Solutions to homework 2

Warren D. Smith
wds@research.NJ.NEC.COM

date = October 23, 1998

1 Steering a car

The condition is that all 4 wheels must be tangent to concentric circles. Since the 2 back wheels are fixed, the circle center must lie on the line through the back axle. Hence

$$L\cot\theta_2 - L\cot\theta_1 = W. \tag{1}$$

Note, turning the front wheels so they are parallel, i.e. so $\theta_1 = \theta_2$, which is what the steering mechanisms on many cars do, is wrong and is not the same as this. However, it is the same thing asymptotically in the limit where the $|\theta|$'s are small.

A symmetric 3-link planar mechanism (two of the links being attached to the wheels and riveted to the car body, the third floating link joins the other two) ought to work to provide a better approximate solution than $\theta_1 = \theta_2$ over some nonzero interval of allowed steering angles. I have not computed the optimal dimensions of the 3 links for this purpose (they depend on the angle range, degenerating to the parallelogram mechanism in the limit when this range tends to zero width), but I think some cars do use this idea. This idea is certainly just as cheap as building it wrong, so there is no excuse not to (other than the desire to waste fuel and make you replace your tires more often).

I do not know what the minimum number of links is in a planar mechanism for enforcing the *exact* steering condition (EQ 2). One solution (which is rather flawed since one can only turn left!) found by both Jade Vinson & me, is to attach 3 Hart 5-bar perpendicular-line drawing devices to each front wheel and one of the back wheels, and attach them all at their "pens" (which will be the common circle center). This takes 15 bars in addition to the wheels themselves. Unfortunately this center moves out to ∞ along the rear wheel axis line when you try to steer the car straight, whereas the mechanism must be made of finite size parts.

The problem here is analogous to the problem faced by somebody trying to draw a circular arc of very large radius, with a very small compass! One of the bandied applications of the "inversor" mechanism was for solving that problem: by performing an inversive transformation, a circle of very large radius may be mapped to one of very small radius. Thus a small compass attached to a small inversor (in total, a 6-bar mechanism) will enable drawing a circular arc of very large radius.

Anyhow, my steering solution is to use

$$\tan\theta_2 = \frac{L\tan\theta_1}{L + W\tan\theta_1} \tag{2}$$

in which all the quantities are finite. The $\tan\theta$'s may be extracted using a 2-bar mechanism on each wheel (one of the bars being the wheel line itself, the other sliding telescopically) if we permit sliders, or the sliders may be emulated with a 5-link line drawer attached to the car body, plus another attached to the wheel, joined at their "pens." Finally, an inversor may be used to enforce the collinear Moebius transformation, which takes an additional 4 links. The total is 24 links, or only 4 links if you permit 4 sliding widgets. Quite likely there is a simpler solution, though.

2 3-bar linkage rotation criteria

Well, a can rotate 360° iff $b+c \ge 2+a$ and $|2-a| \ge |c-b|$. All you need is the triangle inequality (and its converse form) applied 3 different ways. Similarly c can rotate 360° iff $b+a \ge 2+c$ and $|2-c| \ge |a-b|$.

3 Chebyshev approximate line drawing devices

WLOG make the origin lie at the pen point when the pen is in the (pictured) central position, and let the y-axis be the axis of symmetry. The curve traced by the pen is an algebraic curve of 6th degree of the form $f(x^2, y) = 0$. Such a curve has (or anyway we want it to have) a y-min at $x = 0$, y-maxes at $x = \pm p$, y-mins at $x = \pm q$, and finally, at the extremes $x = \pm r$ of the interval of approximation, there will automatically be a y-max in that interval. (Or, the words "min" and "max" could be interchanged, or the sign of the y axis changed, which is the same thing, and this won't matter to us.) Let z be the value of the y-coordinate at the y-max when $x = \pm p$. So we want

$$f(0,0) = f(q^2,0) = f(p^2,z) = f(r^2,z) = 0 \tag{3}$$

and

$$f_1(q^2,0) = f_1(p^2,z) = 0 \tag{4}$$

where the subscript 1 denotes partial derivative with respect to the first argument. The formula for $f(x^2, y)$ is deducible and may be expressed in terms of a, b, h, w. Now the equations $f(p^2,z) = f_1(p^2,z) = f(r^2,z) = 0$ are always solvable since 3 equations in 3 unknowns p, r, z. That leaves us with two equations $f(q^2,0) = 0$, $f_1(q^2,0) = 0$, in 1 unknowns q. These equations will generally not have a solution, because 2 equations generally overdetermine 1 unknown! However, if a, b, h, w were picked correctly by Chebyshev (or us) to make an equiripple line drawer, they will have a solution!

