Not for publication, June 18, 2016

A general approach to Bürgi's *artificium*

E.Engeler

Abstract

Bürgi's method for the computation of the sine function is based mathematically on the fact that this function satisfies a differential equation. The solution is approximated by a sequence of functions computed simutaneously at discrete values. This observation motivates a generalisation by which we obtain a massively parallel algorithm for second-order differential equations in general. We also show the psychology of invention may have originally lead to these approximations. Since the limit of the sequence is close to, but does not coincide with the solution, Bürgi's *artificium* consists of "normalizing" the approximation by a constant factor, whereby he obtains the exact values as discussed in [2]. This, however does not generalize.

1. Approaching Euler (with a file)

Bürgi was a Swiss watchmaker, astronomer and mathematician. He was a perfect artisan who created ingenious, beautifully finished pieces of mathematical and astronomical instruments [1], and also very original mathematical algorithms [2].

Imagine him as an instrument-maker, how he worked on a piece of metal, roughly of the shape desired, and file it down going over the the workpiece time and time again. Each time he would inspect the *shape* of the piece at each place, that is its present slope there, and there file away by an amount that depends on how far from the desired slope it is at this place, until he is satisfied. This, I imagine might be the mindset behind his computing

[1]On display at the Swiss National Museum in Zurich, and as illustrations in the beautifully produced book [1] by F. Staudacher.

[2]Logarithms and, here to be discussed, trigonometric functions.

the shape of the sine curve.

A legitimate mathematical interpretation of this work-plan for the given *shape* is to view it as a curve $y = g(x)$ given by a differential equation. We start by considering a *first-order differential equation* $y' = f(x, y)$ and assume that f satisfies a Lipschitz condition in a domain D, e.g. $|f(x, y_1) - f(x, y_2)| < c|y_2 - y_1|$ for some constant c and for all $(x, y_1), (x, y_2) \in D$. As is well known, the equation then has a unique solution for any given initial value $(x_0, y(x_0))$ in D.

Let us now choose an arbitrary function g_0, with initial value $g_0(a) = y(a)$, lying within the domain D as a proposed approximate solution for the given equation. Let $x_0, x_1, \ldots, x_N$ be a discretisation of the interval for which we seek the solution, and let $d_n = x_{n+1} - x_n$ denote the sizes of the respective intervals. With the values $g_0(x_0), g_0(x_1), \ldots$ we now start a sequence of functions $g_0(x), g_1(x), g_2(x), \ldots$ with which we intend to approach the solution $y(x)$:

$$g_{t+1}(x_{n+1}) = g_t(x_n) + d_n f(x_n, g_t(x_n)), n = 0, 1, \ldots, N - 1, t = 0, 1, \ldots, N - 1.$$

This is meant to correspond to the filing away at the workpiece with a straight file, except that the successive modifications might involve adding material at places where $f(x_n, g_t(x_n))$ is positive. So perhaps we should better imagine a sculptor working on wet clay with a flat spatula. Then the process is immune to taking away too much and ruining the workpiece.

What is the shape obtained by this process ? Consider the Euler solution of the given differential equation, given the above discretisation and condition. It would proceed as follows:

$$e_0(x_0) = y(a), e_0(x_{n+1}) = e_0(x_n) + d_n f(x_n, e_0(x_n)), n = 0, 1, \ldots, N - 1.$$

Instead of starting the Euler approximation at x_0 with initial value $y(x_0)$, we could also start it simultaneously at all points $(x_n, g_0(x_n))$ on the proposed first approximation curve g_0, that is we look at the functions $e_n(x_m)$ defined by

$$e_n(x_n) = g_0(x_n), e_n(x_{m+1}) = e_n(x_m) + d_m f(x_m, e_n(x_m)), m = n, n + 1, \ldots, N - 1.$$

Now notice that

$$g_t(x_n) = e_{n \dot{-} t}(x_n) \quad \text{for} \quad n, t = 0, 1, \ldots, N,$$

