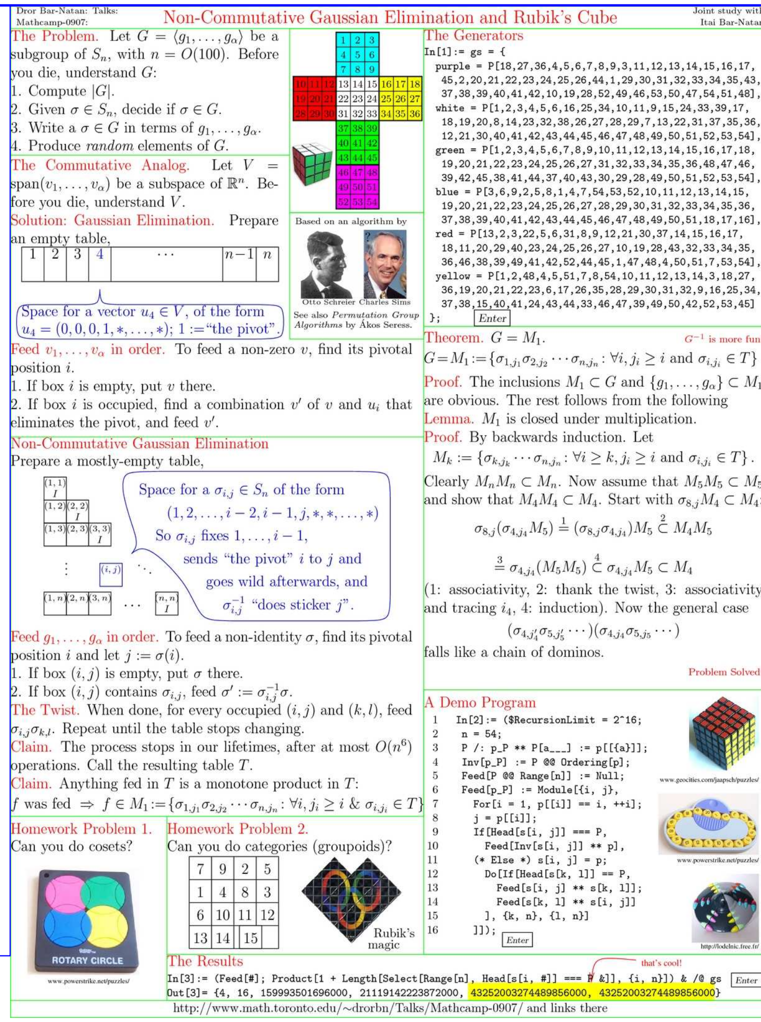
The Hardest Math I've Ever Really Used, 1

Dror Bar−Natan at St. Bonaventure Univesrity
October 15, 2011

Abstract. What's the hardest math I've ever used in real life? Me, myself, directly - not by using a cellphone or a GPS device that somebody else designed? And in "real life" — not while studying or teaching mathematics?
I use addition and subtraction daily, adding up bills or calculating change. I use percentages often, though mostly it is just "add 15 percents". I seldom use multiplication and division: when I buy in bulk, or when I need to know how many tiles I need to replace my kitchen floor. I've used powers twice in my life, doing calculations related to mortgages. I've used a tiny bit of geometry and algebra for a tiny bit of non-math-related computer graphics I've played with. And for a long time, that was all. In my talk I will tell you how recently a math topic discovered only in the 1800s made a brief and modest appearance in my non-mathematical life. There are many books devoted to that topic and a lot of active research. Yet for all I know, nobody ever needed the actual formulas for such a simple reason before.
Hence we'll talk about the motion of movie cameras, and the fastest way to go from A to B subject to driving speed limits that depend on the locale, and the "happy segway principle" which is a the heart of the least action principle which in itself is at the heart of all of modern physics, and finally, about that funny discovery of Janos Bolyai's and Nikolai Ivanovich Lobachevsky's, that the famed axiom of parallels of the ancient Greeks need not actually be true.



Dror Bar-Natan: Talks: Mathcamp-0907:

Non-Commutative Gaussian Elimination and Rubik's Cube

Joint study with Itai Bar-Natan

The Problem. Let $G = \langle g_1, \dots, g_\alpha \rangle$ be a subgroup of S_n, with $n = O(100)$. Before you die, understand G:
1. Compute $|G|$.
2. Given $\sigma \in S_n$, decide if $\sigma \in G$.
3. Write a $\sigma \in G$ in terms of $g_1, \dots, g_\alpha$.
4. Produce *random* elements of G.