The condition for simultaneous solvability of $f(q^2,0)=0$ and $f_1(q^2,0)=0$ is that

$$\mathrm{Discriminant}_{(x^2)} f(x^2,0)=0. \tag{5}$$

(The discriminant of a univariate polynomial is zero iff it has a multiple root.) That eliminates q and constitutes a pure relationship among a,b,h,w, which is going to be precisely the condition for equiripple error. Slick, eh?

With this strategy prepared, we merely have to do the algebra, or more precisely, have `MAPLE` do it for us. The coordinates of the anchorpoints are

$$P_x=\pm a/2, \quad P_y=h+\sqrt{4b^2-(a+w)^2}/2. \tag{6}$$

Let θ be the angular departure of the triangle's orientation from its orientation in the central position. Let $s=\sin\theta$, $c=\cos\theta$. Then $c^2+s^2=1$. Then

$$(y-hc-ws/2-P_y)^2+(x-hs+wc/2-P_x)^2-b^2=0 \tag{7}$$

$$(y-hc+ws/2-P_y)^2+(x+hs-wc/2+P_x)^2-b^2=0 \tag{8}$$

Now we tell `MAPLE`

```
e1 := resultant(
        (y - h*c - w*s/2 - Py )^2
      + (x - h*s + w*c/2 - Px )^2
      - b^2,
     c^2+s^2-1, c);
e2 := resultant(
        (y - h*c + w*s/2 - Py )^2
      + (x + h*s - w*c/2 + Px )^2
      - b^2,
     c^2+s^2-1, c);
e3 := resultant(e1, e2, s);
e4 := lprint(factor(e3));
```

which will derive the equation for $f(x^2,y)$ (as one of the factors in `e4`). Also, you could just directly extract an expression for the 3-bar curve from my paper, which would be a lot easier for poor `MAPLE`, but I just did this so you could see an example of resultants in action. Then, we let this factor be `f(xsquared,y)` and tell `MAPLE` to compute

```
simplify(discrim(f, xsquared));
```

which (set to 0) will give us our equiripple criterion.

I SHOULD WRITE THE EXPLICIT CRITERION???

4 Life is universal

"Life" is defined on an infinite checkerboard as follows. There are live and dead squares. A dead square is "born" if it has exactly 3 live neighbors (from among the 8). A live cell survives if it has exactly 2 or 3 live neighbors.

Conway invented Life in an effort to make a very simple rule set supporting complex (Turing universal) behavior[1]

It is impossible to compete in Life unless you know something about the huge number of Life Forms that have been constructed since Conway originally invented the game.

Here is a list of some references:

1. Berlekamp, Conway, and Guy: Winning Ways (for your Mathematical Plays), Volume 2, 1982. Academic press.

2. BYTE magazine: Sep 75, Oct 75, Dec 75, Jan 76, Dec 78, Jan 79, Apr 79, Oct 80, Jul 81.

3. Complex Systems: Apr 87, Dec 87, Dec 90, Feb 91, Oct 92.

4. Martin Gardner: Wheels, life, and other mathematical amusements, Freeman 1983.

5. Omni: Oct 84.

6. Scientific American: Oct 70, Nov 70, Jan 71, Feb 71, Mar 71, Apr 71, Nov 71, Jan 72, Dec 75, Mar 84, May 85, Feb 87, Aug 88, Aug 89, Sep 89, Jan 90.

7. Claus Emmeche: The Garden in the Machine: the Emerging Science of Artificial Life (Princeton University Press, 1994).

8. William Poundstone: The Recursive Universe, William Morrow and Co. 1985.

9. J.Gravner & D.Griffeath: Cellular automaton growth on Z^2, Adv. Appl. math. 21 (1998) 241-304.

Here is a quick summary of a few of the amazing things that Lifers have created:

1. Still lives. These are configurations which persist forever without change. All nonisomorphic still lives with n live cells for $n\le 21$ have been enumerated by Mark Niemec. Here are some of the simpler ones:

```
OO      OO       OO..   OO
OO     O..O      O...  O..O
        OO       .OOO  O..O
                 ...O   OO
block  beehive  eater   pond
```

   Eaters have a remarkable capacity to "eat" things (gliders, spaceships, blinkers, etc.) located at their "mouth" end (the NW end, as pictured).

