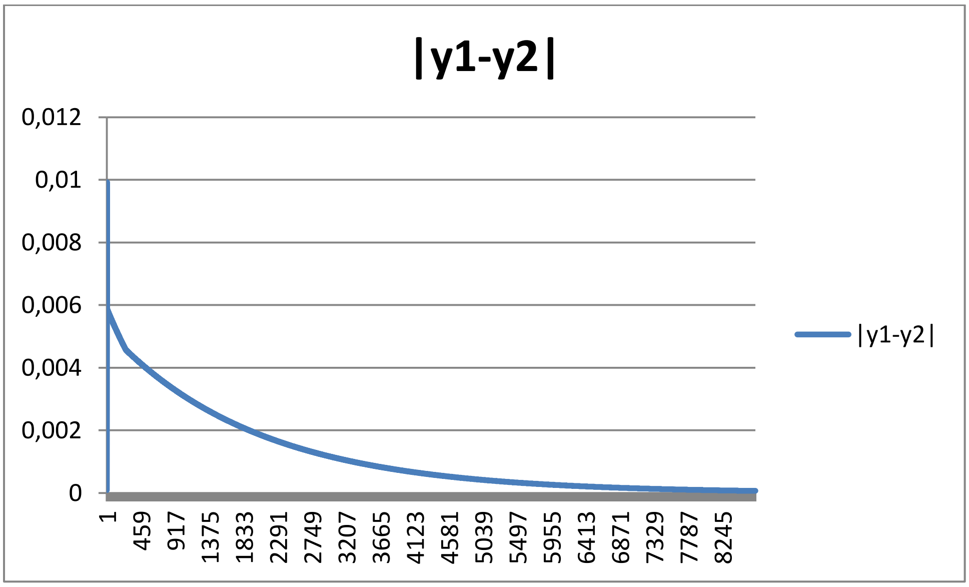

Figure 4 Behaviour of the y variables in the algorithm.

Hence, we may conclude that requirement ii. is not violated in the algorithm.

The third requirement is that the b vector remains within close range of unity or, equivalently , $\underline{a}$ and $\underline{b}$ in $\mathbb{Q}_{fin}{}^{+} \times \mathbb{Q}_{fin}{}^{+}$ such that $\exists_{k\in\mathbb{N}}\ a_i \in B_k^+(1)$ and $\exists_{l\in\mathbb{N}}\ b_i \in B_l^+(1)$ for $i = 1{,}2$.

For b1 we show

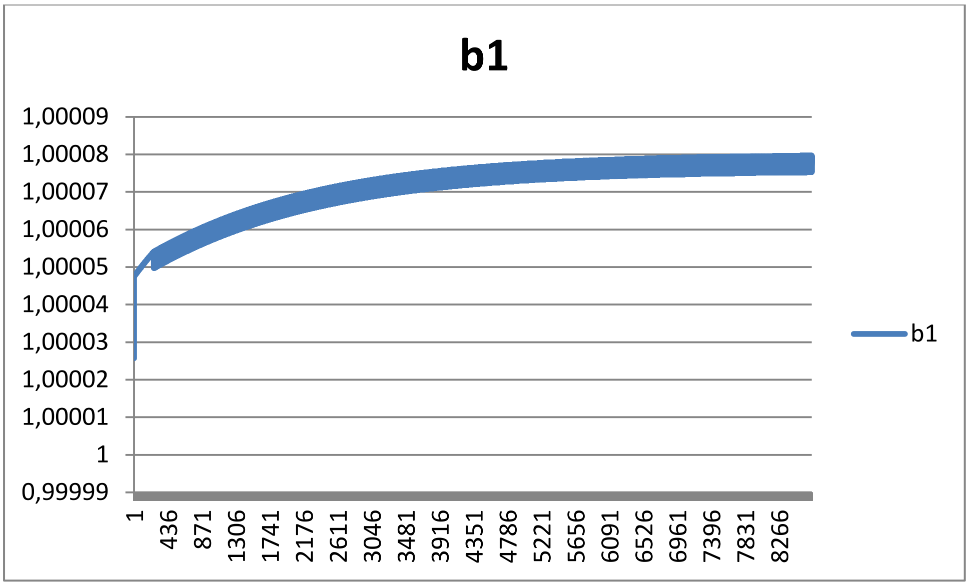

Figure 5 The requirement that b vector remains within close range of unity.

