M365C (Rusin) HW5 comments

1. The book has proofs that, under these hypotheses, $\{1/b_n\}$ converges (to $1/\lim(b_n)$) and then that the sequence $c_n = (a_n) \cdot (1/b_n)$ converges as well (to $(\lim a_n) \cdot (1/\lim b_n)$). But it is not a bad idea for you to prove this directly: if $a_n \to A$ and $b_n \to B$ then given any positive ϵ we can find an $N = N(\epsilon)$ past which every $|a_n - A| < \epsilon|B|/4$ and $|b_n - B| < \epsilon|B|^2/2|A|$ and furthermore (taking N larger if necessary) we may assume $|b_n - B| < |B|/2$. (Note that this last inequality forces $|b_n| > |B|/2$.)

Then we can estimate how far a_n/b_n is from its eventual limit, A/B:

$$\begin{aligned}
\left|\frac{a_n}{b_n} - \frac{A}{B}\right| &= \frac{|B\,a_n - A\,b_n|}{|B||b_n|} \\
&= \frac{|B\,a_n - B\,A + B\,A - A\,b_n|}{|B||b_n|} \\
&\le \frac{|B\,a_n - B\,A| + |B\,A - A\,b_n|}{|B|^2/2} \\
&= \frac{|a_n - A|}{|B|/2} + |A|\frac{|B - b_n|}{|B|^2/2} \\
&< \frac{\epsilon}{2} + \frac{\epsilon}{2} = \epsilon
\end{aligned}$$

2. The distance $d(a_n, a_m)$ works out to $|1/(n+1) - 1/(m+1)|$. (It helps to note that $x^n \ge x^m$ for all $x \in [0,1]$ if $n < m$: the graphs don't cross.) So given $\epsilon > 0$, for any n and m which are both larger than $1/\epsilon$, we have $d(a_n, a_m) < \max(1/n, 1/m) < \epsilon$.

You might think the sequence is "trying" to converge to a discontinuous function which is zero everywhere except at $x = 1$. But convergence in this metric space is not the same as what is called "pointwise convergence". Rather, the a_n are converging to the function a_* which is zero for every x. In fact, just as above, we compute $d(a_n, a_*) = 1/(n+1)$, which is easily forced to be less than any pre-assigned ϵ.

Of course, one can prove this is a Cauchy sequence by *first* showing that it converges!

3. Using the definition of a Cauchy sequence, with $\epsilon = 1$, shows that a sequence is a Cauchy sequence if and only if it is eventually constant (i.e. all but a finite number of terms are equal to each other). Such sequences are obviously also convergent.

4. I'm sorry, I tend to forget that Continued Fractions is not part of the current high school curriculum. What you need to know is this: by the Continued Fractions algorithm (or otherwise), you know that there exist infinitely many rational numbers p_n/q_n which are "really good" approximations to π: for example we can insist that $|p_n/q_n - \pi| < 1/q_n^2$. Then for each such n we have $|p_n - q_n\pi| < 1/q_n$. Then by the periodicity of the sine function,

$$|\sin(p_n)| = |\sin(q\pi + (p_n - q_n\pi))| = |\sin(p_n - q_n\pi)| < |p_n - q_n\pi| < 1/q_n$$

since $|\sin(x)| < |x|$ for all $x \in \mathbf{R}$. So the integers p_n form a subsequence whose sines approach zero.

5. If the x_n do converge, say to some number L, then taking the limit of both sides of the equation $x_{n+1} = x_n(2 - Ax_n)$ shows that $L = L(2 - AL)$. That's a quadratic equation, and its roots are 0 and $1/A$. So either the sequence converges to $1/A$, or it converges to 0, or it does not converge at all. Some examples suggest that the first possibility is the right one. We verify this by using the definition of convergence. But note that that definition emphasizes the distances to the limit rather than the individual numbers x_n!

So let us define a new sequence $e_n = x_n - 1/A$, so that $x_n = 1/A + e_n$. We will show the e_n decrease to zero by using the given recurrence relation: it implies that $e_{n+1} = x_n(2-Ax_n)-1/A = (1/A+e_n)(1-Ae_n)-1/A = -Ae_n^2$, or equivalently $Ae_{n+1} = -(Ae_n)^2$. By induction we conclude $Ae_n = -(Ae_0)^{(2^n)}$ for every $n > 0$, so that the sequence e_n converges (very rapidly!) to zero as long as $|Ae_0| < 1$, i.e. as long as $0 < A < 2$. (One can apply this process for any nonzero A but as you can see one must start with a comparably small initial "error" $e_0 = x_0 - 1/A$.)

Note: This sequence arises from applying Newton's Method to the function $f(x) = 1/x - A$. A similar analysis applies in every application of Newton's method, with similarly rapid convergence: it's called "quadratic convergence", and in practice it gives twice as many decimal digits of accuracy with each repetition of the recurrence. For example, one may use it on $f(x) = 1/x^2 - A$ to compute $1/\sqrt{A}$ using the recurrence

$$x_{n+1} = \frac{1}{2}x_n(3 - Ax_n^2) = x_n(1 + \frac{(1 - Ax_n^2)}{2})$$

These two examples are especially nice because they allow the computation of inverses and square roots using only addition, subtraction, and multiplication, not division (except by 2 — a bit shift, in computer lingo).

Mathematically that's interesting because there are metric spaces other than $\mathbf{R}$ where we might want to perform computations, specifically the ring of all real power series in one variable X: we give this set a metric like the 2-adic metric; where $d(f, g) = 2^{-r}$ and r is the lowest power of X that appears in $f - g$. I invite you, for example, to compute $1/\sqrt{A}$ in the case where $A = 1 - 4X$; it's a power series which we can approximate with the sequence defined recursively in the previous paragraph:

$$f_0 = 1$$
$$f_1 = 1 + 2X$$
$$f_2 = 1 + 2X + 6X^2 + 20X^3 + 16X^4$$
$$f_3 = 1 + 2X + 6X^2 + 20X^3 + 70X^4 + 252x^5 + 924X^6 + 3432X^7 + 8496X^8 + \ldots$$

and so on. Again there is "quadratic convergence": each successive iteration doubles the number of coefficients that occur in the actual Taylor series of $f(X) = 1/\sqrt{1 - 4X}$. In particular, each f_n has all the coefficients correct up to and including that of X^{2^n-1}. Call that last correct coefficient c_n.

This particular Taylor series is especially interesting because the coefficients happen to be the numbers in the middle column of Pascal's triangle. Those numbers are easily seen to have lots of prime factors. In particular, that last "correct" coefficient in f_n is divisible by all the primes between 2^n and 2^{n+1}, that is, all the n-bit primes (when written in binary)!

So here is a very easy way to factor a large number N (let's say, a number written out with a few hundred bits, in binary). Such a number only has prime factors which are at most a couple hundred bits long, right? So we need only repeat the following steps for $n = 1, 2, \ldots$ a couple hundred times:

(1) Compute f_n from the above recursion.
(2) Pick out the coefficient c_n.
(3) Compute the greatest common divisor $\gcd(N, c_n)$.

(The last step can be accomplished very quickly using the Euclidean Algorithm, and in fact all the computations may be done modulo N if you wish.)

When, for example, N is a composite number used for encryption in the common RSA algorithm, it is typically the product of two prime numbers, almost surely one of them appearing in step 3 for one value of n and the other appearing for a later value of n. There are refinements of the algorithm to handle exceptional cases, so really there appears to be no bottleneck! Factoring large numbers is actually easy!

Or is it?