2. Oscillators. The simplest is the "blinker" `OOO` which rotates 90° every tick, a period-2 oscillator. It is now known that there are configurations which oscillate with period n for all $n\ge 1$ except for $\{19,23,27,31,37,38,41,43,49,53\}$, which remain open.

[1] Another simple Boolean 2D CA rule set was devised at about the same time by Edwin Banks [Information processing and transmission in cellular automata, MAC TR-81 MIT project MAC (1971)]. It is defined by the 3 pictures below, plus rotational invariance, plus the rule that every cell-state stays the same next time unless one of these 3 pictured change-rules applies:

$$\begin{array}{ccc} & 1 & \\ 0 & 1 & 1 \\ & 0 & \end{array} \to 0 \qquad \begin{array}{ccc} & 0 & \\ 1 & 0 & 1 \\ & 1 & \end{array} \to 1 \qquad \begin{array}{ccc} & 1 & \\ 1 & 0 & 1 \\ & 0 & \end{array} \to 1 \tag{9}$$

Banks showed it is Boolean-universal, i.e. arbitrary boolean circuitry may be encoded with only polynomial slowdown. He showed this by demonstrating how to make "wires" along which signals would propagate at constant speed (1 cell per timestep), an oscillatory "pulse generator," widgets allowing signals to fan out, turn corners, and cross over, and a gate performing $x=y$ AND NOT z.

3. Gliders and spaceships. The two simplest examples are shown.

```
 .O..O     ..O
 O....     O.O
 O...O     .OO
 OOOO.
spaceship  glider
```

These move S and diagonally SE (respectively) with speeds $c/2$ and $c/4$ respectively. Gardner claims Conway showed $c/4$ is fastest possible for a diagonal mover, but I do not see a proof of that. Many other kinds are known including horizontal movers at speeds $c/5$, $c/4$, $c/3$, $2c/5$, and $c/2$ and and diagonal movers at speeds $c/4$ and $c/12$.

4. Glider guns. Many are known. These emit a glider every n generations. The first one was discovered by R.W.Gosper and has $n = 30$; it uses two "queen bee" shuttles that collide every 30 generations in a particular way to create a new glider. Dietrich Leithner found a way to interleave two period-30 guns to make the fastest known glider gun with $n = 15$.

```
http://www.bigdaddyo.com/~paul/patterns/p15.html
```

The following configuration by N.Gotts & P.Callahan of 10 live cells in an 8×35 rectangle will (after about 2000 ticks) evolve into a glider gun emitting in the SW direction. This (10) is the smallest number of live cells known to grow without bound; an exhaustive search allegedly shows that no 8-cell configuration can do so.

```
.OO
OO.
.O.
................................O.
...............................O.O
...............................O..
..............................O...
```

In the other direction, devices have been made to double the period of a glider gun, allowing arbitrarily slow ones. Spaceship guns are also available.

5. Puffer trains. These travel along at some speed, leaving a trail of "exhaust" behind them. One of the simplest examples, with only 11 live cells, is C.L.Corderman's "switch engine puffer train"

```
 .O.O......................
 O.........................
 .O..O.....................
 ...OOO....................
 ..........................
 ........................OO
 ........................O.
```

It moves diagonally NW at speed $c/12$ leaving 8 "blocks" behind every 288 ticks.

6. Small configurations with very complex behavior:

```
..O......    .OO      OOO...O  O.....O.
....O....    OO.      .O..OOO  ..O...O.
.OO..OOO.    .O.      .....O.  ..O..O.O
                               .O.O....
 acorn  r-pentomino  rabbits  bunnies
```

The acorn and r-pentomino respectively simmer down into a union of simple objects (still lives, oscillators, and gliders) after 5209 and 1103 generations.

7. A stable reflector is a still life which can reflect a glider (through 90 or 180 degrees) without suffering any permanent damage. As Conway himself pointed out back in the 1970's, a stable reflector can be used to construct oscillators of all sufficiently large periods. Several stable reflectors are known; the minimum known recovery time (time before it is ready to reflect another glider) is 623. Much simpler reflectors are possible if considerable constraints are placed on the glider arrival times and if you don't require stillness.

8. Totally aperiodic configuration: A finite configuration which grows so that each cell in the universe eventually lives and dies an infinite number of times, but the sequence is aperiodic. Gosper & Wechsler managed to find 2 examples, involving waves of gliders going back and forth across a widening river, while four engines busily lengthen the banks and launch new gliders.