The b2 behaves similarly. Hence the algorithm also complies to requirement iii.

The important result is that the numerical trend in equation (**) is that $\Delta \to \Delta_0$ and $\Delta_0 \approx -0.0005$.

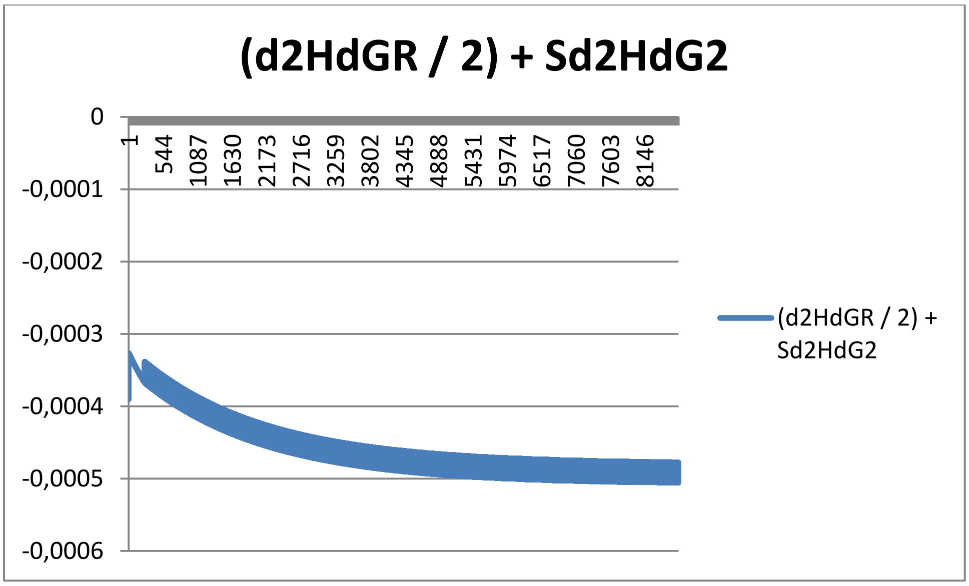

Figure 6 Iteration steps n=1,...N (N=8000) for Δ from equation (**).

This result is remarkable because for increasing step in the iteration or, equivalently, a next step in the limit $0 < \delta g_a < \delta g_b \to 0$ under the requirements i., ii. and iii., we see the Δ reaching a asymptote $\Delta \to \Delta_0$ and $\Delta_0 \approx -0.0005$ instead of steadily approximating to zero.

The impression of reaching a asymptote for $\Delta \to \Delta_0$ and $\Delta_0 \approx -0.0005$ is supported by a run with $1.99 < d2HdGR = \frac{\delta}{\delta g}\left(\frac{\delta H}{\delta g}\right)\left(\underline{x}^{(n+1)}, \underline{y}^{(n+1)}; \underline{a}^{(n+1)}, \underline{b}^{(n+1)}\right) < 2.01$. We sampled $N = 125$ from the $M = 1 \times 10^6$ total data points. For $\Delta = \left(\frac{\mathrm{d2HdGR}}{2}\right) + \mathrm{Sd2HdG2}$ which is the numerical equivalent of equation (**) we found