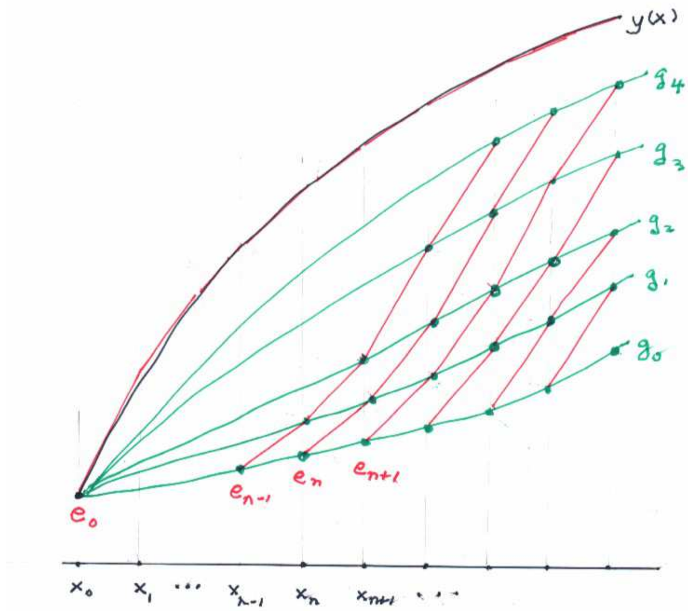

Figure 1: Approaching Euler

where $n \dot{-} t$ is 0 for $t > n$, $n - t$ else. For $t = N$ we therefore have $g_N(x_n) = e_0(x_n)$. The algorithm converges to the Euler solution for $y(x)$ with the given initial value.

2. Generalization

The artisan knows that *shape* also means *curvature*; he goes over the piece time t and time $t + 1$ again and remembers at each place the present curvature, compares it with the desired one and modifies the piece accordingly. Mathematically this means that one needs to determine the influence of the second derivative at each point.

We consider therefore a *second-order differential equation* $y'' = f(x, y, y')$ to be solved for an interval (a, b) with some boundary conditions and modify the above approach as needed. This equation may be considered as a system of two first-order equations $y_1' = y_2, y_2' = f(x, y_1, y_2)$, or in vector notation

$$\mathbf{y} = (y_1, y_2), \mathbf{f}(x, \mathbf{y}) = (y_2, f(x, y_1, y_2)), \mathbf{y}' = \mathbf{f}(x, \mathbf{y}).$$

The recursive Euler approximation used above now concerns a vector function $\mathbf{k}_t(x_n)$ and has the form

$$\mathbf{k}_{t+1}(x_{n+1}) = \mathbf{k}_t(x_n) + d_n \cdot \mathbf{f}(x_n, \mathbf{k}_t(x_n)),$$

taking the same discretization $x_0 = a, x_1, \ldots, x_N = b$ of (a, b) and $d_n = x_{n+1} - x_n$ as above.

The vector functions $\mathbf{k}_t(x_n)$ have two component functions $\mathbf{k}_t(x_n) = (g_t(x_n), h_t(x_n))$ whose individual recursions are

$$g_{t+1}(x_{n+1}) = g_t(x_n) + d_n \cdot h_t(x_n),$$

$$h_{t+1}(x_{n+1}) = h_t(x_n) + d_n \cdot f(x_n, g_t(x_n), h_t(x_n)).$$

To start the recursion we need to know $g_0(x_n)$ and $h_0(x_n)$ for $n = 0, 1, \ldots, N$ as well as $g_t(x_0)$ and $h_t(x_0)$ for all t. The latter are simply the initial conditions for the solution of the original equation; g_0 is assumed given as the original shape of the workpiece as before, and $h_0(x_n)$ can be computed recursively using g_0 and f:

$$h_0(x_{n+1}) = h_0(x_n) + d_n \cdot f(x_n, g_0(x_n), h_0(x_n)).$$

3. Bürgi's refinement

There are two refinements which Bürgi impressively invented for his approximations. One is the use of the now well-known mid-point modification of the Euler algorithm. The second is to use his experience that one would do well to start working at both ends of the piece, which means to use boundary conditions instead of initial conditions: $g_t(x_0)$ as before, but now $h_t(x_N)$. The above recursions are easily adapted to these refinement; we discuss them in the following section as they apply to Bürgi's computation of the sine function.

Bürgi of course didn't know about differential equations and Euler was not born during his lifetime. But he had a lot of experience with the trigonometric functions. So, if one explained to him the notion of the derivative, he would at once have noticed its relation to differences, he would also not be astonished to hear that $\sin'(x)$ is $\cos(x)$ and $\cos'(x) = -\sin(x)$. The differential equation underlying Bürgi's tabulation is therefore $y'' = -y$, the interval is $(0, \pi/2)$ and the boundary conditions are $y(0) = \sin(0) = 0$ and $y'(\pi/2) = \cos(\pi/2) = 1$.

x	sin x	g 3	h 2	g 2	h 1	g 1	h 0	g 0	x
0	0	0		0		0		0	0
			1.0281334		1.0048331		0.9817477		
30°	*0.5*	0.5383294		0.5261294		0.5140419		0.5	30°
		0.5		*0.5*		*0.5035636*			
			0.7526527		0.7356814		0.7199483		
60°	*0.8660254*	0.9324174		0.9113313		0.891006		0.875	60°
		0.8660287		*0.8660715*		*0.8728183*			
			0.2754807		0.2691517		0.2617994		
	1								
90°		1.0766588		1.0522588		1.0280838		1	90°
		1		*1*		*1*			