9. "Breeders." These have populations which grow quadratically with time. Dean Hickerson made a 603-cell puffer train moving E at speed $c/2$ whose exhaust turns into NE-shooting glider guns. This fills a triangular region with gliders, whose area grows quadratically with time.

10. A 187-cell "spacefiller" fitting in a 27×27 box has been devised by H.Holzwart, D.Bell, A.Hensel, & T.Coe which fills the entire plane to a density of 1/2 with "stripes."

```
OOOOOOOO
........
OOOOOOOO
........
OOOOOOOO
........
stripes. For the spacefiller see
 http://www.cs.jhu.edu/~callahan/
   patterns/sf/sfiller.html
or the Griffeaths paper.
```

The spacefiller in action looks like a 45° tilted square filled with stripes, and the box expands at a speed of $c/2$. The key was the careful design of "stretchers" at the corners of the square. This creature is optimal in the following senses. First, the "stripes" pattern is the highest possible density (1/2) for any infinite still life (indeed for any pattern such that for each cell, ≤ 3 of its 8 neighbors are live). This was proven by Noam D. Elkies ["The still-Life density problem and its generalizations," in book "Voronoi's Impact on Modern Science"] and seems to be a much more difficult theorem than it looks. Second, I claim the speed $c/2$ of expansion of the box is best possible. The proof is as follows. We claim that if a pattern is contained in a horizontal square of side N, after $N/2$ more steps it will be contained in the $45°$ tilted square whose sides touch its corners. After $N/2$ more steps it will be contained in the horizontal square of side $2N$ whose sides touch its corners. (These lemmas may be established by seeing that a horizontal or diagonal line containing N live cells bordering a sea of death will cause at most $N-2$ cells to be born in the line next to it next tick.) Hence for any pattern whatever surrounded by a sea of death, there must be an enclosing square whose sides are growing at velocity $c/2$.

11. Garden of Eden object: has no predecessor. A counting argument based on the fact that

```
.....      .....
.....      .....
.....  and ..O..
.....      .....
.....      .....
```

have the same successor will show that some configuration with no ancestor must exist inside a box $10^{10} \times 10^{10}$. The simplest example known, with only 143 live cells, fits in a 14×14 box. It is due to Achim Flammenkamp:

```
.O............
...O.O.O.O..O.
O...O.O.O.O...
...O.........O
.O..O.....O.O.
......O......O
.O...O..O.O.O.
O.O.O........O
.......O....O.
O...O.O...O..O
...O.O.O.O....
....O...O..O.O
.O...O...O..O.
..O.O.O.O..O.O
```

12. Fast algorithms: It is possible to simulate Life for N cell toggling operations while using *far* less than N computer operations, in most cases. The idea is to make a "quadtree" (a $2^n \times 2^n$ square has 4 child subsquares, each $2^{n-1} \times 2^{n-1}$) and use a "hash table" to record previously encountered configurations in quadtree squares. These may be updated for a large number of generations with no work (since the answer is just looked up in the table). Most life configurations involve mostly empty or time or space periodic patterns, and thus may be simulated extremely effectively by this algorithm. For example, Gosper's original "puffer train" may be simulated for about 2^n cell togglings in only $O(n)$ actual computer time. [R.W.Gosper: Exploiting Regularities in Large Cellular Spaces, Physica 10D (1984) 75-80]. For small quadtree cells, e.g. 32×32, you can often take advantage of "word wide parallelism" (e.g. bitwie AND on a 32-bit word does 32 boolean operations in parallel) to speed things up by a large constant factor.

Anyhow... to get back to the original problem. Berlekamp, Conway, and Guy in their book sketched a proof that a universal computer may be built out of an initially finite Live configuration. Their proof is roughly as follows. They propose boolean "circuits" based on glider collisions. Signals are carried by streams of gliders. Conway devised AND, OR, and NOT gates, glider stream copying devices, and devices allowing signals to cross over, turn corners, and get pushed sideways and/or delayed by a few pixels. The "eater" will swallow a glider stream (useful for eliminating unwanted signals). Various 2-glider collisions allow the synthesis of a block, a pond, a blinker, an eater, or nothing (the gliders vanish). This was merely the beginning of a vast exploration of "glider collision chemistry" by many investigators. (For example, glider collisions can synthesize a glider gun. Every still life with ≤ 14 live cells can be synthesized by glider collisions. A glider can react with a block to eliminate both.) A "clock" can be made with a glider gun. With these tools, you can build a circuit that implements any desired finite state machine. Another neat trick explained by Conway is a method for creating glider flotillas – these might naively seem to require glider guns which fire through one another, which might seem impossible, but in fact gliders can be maneuvered sideways by pushes from other gliders to enable creation of a very wide range of glider flotillas.

