Polynomial And Recurrence Equations

Martyn Braithwaite 21 September 2008

Ok we all know the basics of differential calculus in relations to the rate of change of a functions value, that:

$$\frac{d(c)}{dx} = 0$$

$$\frac{d(c*x)}{dx} = c$$

$$\frac{d}{dx}\left(c*x^n\right) = c*n*x^{n-1}$$

That the function, $f(x) = c*x$ if it has domain $\forall \boldsymbol{Z}$, it's codomain is $\forall x : x\epsilon\{c*x : x\epsilon\boldsymbol{Z}\}$ that it's difference function (we will denote here by Δ), $\Delta f(x) = c$. Similarly, that $f(x) = c*x^n$ maps onto the codomain: $\forall x : x\epsilon\{c*x^n : x\epsilon\boldsymbol{Z}\}$ from the domain $\forall\boldsymbol{Z}$ and has difference function $\Delta f(x) = g(x) + c*n*x^{n-1}$. And more generally that $\Delta f(x_n) = y : y\epsilon\{\forall y : y = f(x_{n+1}) - f(x_n)\}$ where the domain of $x\epsilon X$ is $X \subset \boldsymbol{Z}$.

We are also aware of recurrence relations, functions such as: $R_x = R_{x-1} + R_{x-2}$ for $x_0 = 1$ and $x_{i<0} = 0$ generate sequences such as the Fibonacci series. And the difference function: $\Delta R_x = R_{x-1}$. What I propose is a method of measuring the rate of change of ratios after the fashion of the derivative, using the distinctive symbol: Ω.

We define it as follows on the domain $X = \boldsymbol{Z}$:

$$\Omega R_x = \forall a : a\epsilon\left\{a = \left\{\frac{R_x}{R_{x-1}}\right\}\right\}$$

Furthermore, we observe that: $\Omega R_x < 0$ the sequence will switch between negative and positive and is not monotonic, if $\forall\Omega R_x > 0$ then the sequence probably is monotonic. Also that:

$$|\Omega R_x| > 1 \Rightarrow \lim_{x\to\infty} R_x \to \infty$$

$$|\Omega R_x| < 1 \Rightarrow \lim_{x\to\infty} R_x \to 0$$

and $\lim_{x\to\infty}\Omega R_x \to \infty$ suggests the presence of an exponent or product of recurrence terms. And finally the identities:

$$R_x = R_{x-1} \Rightarrow \Omega R_x = 1$$

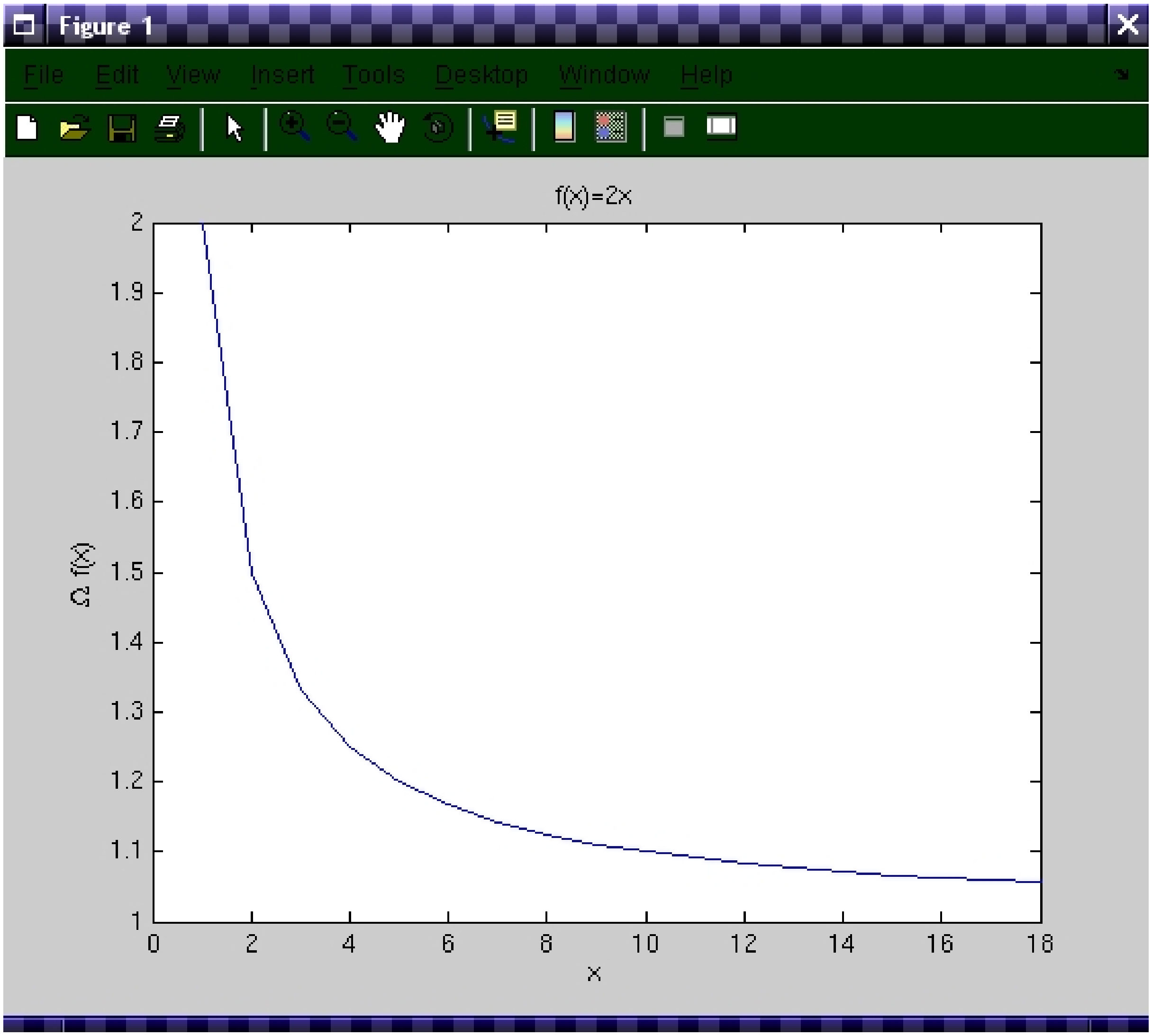
$$R_x = c*R_{x-1} \Rightarrow \Omega R_x = c$$