The Commutative Analog. Let $V = \mathrm{span}(v_1, \dots, v_\alpha)$ be a subspace of $\mathbb{R}^n$. Before you die, understand V.

Solution: Gaussian Elimination. Prepare an empty table,

1	2	3	4	$\cdots$	$n-1$	n

Space for a vector $u_4 \in V$, of the form $u_4 = (0,0,0,1,*,\dots,*)$; 1 := "the pivot".

Feed $v_1, \dots, v_\alpha$ in order. To feed a non-zero v, find its pivotal position i.
1. If box i is empty, put v there.
2. If box i is occupied, find a combination v' of v and u_i that eliminates the pivot, and feed v'.

Based on an algorithm by Otto Schreier, Charles Sims. See also *Permutation Group Algorithms* by Ákos Seress.

Non-Commutative Gaussian Elimination
Prepare a mostly-empty table,

Space for a $\sigma_{i,j} \in S_n$ of the form $(1, 2, \dots, i-2, i-1, j, *, *, \dots, *)$. So $\sigma_{i,j}$ fixes $1, \dots, i-1$, sends "the pivot" i to j and goes wild afterwards, and $\sigma_{i,j}^{-1}$ "does sticker j".

Feed $g_1, \dots, g_\alpha$ in order. To feed a non-identity σ, find its pivotal position i and let $j := \sigma(i)$.
1. If box (i,j) is empty, put σ there.
2. If box (i,j) contains $\sigma_{i,j}$, feed $\sigma' := \sigma_{i,j}^{-1}\sigma$.

The Twist. When done, for every occupied (i,j) and (k,l), feed $\sigma_{i,j}\sigma_{k,l}$. Repeat until the table stops changing.
Claim. The process stops in our lifetimes, after at most $O(n^6)$ operations. Call the resulting table T.
Claim. Anything fed in T is a monotone product in T:
f was fed $\Rightarrow f \in M_1 := \{\sigma_{1,j_1}\sigma_{2,j_2}\cdots\sigma_{n,j_n} : \forall i, j_i \geq i \;\&\; \sigma_{i,j_i} \in T\}$

The Generators
```
In[1]:= gs = {
  purple = P[18,27,36,4,5,6,7,8,9,3,11,12,13,14,15,16,17,
    45,2,20,21,22,23,24,25,26,44,1,29,30,31,32,33,34,35,43,
    37,38,39,40,41,42,10,19,28,52,49,46,53,50,47,54,51,48],
  white = P[1,2,3,4,5,6,16,25,34,10,11,9,15,24,33,39,17,
    18,19,20,8,14,23,32,38,26,27,28,29,7,13,22,31,37,35,36,
    12,21,30,40,41,42,43,44,45,46,47,48,49,50,51,52,53,54],
  green = P[1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,
    19,20,21,22,23,24,25,26,27,31,32,33,34,35,36,48,47,46,
    39,42,45,38,41,44,37,40,43,30,29,28,49,50,51,52,53,54],
  blue = P[3,6,9,2,5,8,1,4,7,54,53,52,10,11,12,13,14,15,
    19,20,21,22,23,24,25,26,27,28,29,30,31,32,33,34,35,36,
    37,38,39,40,41,42,43,44,45,46,47,48,49,50,51,18,17,16],
  red = P[13,2,3,22,5,6,31,8,9,12,21,30,37,14,15,16,17,
    18,11,20,29,40,23,24,25,26,27,10,19,28,43,32,33,34,35,
    36,46,38,39,49,41,42,52,44,45,1,47,48,4,50,51,7,53,54],
  yellow = P[1,2,48,4,5,51,7,8,54,10,11,12,13,14,3,18,27,
    36,19,20,21,22,23,6,17,26,35,28,29,30,31,32,9,16,25,34,
    37,38,15,40,41,24,43,44,33,46,47,39,49,50,42,52,53,45]
};   Enter
```