Next, we require an infinite memory. Marvin Minsky showed that just two non-negative integer "counters," which may be (on command) incremented, decremented, or tested for 0, suffice to simulate a universal Turing machine, although three are far more efficient. (See §15.1 of Minsky's "Computation: Finite and Infinite Machines".)

To make a counter, Conway used a "block." Incrementing is pushing the block 1 unit over, decrementing is pulling it. His construction was simplified and made totally explicit by Dean Hickerson, who writes:

> After examining a couple thousand collisions between 2 gliders and a block, I found a 2-glider salvo which can pull a block 1 unit and a 3-glider salvo which can push it 1 unit. I then built a "shotgun" which produces the necessary salvos and a control device which releases them when instructed to do so by external circuitry.... The

device automatically reports the transition from 1 to 0. (If you want to know if the register currently equals 0, just send an increment command followed by a decrement command.)

The only problem with the resulting computer is it is slow: it takes time of order 2^n to access the nth bit in the memory!

There are two strategies I know of to make a Life computer which can simulate a Turing machine with only polynomial slowdown. The first is to try to build a RAM, and the second – which has been implemented in full by Paul Callahan

```
http://www.cs.jhu.edu/~callahan/patterns/
  delay/dldesc.html
```

is to make an "extensible delay line" memory.

A strategy, which might work, to build a RAM is as follows. First, we use a puffer train, e.g. the switch engine, to leave an infinite trail of blocks behind. Perturbations in the locations of these blocks will serve as the bits in our RAM. Now, to access the Nth bit, we send a slow diagonal-moving ship out along the trail. Later, we launch a fast ship (a salvo of gliders, I would imagine) timed to catch up with and collide with the fast ship at the right location to access bit N. One must design appropriate kinds of spaceships and appropriate kinds of collisions between glider flotillas and spaceships. You want collisions to totally destroy the ships but "push" or "pull" the target block. With these you can "write" a memory bit. You also want collisions which end up sending a glider back at us, or not, depending on where the target block was (these allow "reading' a memory bit). The timing may be performed by a hardwired binary counter with B bits, allowing access to order 2^B bits of RAM. Additional bits of RAM beyond some base size will have to be accessed by programming the computer in low memory to implement a larger counter.

Callahan's strategy, which has been fully computer tested, involves an endlessly circulating stream of spaceships trapped between reflectors. The reflectors actually are NOT gates based on spaceship guns and spaceship mutual annihilation reactions. The spaceships store the memory bits. The reflectors move apart slowly (actually, in Callahan's design they can actually be commanded when and when not to move apart). You wait for the right bit to glide by and then test for its presence with the aid of a reaction that emits gliders when spaceships move by. To write a 1 to the memory we can annihilate a spaceship in the logical NOT portion of the loop by colliding something with it. We similarly write 0s.

Open questions: Conway has claimed that Life computers can be programmed to self-destruct (so that it is undecidable whether a Life pattern dies out). But his proof of this was very sketchy. Also Conway claimed Life computers could be made which would replicate themselves (like bacteria), this time without even attempting to sketch a proof. I'm unsatisfied with the rigor behind these claims. Conway has offered 50 dollars to find an explicit example of a configuration with a parent, but no grandparent. Another question: can a computer which builds itself a *2-dimensional* infinite RAM array be made, which would allow accessing the nth bit of memory in time $O(\sqrt{n})$ rather than $O(n)$?

5 Rotation matrices

First, we show that any 3D rotation matrix is representable as $A^\alpha B^\beta A^\gamma$ where α, β, and γ are fractional powers and A and B are respectively rotations about the x and y axes. I.e. every 3D rotation may be done by 3 rotations about the x, y, and x axes successively, if the angles are chosen appropriately. (And 3 is best possible since 3D rotations are a 3-degree of freedom set. Also 2 is best possible since using just one matrix A would always leave us on a great circle, clearly insufficient.)

Why is this? Well, it suffices to show that any two points on the sphere may be moved to two canonical positions (at the same distance apart). The canonical destinations of the two points are WLOG in the xz-plane with minimum average z, i.e. symmetrically placed about the z axis, i.e. $(\pm\sqrt{1-z^2}, 0, z)$.