$$R_x = (R_{x-1})^2 \Rightarrow \Omega R_x = (R_{x-1})^2$$

$$R_x = (R_{x-1})^{n>2} \Rightarrow \Omega R_x > R_x$$

When you have the sequence $\lim_{x\to\infty}\Omega f(x) \to 1$ then $O(f(x))$ (where O represents the highest order) is constant and strictly finite so is a polynomial. These tests and more could be implimented inside ENiX to assist recognition of mathematically governed sequences.

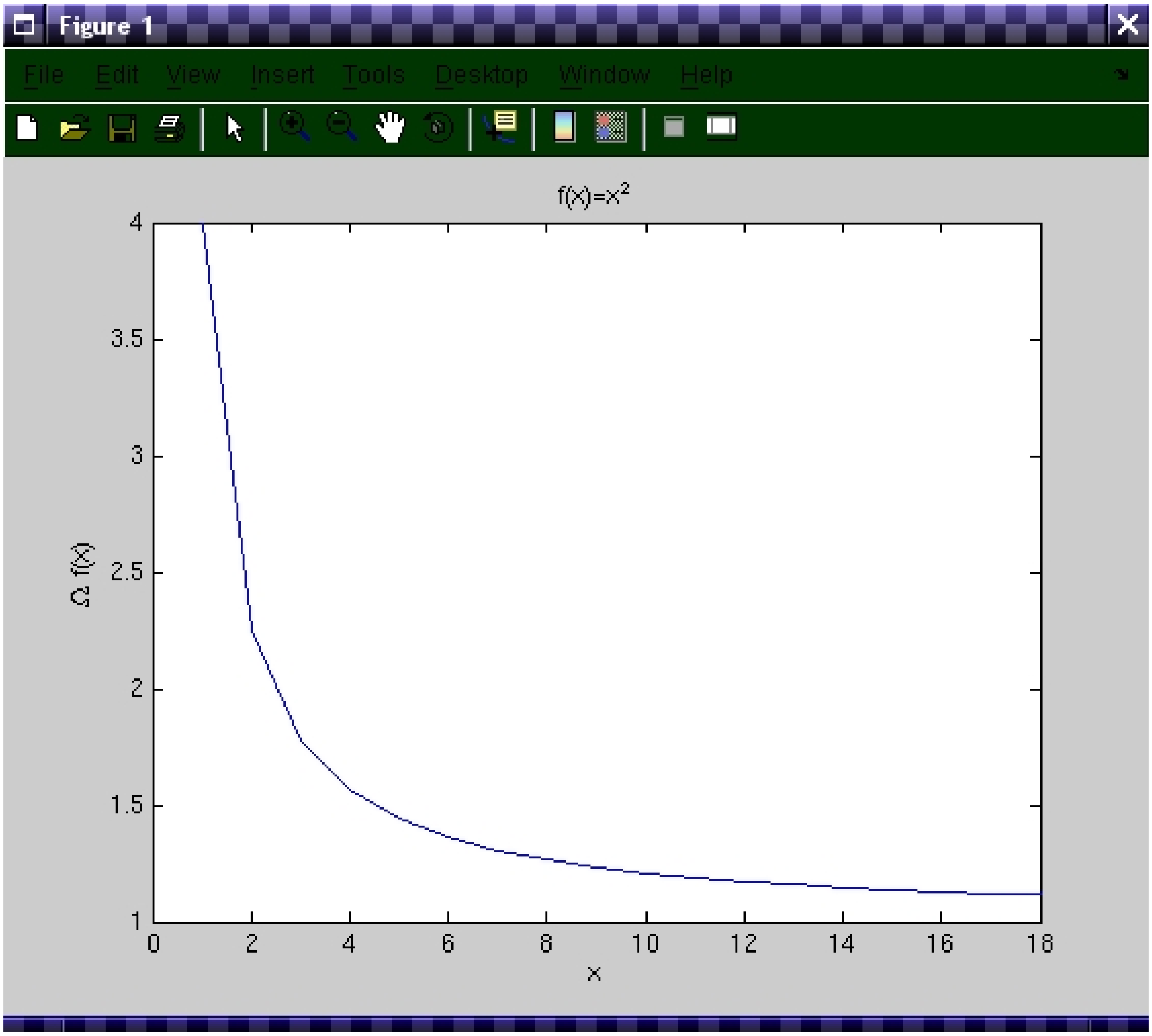
Graphs From Matlab:



Graph 1 (Matlab).

Graph 1 shows the Ω function for an order 1 polynomial. Note that $\Omega f(x)$ seems to be converging towards 1.

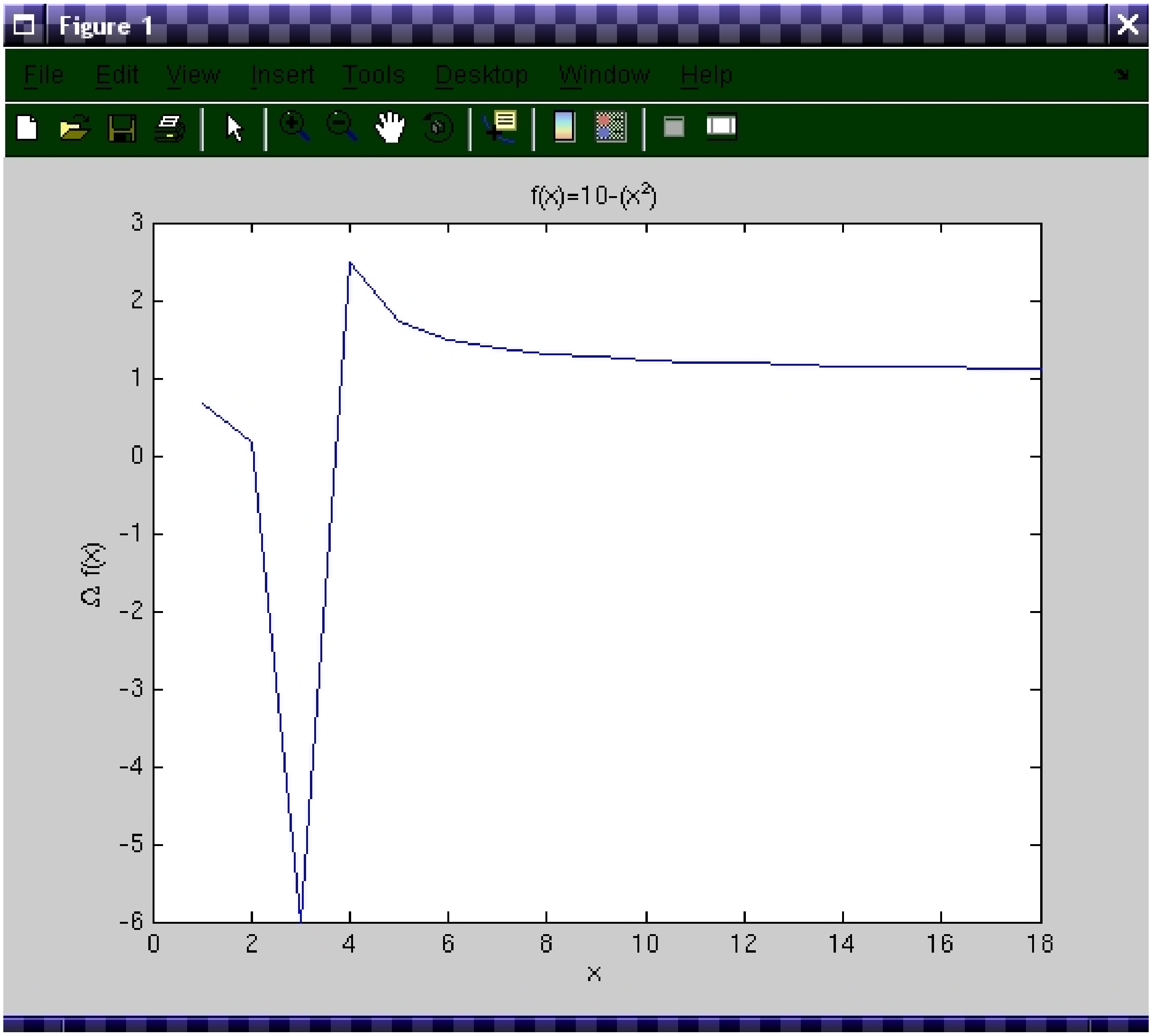
Graphs From Matlab:



Graph 2 (Matlab).