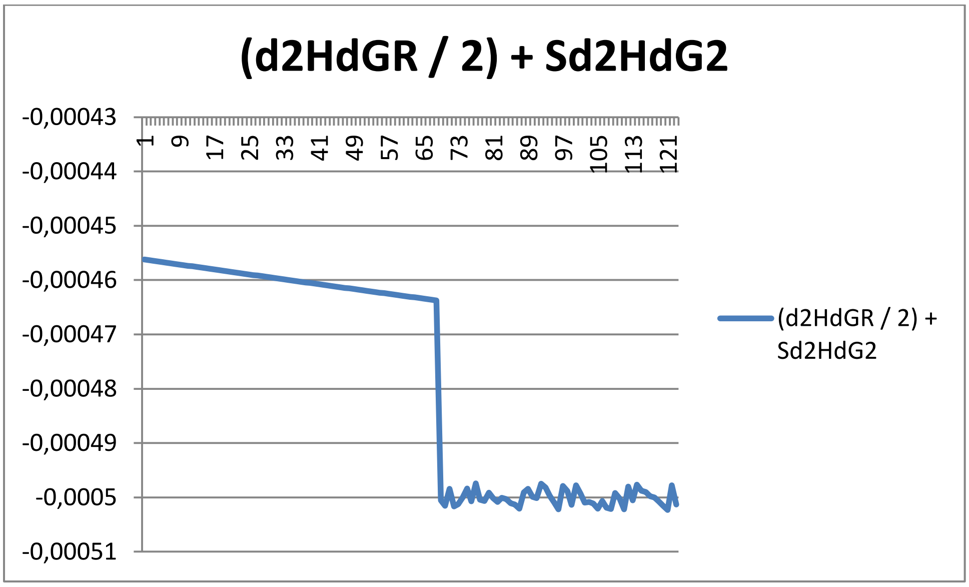

Figure 7 Result for $\Delta = \left(\frac{\mathbf{d2HdGR}}{2}\right) + \mathbf{Sd2HdG2}$ in a run with $M = 1 \times 10^6$ total number data points resulting in a sample of $N = 125$ with d2HdGR in the interval (1.99,2.01).

It is not shown in the previous figures but it is obvious from the algorithm that $\varepsilon_1^{(n)} \to 0$ and that $\left(x_1^{(n)}, x_2^{(n)}\right) \to (0,0)$ and $\left(y_1^{(n)}, y_2^{(n)}\right) \to (0,0)$.

Conclusion and discussion

In the paper a numerical trend, presented in figures 6 and 7, of the Δ of equation (**) was found that runs counter to the basic algebra of the difference quotient. The observed trend is not accidental and appears to be similar nearly every time the (randomly starting) algorithm is run. The requirements in the iteration are complied in the algorithm as can be seen from the figures in the previous section. Moreover, the exclusive display of data points having $\frac{\delta}{\delta g}\left(\frac{\delta H}{\delta g}\right)\left(\underline{x}^{(n+1)}, \underline{y}^{(n+1)}; \underline{a}^{(n+1)}, \underline{b}^{(n+1)}\right) \approx 2$ can be justified because for those points comes closest to the differential quotient that one aims at in the iterative numerical equivalent of taking the limit: $0 < \delta g_a < \delta \overline{g} < \delta g_b \to 0$.

In a run of 10^4 iterations, data points following the display condition $1.5 < d2HdGR < 2.5$ occur approximately 2000 times. In a run of 10^6 iterations, data points following the display condition $1.99 < d2HdGR < 2.01$ occur 125 times.

In addition it should be noted from equation (**) it straightforwardly follows that mathematically $\Delta = 0$. Perhaps superfluous, it also follows from the fact that $S(\zeta)\frac{\delta}{\delta g_a}\left(\frac{\delta H}{\delta g}\right) = -\frac{1}{1+\zeta}\frac{\delta}{\delta g_b}\left(\frac{\delta H}{\delta g}\right)$ because ζequal to $\delta g(\underline{x};\underline{a})/\delta g\left(\underline{y};\underline{b}\right)$. It then is easy to see that $(1+\zeta)\delta g_b = 2\delta\overline{g}$ because $\delta\overline{g} = \frac{1}{2}(\delta g_a + \delta g_b)$. Hence, $S(\zeta)\frac{\delta}{\delta g_a}\left(\frac{\delta H}{\delta g}\right) = -\frac{1}{2}\frac{\delta}{\delta\overline{g}}\left(\frac{\delta H}{\delta g}\right)$ and from (**) we have that $\Delta = 0$.