Theorem. $G = M_1$. G^{-1} is more fun!
$G = M_1 := \{\sigma_{1,j_1}\sigma_{2,j_2}\cdots\sigma_{n,j_n} : \forall i, j_i \geq i \text{ and } \sigma_{i,j_i} \in T\}$.
Proof. The inclusions $M_1 \subset G$ and $\{g_1, \dots, g_\alpha\} \subset M_1$ are obvious. The rest follows from the following
Lemma. M_1 is closed under multiplication.
Proof. By backwards induction. Let
$$M_k := \{\sigma_{k,j_k}\cdots\sigma_{n,j_n} : \forall i \geq k, j_i \geq i \text{ and } \sigma_{i,j_i} \in T\}.$$
Clearly $M_nM_n \subset M_n$. Now assume that $M_5M_5 \subset M_5$ and show that $M_4M_4 \subset M_4$. Start with $\sigma_{8,j}M_4 \subset M_4$:
$$\sigma_{8,j}(\sigma_{4,j_4}M_5) \stackrel{1}{=} (\sigma_{8,j}\sigma_{4,j_4})M_5 \stackrel{2}{\subset} M_4M_5$$
$$\stackrel{3}{=} \sigma_{4,j_4}(M_5M_5) \stackrel{4}{\subset} \sigma_{4,j_4}M_5 \subset M_4$$
(1: associativity, 2: thank the twist, 3: associativity and tracing i_4, 4: induction). Now the general case
$$(\sigma_{4,j_4'}\sigma_{5,j_5'}\cdots)(\sigma_{4,j_4}\sigma_{5,j_5}\cdots)$$
falls like a chain of dominos.

Problem Solved!

A Demo Program
```
In[2]:= ($RecursionLimit = 2^16;
  n = 54;
  P /: p_P ** P[a___] := p[[{a}]];
  Inv[p_P] := P @@ Ordering[p];
  Feed[P @@ Range[n]] := Null;
  Feed[p_P] := Module[{i, j},
    For[i = 1, p[[i]] == i, ++i];
    j = p[[i]];
    If[Head[s[i, j]] === P,
      Feed[Inv[s[i, j]] ** p],
    (* Else *) s[i, j] = p;
      Do[If[Head[s[k, l]] == P,
        Feed[s[i, j] ** s[k, l]];
        Feed[s[k, l] ** s[i, j]]
      ], {k, n}, {l, n}]
    ]]);   Enter
```
www.geocities.com/jaapsch/puzzles/
www.powerstrike.net/puzzles/
http://lodelnic.free.fr/

Homework Problem 1. Can you do cosets?

ROTARY CIRCLE
www.powerstrike.net/puzzles/

Homework Problem 2. Can you do categories (groupoids)?

7	9	2	5
1	4	8	3
6	10	11	12
13	14	15	

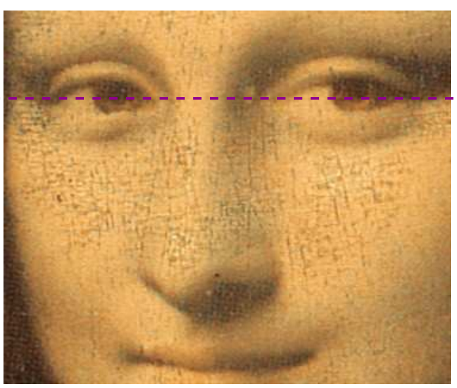
Rubik's magic

The Results
```
In[3]:= (Feed[#]; Product[1 + Length[Select[Range[n], Head[s[i, #]] === P &]], {i, n}]) & /@ gs   Enter
Out[3]= {4, 16, 159993501696000, 2119142223872000, 43252003274489856000, 43252003274489856000}
```
that's cool!

http://www.math.toronto.edu/~drorbn/Talks/Mathcamp-0907/ and links there

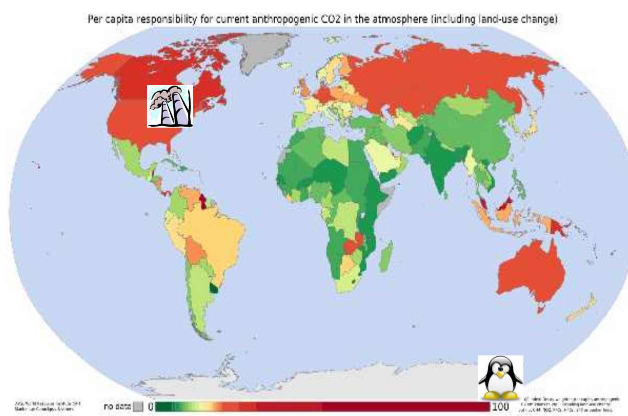
I could be a mathematician ...

...or an art historian...