Graph 2 shows the Ω function for an order 2 polynomial. Note that, again $\Omega f(x)$ seems to be converging towards 1, although the convergence seems to be a lot slower than for order 1 polynomials.

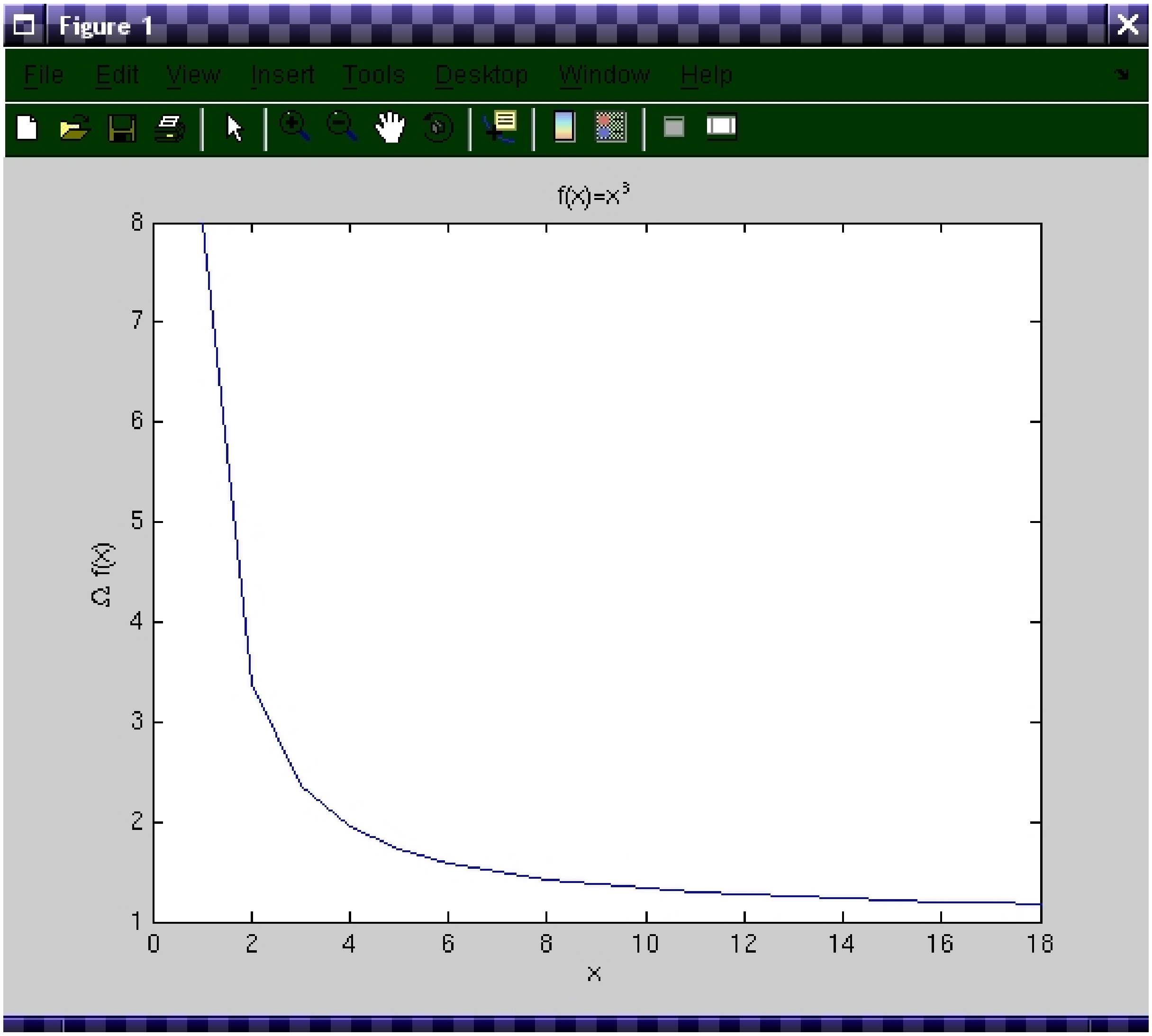
Graphs From Matlab:



Graph 3 (Matlab).

Graph 3 shows the Ω function for an order 2 polynomial. $\Omega f(x)$ is still converging towards 1, despite the initial unevenness.