The first rotation (about the x axis) makes the two points lie in a plane parallel to the xz plane. The second rotation (about the y axis) keeps them in this plane but moves them down (to minimum average z). The third rotation (about x) puts them in the xz plane symmetrically placed about the z axis – mission accomplished. To approximate this arbitrarily closely, we let the two fixed rotations A and B about the x and z axes rotate us through an angle $(\sqrt{5}-1)\pi$ (or any other irrational multiple of π) and then an *integer* power of each rotation will suffice to approximate any *real* rotation angle arbitrarily closely. Hence α, β, and γ could be chosen to be integers if we allow arbitrarily small error.

More generally any $n \times n$ unitary matrix may be approximated arbitrarily[2] closely by a product of two fixed unitary matrices A and B. I claim that it will suffice if A is a rotation in the (x_1, x_2) plane by an irrational multiple of π, times a rotation of complex phase by another irrational multiple of π (the two rationals being incommensurable), followed by a swap of the two coordinates (this swap allows access to the determinant -1 "flipped rotations" by taking odd integer powers of A; if we are interested in forbidding these, we could omit the swap) i.e.

$$A = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{pmatrix} e^{i\phi} \tag{10}$$

in the (x_1, x_2) plane for appropriate constants θ and ϕ. Clearly integer powers of A will allow arbitrarily close approximation of any 2×2 unitary matrix. Hence assume $n \geq 3$. I claim it will suffice if B is the n-cycle permutation $x_1 \to x_2 \to x_3 \to \ldots x_n \to x_1$. Clearly A and B suffice (to the extent that a power of A will approximate the swap $x_1 \leftrightarrow x_2$) to approximate any permutation matrix, since you can move anything anywhere by successive swaps. But this allows us to approximate arbitrary-angle

[2] Also as a special case with $\phi = 0$ of the following argument, any $n \times n$ real rotation matrix may be approximated arbitrarily closely by a product of two fixed rotation matrices A and B.

unitaries in *any* 2-plane (x_p, x_q). By the previous 3D proof we thus have also settled the case $n = 3$. But in fact we can generalize the 3D proof to become an induction on n (e.g. 4D: using our ability to do 3D rotations in arbitrary 3-spaces x_p, x_q, x_r we may then rotate 3 points within such a 3-space to make them all lie in a given 2-plane, then canonize that plane, performing an arbitrary 4D rotation, and so on). QED.

Results of this kind are important for control theory, proving linear algebra theorems, and quantum computation. One often wants to demonstrate that only a few kinds of quantum "gates" (elementary unitary transformations) will suffice to generate all unitary transformations (allowed quantum time evolutions).

6 Use of planar mechanisms made of rigid precise parts to initialize point masses to where you want them

Assuming you had a kit of perfectly rigid rods joinable with perfect rivets at their ends, then you could build a planar mechanism using $O(N)$ rods and rivets to compute

$$z = z_0 + .1(z_1 + .1(z_2 + .1(z_3 + \cdots + .1z_N))) = \sum_{k=0}^{N} 10^{-k} z_k \tag{11}$$

where the zs are 2D coordinates regarded as complex numbers, thus making a "multiscale micromanipulator." The details (this is just n linked pantographs, basically) are in my planar mechanisms paper.

Now, assuming you had the capability of "nailing down" rivets to the "wall" (or attaching them to a vertex of some rigid triangular latticework you'd built by gluing together equailateral triangles made out of more rods and rivets) so that they stayed fixed forevermore, you could then adjust your output z to N decimal places of accuracy, by moving your hands $O(N)$ times with some fixed accuracy of motion not depending on N. Each such motion of your hands, assuming you had near-optimal hands, would require energy of order k_BT I presume (since it reduces entropy by a constant addtive amount, i.e. reduces available phase space volume by a constant multiplicative factor) each time.

Admittedly the assumptions we've made are not too physically realistic (perfectly identical rigid parts, rivets with zero play) but, within those assumptions, you ought to be able to adjust coordinates to N decimal places while only expending energy $O(Nk_BT)$, and this is best possible by the 2nd law of thermodynamics.

In real life, unfortunately, parts made of real elastic materials would thermally vibrate if $T > 0$ with amplitude proportional to $\sqrt{k_BT}/c$ where c is the Hooke's law spring constant (probably the largest thinkable value in practice would arise from diamond parts). In that case N decimal place precision would require cooling some parts to exponentially low temperatures, which, even with perfect Carnot efficiency would (with parts of fixed size) would require exponentially large energy expenditure.