Witnessing figures 6 and 7 we did not find a steadily convergence of $\Delta^{(\mathrm{n})}$ to zero for increasing iteration step $n = 1,2,\ldots.N$. No doubt the argument comes to mind that the recurrence of the trend $\Delta \to \Delta_0$ and $\Delta_0 \approx -0.0005$ under the conditions showing the 'favoured' data points is a, principal, flaw somewhere in the computations of difference quotients not inherent in the data of the function. However, what kind of flaw would that be?

If we assume there was no error made in the translation of the algorithm to the program, then the finitude of the precision of the computer comes to mind as being the possible source of the persistence of the error $\Delta \neq 0$. In the simulation we employed the IEEE 754-2008 Standard for Floating-Point Arithmetic. This standard is the most widely used for floating point computations. Moreover, although it is not constructive evidence, it can be argued that an asymptote-Δ, like in e.g. figure 7, will recur under better precision conditions but only later in the sequence because the precision of the computer is higher.

Furthermore but no surprise, it makes no difference if one runs the algorithm in VBA on a Toshiba (with Windows 7) or a HP (using Windows Vista) computer.

A second argument against the correctness of the conjectured asymptote-Δ is that it indeed makes a difference which algebraic expression is used in the algorithm. With the present function, $H_\lambda(x;a) = 2\overline{\lambda}ax + (a^2-1)x^2$ one can verify that for α of the order of 1×10^{148} and β of the order of 1×10^{150} the first order difference quotient in the form

$$\left[\frac{\delta H}{\delta g}\left(\underline{x};\underline{a}\right)\right]_{Stable} = 1 + \frac{(a_1^2-1)x_1^2 - (a_2^2-1)x_2^2}{2\overline{\lambda}(a_1x_1 - a_2x_2)}$$

is stable and produces values around unity when conditions i., ii. and iii. of the previous section are observed. However, the algebraic equivalent

$$\left[\frac{\delta H}{\delta g}\left(\underline{x};\underline{a}\right)\right]_{Unstable} = 1 + \frac{a_1x_1 + a_2x_2}{2\overline{\lambda}} + \frac{(x_2-x_1)(x_2+x_1)}{2\overline{\lambda}(a_1x_1 - a_2x_2)}$$

is unstable and produces a, ' $\pm\infty$ ', fluctuating result such that $\nexists_{n\in\mathbb{N}}\left[\frac{\delta H}{\delta g}\left(\underline{x};\underline{a}\right)\right]_{Unstable} \in B_n(1)$. For the stable form we do have $\exists_{n\in\mathbb{N}}\left[\frac{\delta H}{\delta g}\left(\underline{x};\underline{a}\right)\right]_{Stable} \in B_n(1)$. Hence, algorithms obtained from the function $H_\lambda(x;a)$ are vulnerable to the employed algebraic form.

However, the asymptote-Δ conjecture for equation (**) is a stable numerical recurring trend in the numerical simulation of the quotients.

Another argument against $\Delta \to \Delta_0$ ($\Delta_0 \approx -0.0005$) could be that the algorithm is a bad way to numerically approximate a second order derivative. It should be noted that the algorithm obeys $\delta g_a = \delta g(\underline{x};\underline{a}) \to 0$ faster than $\delta g_b = \delta g\left(\underline{y};\underline{b}\right) \to 0$ such that $\delta\overline{g} \to 0$. In the body of the paper we already showed that on the level of g-variables in $H_\lambda(g) = g + g^2$ the expected second order derivative readily obtains. As a check in the algorithm we, in addition, observe the second order difference quotient $d2HdGR$ approaching the value of 2 in the recurrent limit process of the algorithm.