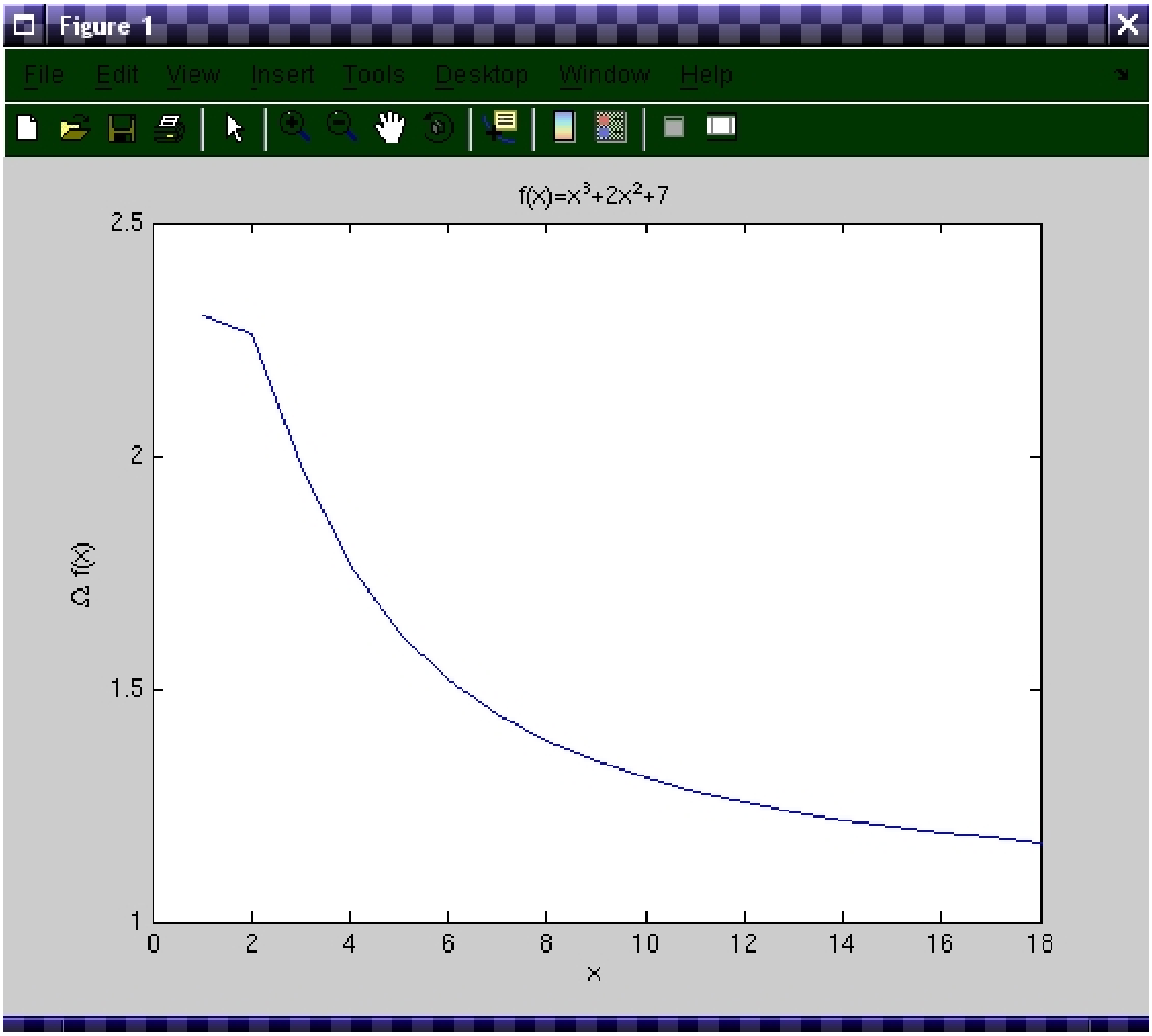
Graphs From Matlab:



Graph 4 (Matlab).

Once again a polynomial with an $\Omega f(x) \to 1$ as $x \to \infty$. Graph 4 shows this order 3 function converging more slowly towards 1 than order 1 or order 2 polynomials.