But still, in classical mechanics with elastic materials which were true continua, one could build robots which build smaller robots attached to their hands which build smaller robots attached to *their* hands, each robot a constant factor smaller and colder than the previous one. (Note: now we are abandoning the model of a fixed finite set of available rod lengths.) In this case we could still achieve N decimal manipulation accuracy of the last robot's finger with only polynomial in N energy expenditure.

But, of course, classical mechanics isn't true – there would be quantum zero point motion; and materials are not continua – they are made of atoms. One might argue, though, that this argument is sufficient theoretically to somewhat support my "refutation" of Church's thesis in classical mechanics with point masses, since the Turing machine we are "competing" against has to expend at least a constant "cost" per input bit (Turing machine tape costs money!).

Unfortunately for this argument, the smaller robots would have weaker muscles and bones, which seems to imply that the adjustment *time* would still be exponentially long, even though the *energy* expenditure would be only polynomial[3].

So in the final analysis, my refutation of Church's thesis doesn't seem to hold much water, in the sense that there does not seem to be a "practically achievable" way (of this nature, anyhow, and assuming we lived in a universe with the right kind of classical continuum mechanics) to outcompete Turing machines. It still holds water in the sense that there do exist initial states of N point masses which do traverse infinitely complicated trajectories in 1 minute.

7 Finite versus infinite matrices

$\exp(M) = I + M + M^2/2! + M^3/3! + \ldots$ obviously converges for finite dimensional matrices in any matrix norm, e.g. the L_2 norm or Frobenius norm, like $c^n/n!$ eventually. But it does not converge, in either of these norms, for the infinite diagonal matrix Λ with diagonal $(0, 1, 2, 3, \ldots)$. I admit that in this example, there is still elementwise convergence. But elementwise convergence is pretty worthless, since usually nobody cares about the elements of the matrix, what actually usually matters is whether the matrix series converges when applied to particular test vectors you care about. For example, consider the test vector $v = (1, 2^{-3}, 3^{-3}, 4^{-3}, \ldots)$. The nth term of the matrix series, applied to v, has infinite L_2 norm if $n > 2$, so from v's point of view, the series diverges very badly.

This illustrates an important concept: the set of vectors to which it is well defined to apply some operator is called the operator's *domain*. In this case v is not in the domain of Λ, but $(1, 3^{-1}, 3^{-2}, 3^{-3}, \ldots)$ is. For finite dimensional matrices, you don't need the concept of a domain, since everything is in the domain, but for operators, this concept is very important.

[3] Although if the grasped mass were zero, then only constant time would be required to get the last robot's finger accurate to N decimals, since the increasing weakness of the robots would be amply compensated for by the fact they have to handle smaller masses (their child robots).

You can kill elementwise convergence by rotating to another basis (e.g. containing a bad vector as a basis vector), but you can't kill L_2-norm convergence that way, which is one reason L_2-norm convergence is an important concept and elementwise convergence isn't.

Here is another amusing example: consider $\exp(D) = I + D + D^2/2! + D^3/3! + \ldots$ where D is the differentiation *operator* $D = \mathrm{d}/\mathrm{d}x$. Then the Taylor series theorem may be stated $\exp(cD)f(x) = f(x+c)$ at points x where the radius of convergence of the Taylor series for f is $> c$. However, for appropriate c and $f(x)$ such as $c = 1$, $f(x) = 1/sin(6x + .1i)$, this is never the case for any real x, so in this case the operator series does not converge from the point of view of this particular test vector.

$AB = I \Rightarrow BA = I$ is true for finite dimensional matrices, i.e. left inverse is the same as right inverse. This may be shown by considering numerical algorithms for computing matrix inverses. These algorithms succeed in producing a right inverse (say) of A iff $\det A \neq 0$. Similarly they succeed in producing a left inverse iff $\det A \neq 0$. (The expansion of determinant as a sum of permutations is clearly invariant to matrix transposition, so the "left" and "right" determinants are the same.) So a left inverse exists iff a right inverse exists. But if both exist, they must be the same. Proof: suppose not, i.e. $AB = CA = I$ but $B \neq C$. Then $B = IB = (CA)B = C(AB) = CI = C$.