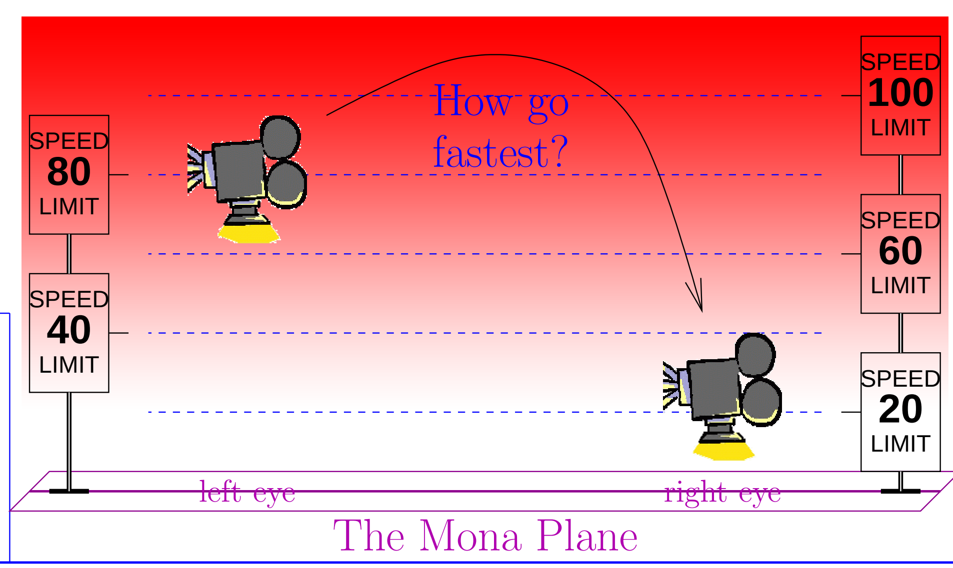


...or an environmentalist.

Al Gore in Futurama, circa 3000AD



Goal. Find the least-blur path to go from Mona's left eye to Mona's right eye in fixed time. Alternatively, fix your blur-tolerance, and find the fastest path to do the same. For fixed blur, our camera moves at a speed proportional to its distance from the image plane:

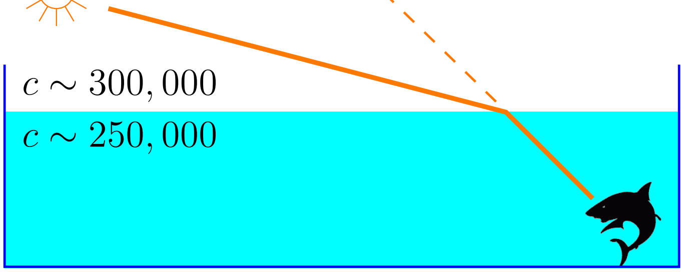


http://www.math.toronto.edu/~drorbn/Talks/StBonaventure−1110/

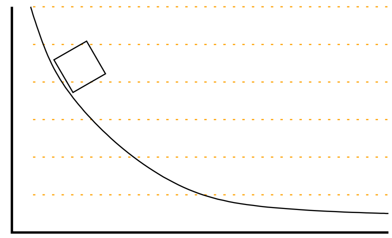
The Hardest Math I've Ever Really Used, 2

Picture credits. Mona: Leonrado; Al Gore: Futurama; Map 1: en.wikipedia.org/wiki/Greenhouse_gas; Smokestacks: gbuapcd.org/complaint.htm; Penguin: brentpabst.com/bp/2007/12/15/BrentGoesPenguin.aspx; Map 2: flightpedia.org; Segway: co2calculator.wordpress.com/2008/10; Lobachevsky: en.wikipedia.org/wiki/Nikolai_Lobachevsky; Eschers: www.josleys.com/show_gallery.php?galid=325;

Fermat's Principle



The Brachistochrone



$$mgh = \tfrac{1}{2}mv^2$$

Bernoulli on Newton. "I recognize the lion by his paw".

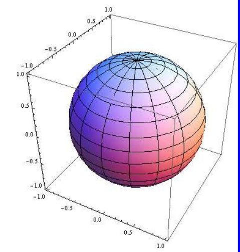
Flatlanders airline route map

576
252
167
131
112
103
100
103
112

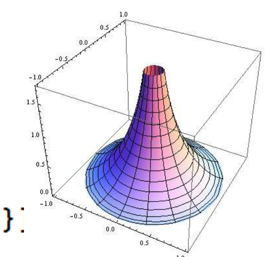
The Least Action Principle. Everywhere in physics, a system goes from A to B along the path of least action.