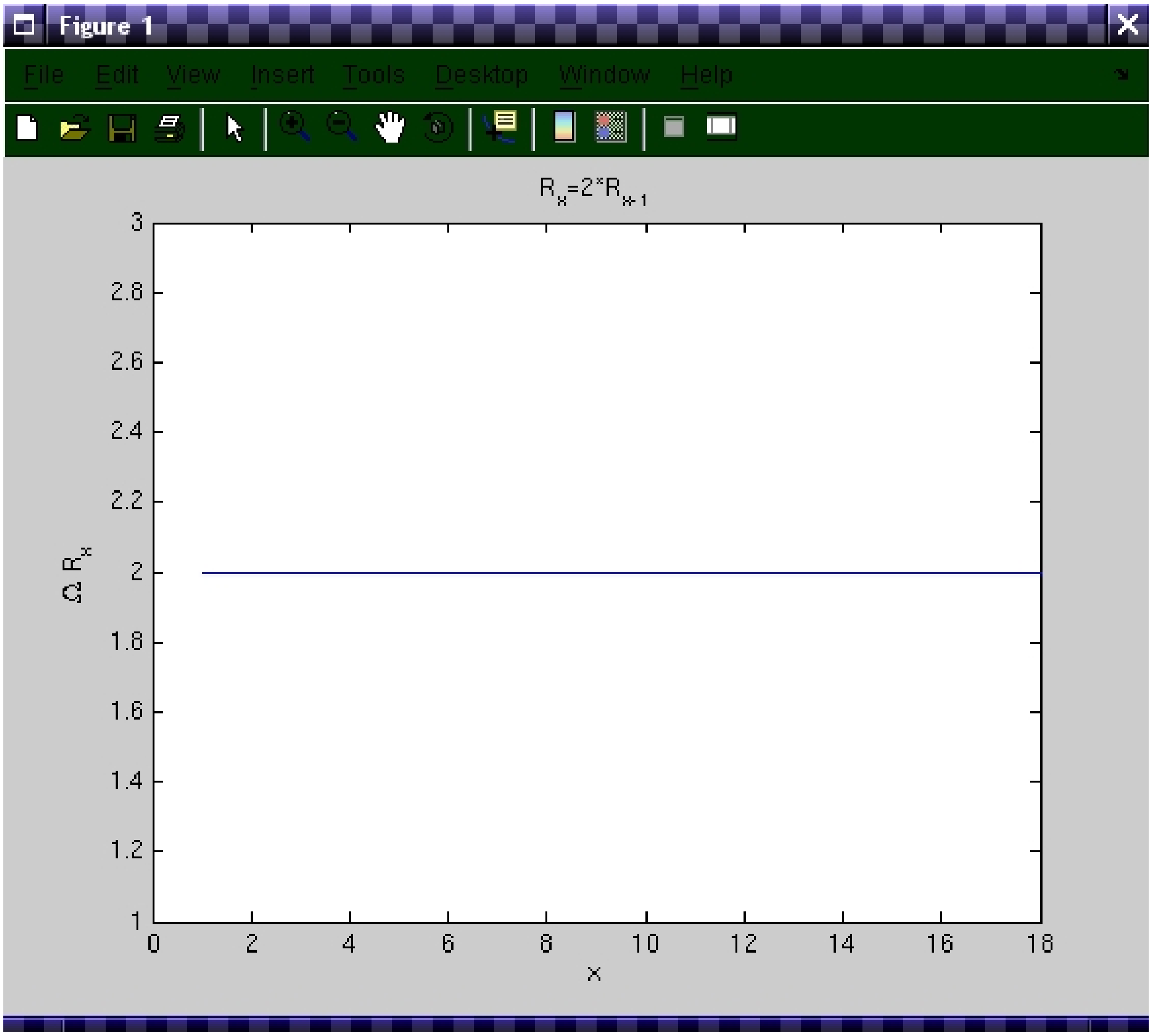
Graphs From Matlab:



Graph 5 (Matlab).

This graph shows the convergence to 1 of $\Omega f(x)$. See how there is a slight deformation from the less significant terms in the polynomial in Graph 5 compared to Graph 4.