However, $AB = I \Rightarrow BA = I$ is *false* for these two infinite matrices:

$$A = \begin{pmatrix} 0 & 0 & 0 & 0 & \cdots \\ 1 & 0 & 0 & 0 & \cdots \\ 0 & 1 & 0 & 0 & \cdots \\ 0 & 0 & 1 & 0 & \ddots \\ \vdots & \vdots & \vdots & \vdots & \end{pmatrix}, \quad B = \begin{pmatrix} 0 & 1 & 0 & 0 & \cdots \\ 0 & 0 & 1 & 0 & \cdots \\ 0 & 0 & 0 & 1 & \cdots \\ 0 & 0 & 0 & 0 & \cdots \\ \vdots & \vdots & \vdots & \ddots & \end{pmatrix} \tag{12}$$

These are the "shift operators," so called because they shift the positive integers left and right by 1. I.e. the right shift performs the permutation $x_1 \to x_2 \to x_3 \to \cdots$. $AB = I$. Unfortunately the left shift irrecoverably loses x_1 so

$$BA = \begin{pmatrix} 0 & 0 & 0 & 0 & \cdots \\ 0 & 1 & 0 & 0 & \cdots \\ 0 & 0 & 1 & 0 & \cdots \\ 0 & 0 & 0 & 1 & \cdots \\ \vdots & \vdots & \vdots & \ddots & \end{pmatrix} \neq I. \tag{13}$$

For finite dimensional square matrices A,B, can $\|I + AB - BA\| < 1$? ($\|\cdot\|$ meaning matrix L_2 norm.) What about for infinite dimensional square matrices?

The answer [I got this problem from something by P.R.Halmos. Also solved by Jade P. Vinson] is:

In the finite n-dimensional case, let $M = I + AB - BA$. Then $\mathrm{Trace}(AB - BA) = 0$ so $\mathrm{Trace}(M) = n$ so it is impossible for all eigenvalues of M to lie inside the unit circle in the complex plane and still sum to n, hence (since $\|M\|$ is the biggest absolute value of an eigenvalue of M) $\|I + AB - BA\| \geq 1$.

In the infinite case: Well, as every quantum mechanics student knows, $DZ - ZD = I$ where $D = \mathrm{d}/\mathrm{d}z$ and Z is the operator that multiplies by z. (This is the commutation relation for position and momentum operators.) So then if we take as an orthogonal basis for a Hilbert space $z, z^2, z^3, \ldots$ (these are orthogonal on the unit circle) then

```
    00000        01000         10000         00000
    10000        00200         02000         01000
Z = 01000    D = 00030    DZ = 00300    ZD = 00200
    00100        00004         00040         00030
    00010        00000         00005         00004
     ...          ...           ...           ...
```

and $I + ZD - DZ$ is the 0 matrix, so $\|I + ZD - DZ\| = 0$. And in fact I claim $\|I + AB - BA\|$ can be any non-negative real.

P.R.Halmos [Commutators of operators II, Amer J. Math. 76,1 (1954) 191-198 & reprinted in Selecta of Halmos] gave the same proofs. He also showed such fine theorems as the fact that any Hermitian operator is a sum of two commutators and hence any operator is a sum of 4 commutators.

As an *open* problem, Halmos asked what the infemal value of $\|I + AB - BA\|$ when A,B are *bounded* operators (i.e. of bounded L_2 norm). Halmos managed by a pretty complicated analysis to show the existence of A,B so that $|I + AB - BA| < 0.97$. Later, in ["A glimpse into Hilbert space" (in T.L.Saaty ed.: Lectures in modern math, Wiley 1963] he claimed a re-analysis of his same construction gets .97 down to $8/9 = .888\ldots$.

Anyway, the **moral** of this exercise is *watch out*! Operators and infinite matrices *cannot* just be treated like finite dimensional matrices. Many theorems fail. Operators feature much richer behavior and a lot more gotchas. A lot of physicists act all their lives as though operators were just matrices and never learn any operator analysis. That is a dangerous path to take. If you want to prove rigorous results in quantum mechanics you have to know operator analysis.

Indeed, it has been said that matrices are "the language of science." You have to know linear algebra to do science and engineering. Well if so, operators are "the language of quantum mechanics!" Not to mention, lots of other math (ODEs, PDEs, path integrals) may be done with tools of operator analysis. The gradual buildup of rigorous theoretical tools for handling operator analysis has been one of the big triumphs of mathematics and mathematical physics during the 20th century, and development continues.

Unfortunately, operator analysis is hard, there is a lot of it, universities don't seem to offer courses in the subject. These 2 books are useful references: Akhiezer & Glazman: Theory of linear operators in Hilbert space, Dover (2 vols in 1). Reed & Simon: Mehtods of modern mathematical physics (4 vols) Academic press.