In case of the restriction $1.99 < d2HdGR < 2.01$ the following figure shows two stages in the iteration where $d2HdGR$ is in the interval (1.99,2.01).

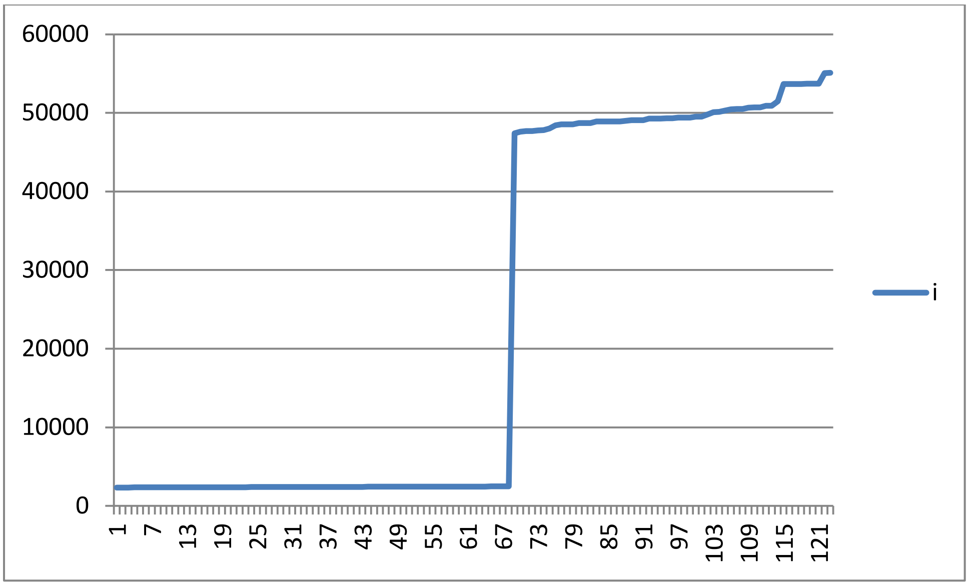

Figure 8 In the restrictive condition $\mathbf{1.99} < d2HdGR < 2.01$ the approximate first 70 selected data points run from the 2350-th to 2480-th iteration step. The rest obeying the condition $\mathbf{1.99} < d2HdGR < 2.01$ occur between the 47417-th and the 55081-th step in the iteration.

From figure 8 we learn that, in this computation, at least two times the algorithm came $1.99 < d2HdGR < 2.01$ close to computing the second order derivative of $H_\lambda(g) = g + g^2$.

Moreover, in constructivism the computer is tacitly supposed to be a passive numerical aid. Indeed the algorithmic results for $\left[\frac{\delta H}{\delta g}(\underline{x};\underline{a})\right]_{Stable}$ and $\left[\frac{\delta H}{\delta g}(\underline{x};\underline{a})\right]_{Unstable}$ is already a caution to a numeric constructivist proof. But instability of computations is easily detected while the recurrence of a $\Delta \to \Delta_0$ and $\Delta_0 \approx -0.0005$ appears to point at a more subtle numerical phenomenon. At its least the possibility of similar phenomena like $\Delta \to \Delta_0$ and $\Delta_0 \approx -0.0005$ should be recognized as a possible nuisance in the use of certain numerical recipes both in constructive as well as in applications like financial mathematics. A financial mathematical application that may come to mind and is the pricing of options. In the latter case, underlying variables e.g. our $(\underline{x};\underline{a})$ variables, can be underlying economic entities influencing the behaviour of the price of the option, e.g. expressed in $g(\underline{x};\underline{a})$.