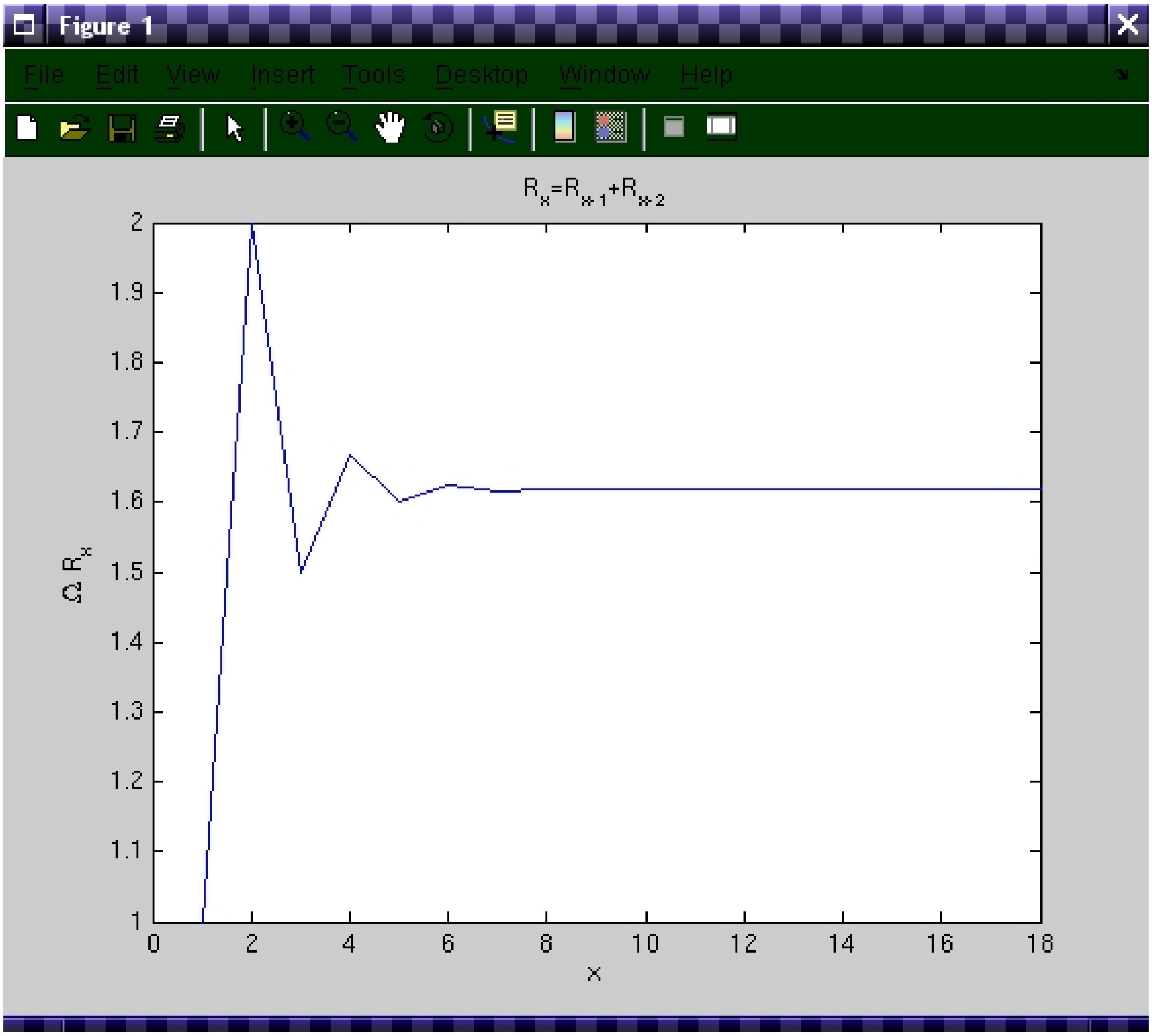
Graphs From Matlab:



Graph 6 (Matlab).

Graph 6 shows a graph of ΩR_x for $R_x = 2 * R_{x-1}$. NB how the graph doesn't converge around 1. This function, $R_x = 2 * R_{x-1}$ is equivalent to the function: $f(x) = R_0 * 2^x$ which is the reason why it is a straight line.

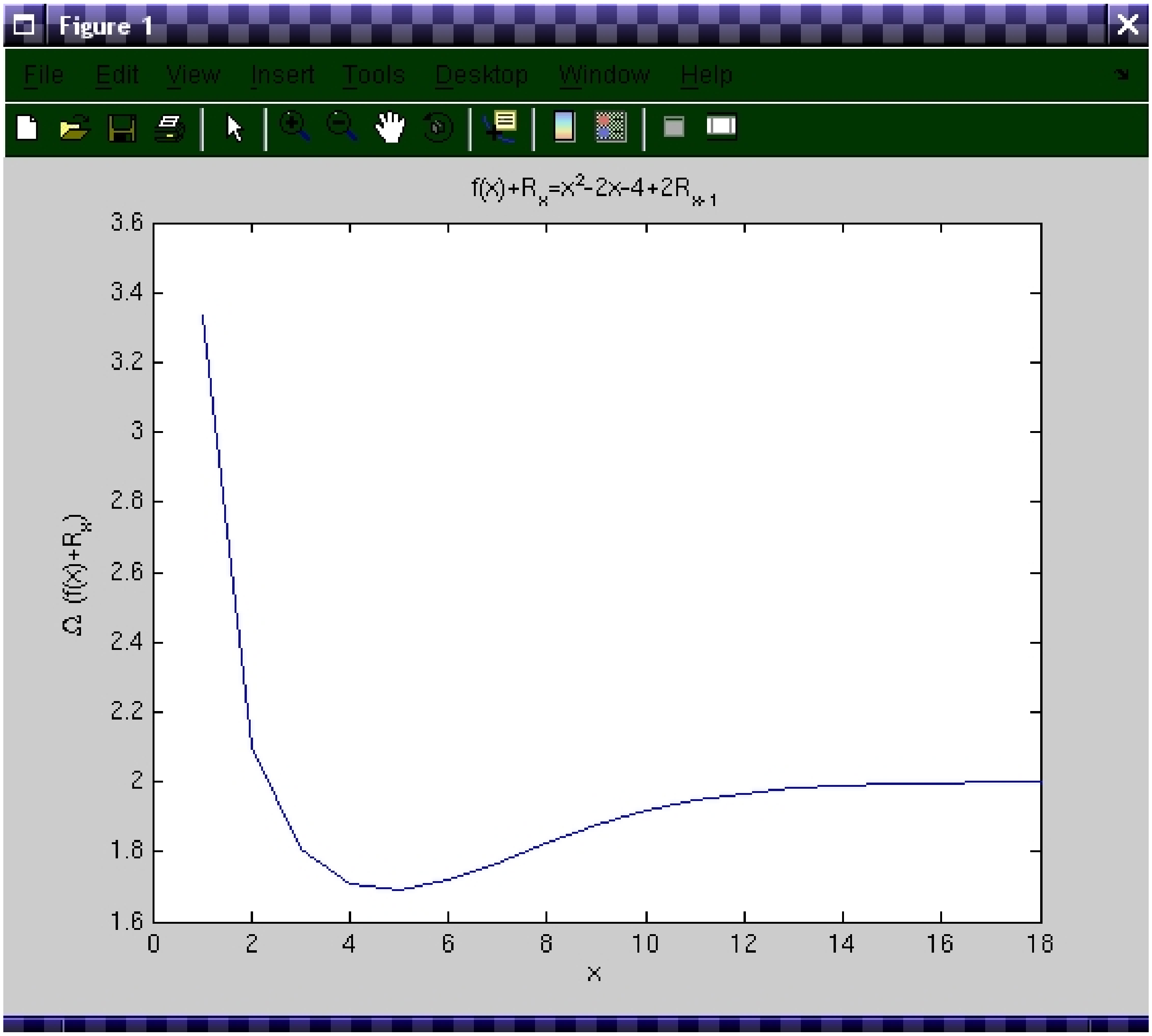
Graphs From Matlab:



Graph 7 (Matlab).

Graph 7 depicts the Ω function of the sequence $R_x = R_{x-1} + R_{x-2}$ for $R_0 = R_1 = 1$ which is the Fibonacci series. Although ΩR_x does converge rapidly to a value, this value is not 1.

Graphs From Matlab:



Graph 8 (Matlab).

This graph, Graph 8, shows a combination of recurrence relation and polynomial generating the sequence. We can see some heavy deformation of the graph of ΩR_x for $R_x = 2 * R_{x-1}$ compared to Graph 6. This initial period of deformation is caused by the polynomial terms. However in the scale of things, the polynomial terms quickly get overwelmed by the recurrence relation and the sequence converges rapidly to 2.

Algorithm:

Ok, so if we have a sequence of numbers, $\{s_0, s_1, ..., s_n\}$ which we believe is generated by the function: $R_x = f(x) + M(R_0, ..., R_{x-1})$, where $f(x)$ is a polynomial and $M(R_0, ..., R_{x-1})$ is a combination of recurrence terms, then if:

$$\lim_{x \to \infty} \Omega R_x \neq 1$$

then:

$$\lim_{x \to \infty} \Omega R_x = \lim_{x \to \infty} \Omega M(R_0, ..., R_{x-1})$$

Now, if $R_x = f(x) + M(R_0, ..., R_{x-1})$, then $R_0 = f(0)$, $R_1 = f(1) + M(R_0)$, $R_2 = f(2) + M(R_1, R_0) = f(2) + M((f(1) + M(R_0)), R_0) = f(2) + M((f(1) + M(f(0)), R_0)$ etc.

If we have a formula for $M(R_0, ..., R_{x-1})$, can we work out what the sequence $\{f(0), f(1), ..., f(n)\}$ is, because if we could, then we can use the test,

$$\lim_{x \to \infty} \Omega f(x) = 1$$

to establish whether or not our M function has been worked out correctly. If it does equal 1, then M is correctly derived. If it does not, then it isn't and M needs modification. Once the above test is satisfied by evaluating the correct M, it is removed from the sequence, $\{R_0, R_1, ..., R_n\}$ to leave: $\{f(0), f(1), ..., f(n)\}$, which our polynomial finder can find without too much difficulty. Then M and $f(x)$ are summed to give us R_x, which is the solution to the sequence, $\{R_0, ..., R_n\}$.

Finding $f(n)$ From R_n And $M(R_0, ..., R_{n-1})$:

$$f(n) = R_n - M(R_0, ..., R_{n-1})$$

So for any sequence, generated by R_x we can construct $M(R_0, ..., R_{n-1})$, use this M function to work out the sequence minus the recurrence component. If the remaining sequence Γ, satisfies $\Omega\Gamma = 1$ then $M(R_0, ..., R_{n-1})$ *might be* correct. Otherwise $M(R_0, ..., R_{n-1})$ is not correct. When M is correct, we can set about finding the polynomial to generate Γ easily.