Baserunner's Optimal Path

By Davide Carozza, Stewart Johnson, and Frank Morgan

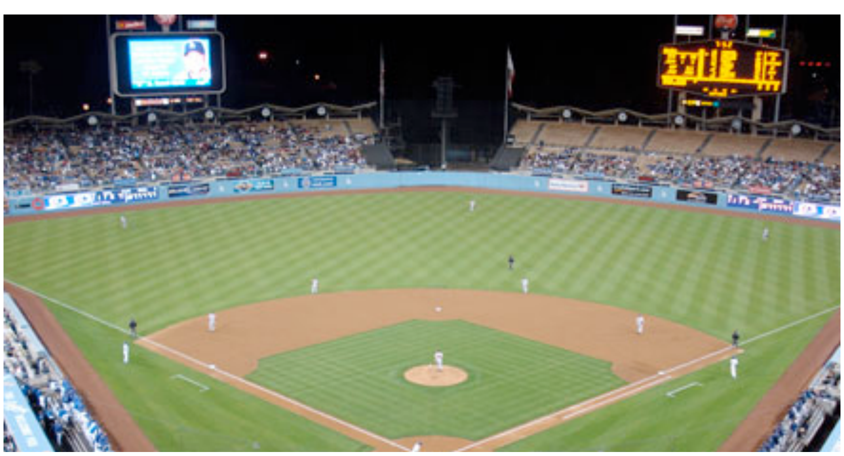
When you hit that final long ball in the World Series of Baseball and know you need the home run, what is your optimal path around the bases? If you run straight for first, you either have to slow to a near stop or go sailing far beyond into the outfield. The standard recommended "banana" path follows the baseline maybe halfway and then veers a bit to the right to come at first base from a better angle to continue towards second. That cannot be ideal. It would have been better to start at an angle to the right to head directly to an outer point on the banana path.

So what is the optimal path? Using a very simple model, we obtain the path of Figure 1. You start out heading about 25˚ right of the base line and run with acceleration of constant maximum magnitude σ, as illustrated by the vectors decorating the path. You slow down a bit coming into first, hit a local maximum speed as you cross second, and start the final acceleration home a bit before crossing third base (see Figure 2). The total time around the bases is about $52.7/\sqrt{\sigma}$, about 16.7 seconds for $\sigma = 10$ ft/sec^2, about 25% faster than following the baseline for 22.2 seconds (coming to a full stop at 1st, 2nd, and 3rd base) and about 6% faster than following a circular path for 17.8 seconds. The record time according to Guiness [G] is 13.3 seconds, set by Evar Swanson at Columbus, Ohio in 1932. His average speed around the bases was about 18.5 mph or 27 ft/sec.

Is it legal to run so far outside the base path? The relevant official rule of Baseball says:

> **7.08 Any runner is out when—**
> (a) (1) He runs more than three feet away from his baseline to avoid being tagged unless his action is to avoid interference with a fielder fielding a batted ball. A runner' s baseline is established when the tag attempt occurs and is a straight line from the runner to the base he is attempting to reach safely.

The rule just says that *after a tag attempt* the runner cannot deviate more than three feet from a straight line *from that point.* The rule doesn't apply until the slugger is almost home, when our fastest path is nearly straight. So our path is legal.



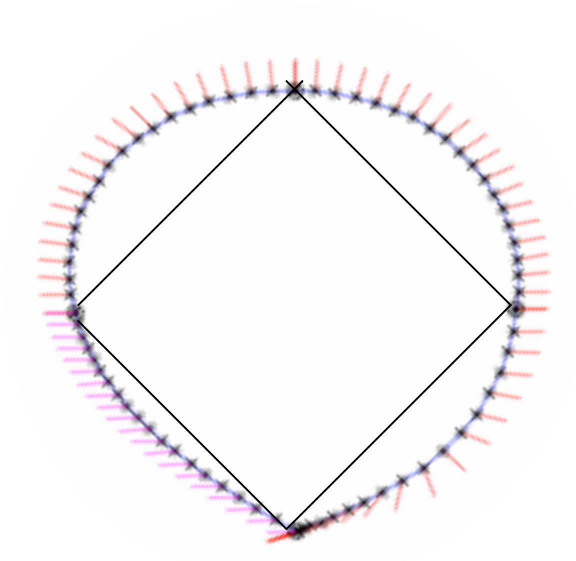


Figure 1. Second picture shows the fastest path around the bases given a bound σ on the magnitude of the acceleration vector, shown at each point. First picture from http://www.bsideblog.com/images/2008/03/baseball-diamond.jpg.