If we accept the conjecture $\Delta \rightarrow \Delta_0$ and $\Delta_0 \approx -0.0005$, it can be stated that constructivist and numerical analysis may explore the unexpected where algebra and analysis lead to the obvious, but necessary, basic conclusion. Although a vanishing of the asymptote-Δ 'on the long run' is to be expected from the algebra, there is no numerical proof constructed yet that for very small δg_b in the given algorithm, the curve will rise to zero such as expected from algebra and analysis. This poses a problem for the idea that 'each constructive result T can be realised as a computer program requiring minimal preparation.... ' (Bishop, 1967).

Moreover, how is one to 'proof' the vanishing of the asymptote in $\Delta \rightarrow \Delta_0$, i.e. $\Delta_0 = 0,$ in a strict mathematical constructivism? Looking at equation (**) it is obviously true but the algorithm suggests that we philosophically could be talking about the invisible colour of the chameleon like we perhaps are doing with the infinite cardinal numbers and their implicit order. Agreed we could run longer chains of computation. However, $M = 10^6$ iterations already is quite substantial and the chance arises that the numerical activities of the computer loose meaning because of a finite numerical precision.

We attempted a test run of $M = 10^7$ and $1.999 < d2HdGR < 2.001$. 18 data points were found and the behavior still is towards an asymptote (see figure below)

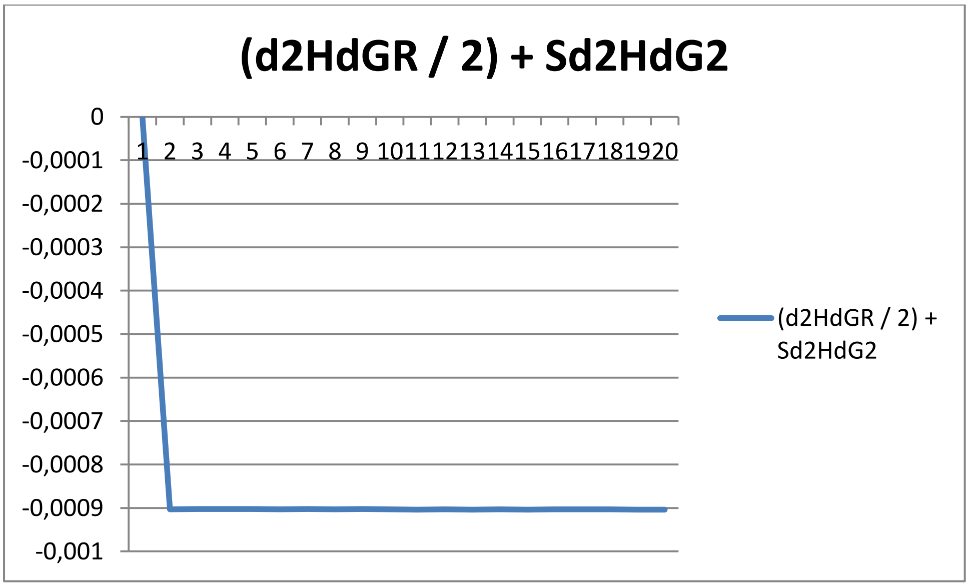

Figure 9 Behaviour of $\Delta \rightarrow \Delta_0$ with total number of iterations $M = 10^7$ in the restriction $1.999 < d2HdGR < 2.001$.

Note that from Figure 9, we can observe that $\Delta_0 \approx -0.0009$ instead of $\Delta_0 \approx -0.0005$ as in the previous analyses. In that respect, it should also be noted that, if in the algorithm, the power, p, of the epsilon in

$$y_2^{(n+1)} = \frac{1}{b_2^{(n)}}\left(\left\{\varepsilon_1^{(n+1)}\right\}^p + \left(b_1^{(n)} y_1^{(n+1)}\right)\right)$$

then the asymptote and appears to be $\Delta_0 \approx \left(\frac{-\text{p}}{1000}\right)$, for at least some of the $0 < p < 1$ in some of the parameter choices of the algorithm described previously, given the display $1.99 < d2HdGR < 2.01$.