Figure 2: Example of an artificium table

The mid-point modification of Euler's algorithm asks for points $\bar{x_n}$ in the interval (x_n, x_{n+1}) at which to take the derivatives for the next step. Assuming $h_t(\bar{x}_n)$ known, the mid-point approximation for g_{t+1} would be $g_{t+1}(x_{n+1}) = g_t(x_n)+d_n \cdot h_t(\bar{x}_{n+1})$. In fact, convergence can be improved by using already computed values of g_{t+1}, and Bürgi takes

$$g_{t+1}(x_{n+1}) = g_{t+1}(x_n) + d_n \cdot h_t(\bar{x}_{n+1}).$$

The values of $h_t(\bar{x}_n)$ are obtained as follows for each t:
Since $h_t(x_N)$ is known as the boundary value $\cos(\pi/2) = 1$, we may take $\bar{x}_N = x_N - d/2$ and set

$$h_t(\bar{x}_N) = h_t(x_N) - d/2 \cdot f(x_N, g_t(x_N), h_t(x_N)) = h_t(x_N) + d/2 \cdot g_t(x_N).,$$

using $f(x, y, z) = -y$ as in the differential equation.
Continuing with $h_t(\bar{x}_{n-1}) = h_t(\bar{x}_n) + d \cdot g_t(x_{n-1})$ we obtain all the values necessary for the recursion g_{t+1}.

4. BÜRGI'S ARTIFICE

As the example above shows, the computed values of $g_t(x_N)$ are in general different from the desired value $\sin(\pi/2) = 1$. Here is the place to use an artifice: The artisan Bürgi "polishes" the result proportionally and proposes the normalised values

$$g_t^p(x_n) = \frac{g_t(x_n)}{g_t(x_N)}, t = 0, 1, \ldots, k$$

These values are added in the above example for $t = 0, 1, \ldots, t_k = 3$. and show an astonishing convergence to the true values. Actually, Bürgi "polishes" only his last computed values $g_{t_k}^p$ for entering into his sine table.

Caveat: Of course, this elegant uniform correction does not generalise to arbitrary differential equations,as simple examples show. Bürgi, in the case of the Sine function, is fortunate or inspired, and his normalization is correct, as shown in [2].

Following a tradition since Ptolemy, Bürgi computes his tables with natural numbers. He presents the progression of approximations as columns from right to left as in the above example. He chooses N and sets $X_N = N$ and the interval $(0, N)$. The interval points are now called X_n and $d = 1$. The functions g_t, h_t are correspondingly renamed G_t, H_t and the intermediate points $\bar{X}_n$. Let $G_0(X_n)$ denote the values in the initial column and adapt the values of $G_0(X_n)$ loosely to $N \cdot \sin(\frac{n\pi}{2N})$.

The recursion which creates the next columns now takes the form

$$H_t(\bar{X}_N) = H_t(X_N) + G_t(X_N)/2, H_t(\bar{X}_{n-1}) = H_t(\bar{X}_n) + G_t(X_{n-1}),$$

$$G_t(X_0) = G_0(X_0), G_t(X_{n+1}) = G_{t+1}(X_n) + H_t(\bar{X}_n).$$

These modifications of course result in a telescoping increase of the following columns. By computing with natural numbers for values and interval size, one also obviates the accumulation of rounding errors. The corresponding computation is shown for our example in [2].

To recover the intended approximation $\sin_t$ of the sine function, we need again to normalise to

$$\sin_t(\frac{n\pi}{2N}) = \frac{G_t(X_n)}{G_t(X_N)}$$

which Bürgi, as mentioned above, performs only for his last column to get the desired values for his table.

A discussion and proof of the resulting convergence is in [2].

5. CODA

In Bürgi's time, the development of mechanical tools for computations lay dormant since antiquity. While he would have been the right person to create a mechanical calculator, there is no record of this having happened. Instead, his artifice is eminently practical for hand computation. It is hopeless as an algorithm for digital computers.

One could consider the generalized algorithm as an example of massively parallel computations : One would simply provide N processors, assign each to one of the points x_n, connect them according to the recursion equations, and compute the various functions locally. Such processors usually have the required multiple input- and output-channels. *Neural nets* are a special case: Individual neurons are very simple processors with limited functionality. They typically have multiple input channels (synapses) but only one output channel (the axon). It is an interesting challenge to overcome these limitations, e.g. by deeper networks of neurons and to apply it to modelling learning and operating regimes of neural nets. But this is another chapter.

REFERENCES

.[1] Staudacher, F.: *Jost Bürgi, Kepler und der Kaiser*, NZZ Libro, Zurich 2016

.[2] Waldvogel, J.: *Jost Bürgi's Artificium of 1586 in modern view, an ingenious algorithm for calculating tables of the sine function*. Elem. Math., to appear.