With small print for quantum mechanics.

```
ParametricPlot3D[{
  Sin[u] Cos[v],
  Sin[u] Sin[v],
  Cos[u]
 }, {u, 0, π}, {v, 0, 2 π}]
```

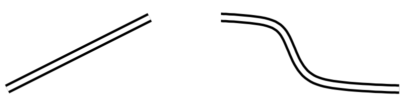


```
ParametricPlot3D[{
  Sech[u] Cos[v],
  Sech[u] Sin[v],
  u - Tanh[u]
 }, {u, 0, e}, {v, 0, 2 π}]
```

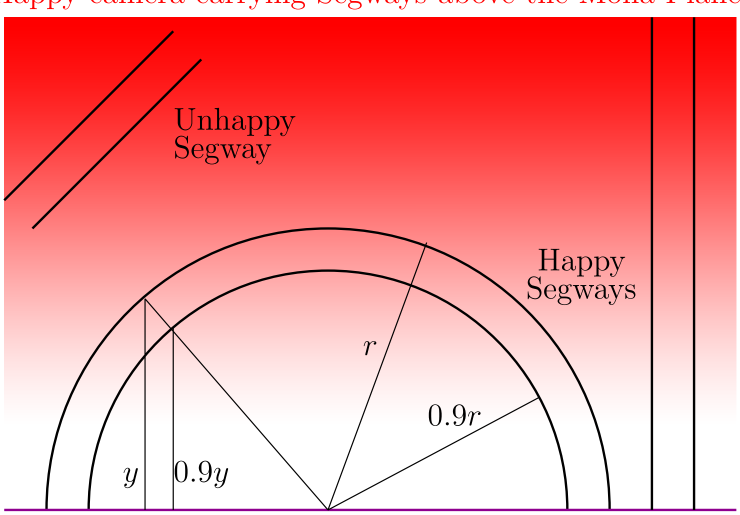


The Happy Segway Principle

A Segway is happy iff both its wheels are

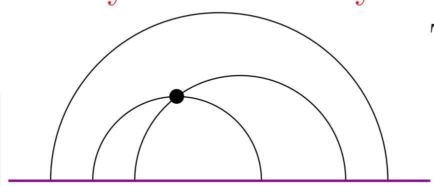


Happy camera-carrying Segways above the Mona Plane



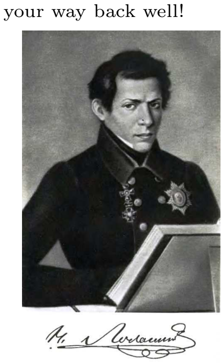
The Mona Plane

The Bolyai-Lobachevsky Plane



Two parallels through one point

Further Fun Facts. • In small scale, $\pi^H \to \pi^E$. In large scale, $\pi^H \to \infty$. • The sum of the angles of a triangle is always less than π. In fact, sum+area= π, so the largest possible area of a triangle is π. • If your friend walks away, she'll drop out of sight before you know it. • There are so many places just a stone throw away! But you'd better remember your way back well!



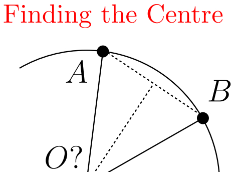
The Actual Code

```
p3.y = p2.y + b*x3p;
x = p1.x-p2.x; y = p1.y-p2.y;
d1 = p1.d; d2 = p2.d;
norm = sqrt(x*x + y*y);
a = x/norm; b = y/norm;
x1p = a*x + b*y;
x0 = (x1p + (d1*d1-d2*d2)/x1p)/2;
r = sqrt((x1p-x0)*(x1p-x0)+d1*d1);
x1pp = (x1p-x0)/r; x2pp = -x0/r;
theta1 = acos(x1pp);
theta2 = acos(x2pp);
t1 = log(tan(theta1/2));
t2 = log(tan(theta2/2));
t3 = t1 + s*(t2-t1);
theta3 = 2*atan(exp(t3));
x3pp = cos(theta3);
d3pp = sin(theta3);
x3p = x0 + r*x3pp;
p3.d = r*d3pp;
p3.x = p2.x + a*x3p;
```

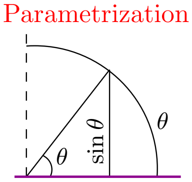
Ops used. $+$, $-$, $\times$, $\div$, $\sqrt{\ }$, cos, sin, tan, arccos, arctan, log, exp.

Some further basic geometry also occurs:

Finding the Centre



Parametrization



$$\theta'(t) = \sin\theta(t)$$
$$\Downarrow$$
$$\theta = 2\arctan e^t$$