The question can now be raised if numerical results faithfully follow the mathematics that is intended by the (constructivist) theorist. The 'minimal preparation' part of Bishop's claim, at least those were numerical analysis is used, can already be questioned with the 'numerical instability versus algebraic equivalence' argument. But we showed it can become more subtle without a constructivist or a builder of e.g. physics or numerical economical models, noticing it. Note that especially with economical models there is a substantial possibility of having deeper structures (our, $\underline{x}$ and $\underline{a}$ variables) that co-determine the behaviour of more global variables (our, $g = g(\underline{x}; \underline{a})$) in a law (our $H_\lambda(g)$).

Apparently, numerical simulation sometimes can add its own peculiarities to the topic one is researching. This is also reported in other numerical studies more in relation to chaotic dynamical systems. For instance computed non-periodic solutions of chaotic differential equations are the consequence of unstably amplified round-off errors, and are not approximate solutions of the associated differential equations (Yao, 2005). Our system is relatively simple in relation to the Lorentz equations that Yao is discussing. Interestingly, we select our initial values $\underline{x}^{(0)}, \underline{y}^{(0)}, \underline{a}^{(0)}$ and $\underline{b}^{(0)}$ at random, but admittedly within restrictions, and still observe the recurrence of the asymptote in the error term. To some this excludes 'chaos' but we must note that the parameterization influences the nature of the asymptote in error term. Of course we are not dealing with numerical integration of partial differential equations like Yao but because of numerical behaviour of $\frac{\delta}{\delta g_a}\left(\frac{\delta H}{\delta g}\right)$ in the following figure

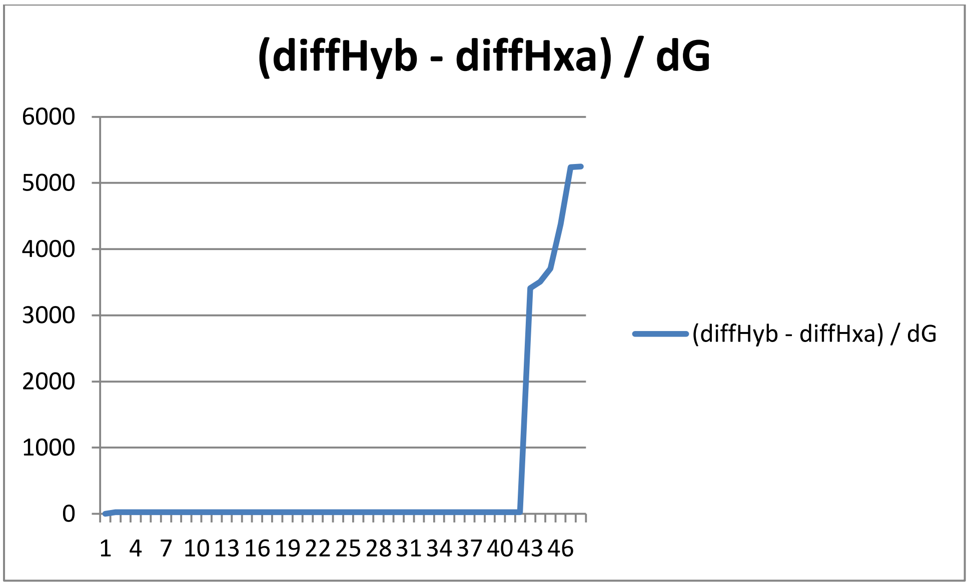

Figure 10 Numerical behaviour of $\frac{\delta}{\delta g_a}\left(\frac{\delta H}{\delta g}\right)$ in the algorithm.

it makes sense to suspect that chaos in the numerical computation of the error is the cause of the asymptote in our simple second order difference system. Yao also argues against the dependence on the initial values as a common sign for numerical chaos.

Implications for philosophical problems in physics modelling

The possibility of unstably amplified round-off errors (such as in figure 10) also touches upon a point which, according to the present author, can be described as the 'still to be better qualified' cry for computer program 'proof' in Bell theorem related foundational physics.

Of course, it all depends on what one wants to consider to be proof in one's field. However, classical mathematical physics proof of the incompleteness of Bell's theorem already was given (Geurdes, 2010) and (Geurdes, 2011). Nevertheless the scientists involved insist on numerical proof. In view of our results on a very simple system, they might be overlooking the peculiarities of a numerical experimentation. The possibility or impossibility of a numerical disproof of Bell's theorem, as can be seen from our conjecture $\Delta \to \Delta_0$ and $\Delta_0 \approx \left(\frac{-\mathrm{p}}{1000}\right)$, is perhaps also not the philosophers stone.

Moreover, there appears an uneven handed approach to a mathematical disprove of Bell's theorem. The experimentalists and physics philosophers obvious did not require any numerical proof before accepting the philosophically equally feeble explanation 'non-locality' in quantum theory. The idea that a computer is *equivalent* to a physical experiment is highly questionable because a computer only may work with finite numbers in a certain precision. Bell's theorem has a broader scope (Bell, 1964) and because we do not know nature, locality in a domain that escapes computer simulation but within the boundaries of the theorem, cannot be excluded.

Constructivist numerical analysis

Perhaps the demand for computer simulation in foundational physics also touches upon the use of computations that could create its own sphere of 'magic'. The present paper shows that numerical simulation may add its own peculiarities to a computer view of mathematical objects and blur its purity of representation. In practical numeric analysis the phenomenon of asymptote error terms is known. The reader is referred to e.g. the construction of the Crank-Nicholson method for the numerical solution of the one-dimensional heat equation (Ames, 1992). However, a thorough investigation in the role of the persistence of numerical errors in theoretical numerical analysis appears necessary. In a single phrase: Unstably amplified round-off-error chameleons thrive in $\mathbb{Q}_{fin}$.

References

Ames, W. (1992). *Numerical Methods for Partial Differential equations (3rd ed).* Boston: Acad. Press. Inc.

Bell, J. (1964). On the Einstein, Podolsky and Rosen paradox. *Physics* , 195-200.

Bishop, E. (1967). *Foundations of Constructive Analysis.* N.Y.: MacGrawHill.

Brouwer, L. (1999). Intuitionsm and Formalism. *Bull (New Series) Am. Math. Soc. 37(1)* , 55-64.

Constable, R. (1986). *Lectures on: Classical proof as programs.* Ithaca: Cornell University.

Dummet, M. (1994). What is mathematics about? In G. Alexander, *Mathematics and Minds* (pp. 11-26). Oxford: Oxford Univ. Press.

Geurdes, J. (2011). A note on Hardy paradox. *Adv. Studies Theor. Phys. 5(11)* , 545-549.

Geurdes, J. (2010). CHSH and local hidden causality. *Adv. Studies Theor. Phys. 4(20)* , 945-949.

Jaquette, D. (2002). Introduction: Mathematics and Philosophy of Mathematics. In D. Jaquette, *Philosophy of mathematics:an anthology* (pp. 1-12). Cornwall: MPG Books Ltd.

Maietti, M. (2009). *A minimalist two-level foundation for constructive mathematics .* Padua: Univ. of Padua.

Munkhammar, J. (2010). *Numerical chaos in fractional order logistic map.* N.Y.: arXiv: gen-math.1011.2389v1.

Sung, B., Moon, J., & Kim, M. (2001). Checking the influence of numerically induced chaos in the computational study of intramolucular dynamics. *Chem. Phys. Lett. 342* , 610-616.

Wittgenstein, L. (1922). *Tractatus Logico Philosophicus.* New York, Oxon: Routledge & Keegan Paul Ltd.

Yao, Y. S. (2005). *Computed Chaos or Numerical Errors.* N.Y.: arXiv:nlin/0506045.