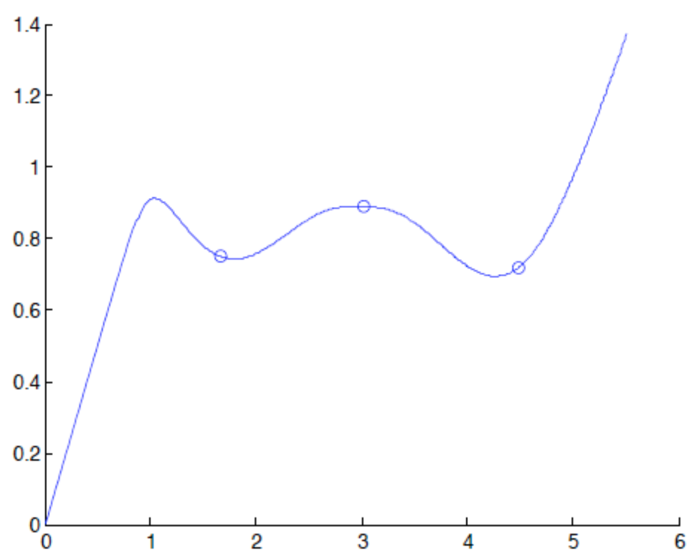


Figure 2. Speed as a function of time. For $\sigma = 10$ ft/sec^2, each unit of time represents 3 seconds and each unit of velocity represents 30 ft/sec. The times for each segment are about 5.1, 4.1, 4.4, and 3.1 seconds, for a total of about 16.7 seconds.

Our model simply assumes a bound σ on the magnitude of the baserunner's acceleration (which includes deceleration and curvature). The locus of the fastest path around the bases is independent of σ because you can scale velocity by λ, acceleration by λ^2, and time by $1/\lambda$. So slow runners should follow the same route as fast ones. At first you might think that a very slow, awkward runner should just walk directly from base to base, except that he'd likely fall down trying to make the sharp turn at first.

To find the fastest path around the bases, we consider the simpler problem of finding the fastest path between two points, given the initial and final velocities, which has a unique solution. Intriguingly enough, for this problem, *total time is not continuous in the prescribed conditions.* Even on the line, consider starting at the origin with initial velocity 1 and going at maximal acceleration for a second, ending with velocity 2; now if, instead, the prescribed final velocity were increased a bit, you would have to start out by decelerating to velocity 0, go backwards to well left of the origin, and then accelerate right to the terminus. . (See Remark after Lemma 1. Fortunately time is lower-semicontinuous, which is what we need to prove the existence of fastest paths.)

For a critical path between bases, the acceleration has constant magnitude σ and remarkably is given by $At + B$ normalized, for some constant vectors A, B. In velocity space, such paths are portions of catenaries (the famous least-energy shape of hanging cables as for suspension bridges), which in general can be absolute minima, local minima, or unstable critical points (see Remarks after Lemma 2).

It is easy to see that a fastest path for bounded $|a|$ also minimizes $\max |a|$ for given time, since if you could reduce $\max |a|$, then by increasing speed along an appropriate portion of the locus in space, you could reduce time. There are, however, more solutions to the second problem. In the example at the end of the Remarks after Lemma 2, all three paths minimize $\max |a|$ for given times $T_1 < T_2 < T_3$.

Given the fastest path between bases for prescribed velocities, we find the shortest path around the bases by minimizing over all choices of velocity at the bases, specifying velocity 0 at the start. We think that the solution is unique, but we know no proof.

Our model is, of course, an oversimplified one, since it assumes that maximum deceleration equals maximum acceleration and that maximum acceleration remains possible at high speeds; taking $\sigma = 10$ ft/sec^2, it leads to a final speed coming into home of about 42 ft/sec, faster than the highest recorded human speed as of August 2009 of 40.5 ft/sec by Usain Bolt, even though his initial acceleration exceeded 18 ft/sec^2 [S].

Acknowledgements. This work stemmed from a Williams College undergraduate colloquium talk by Carozza advised by Morgan. Johnson, who was in the audience, discovered the remarkably simple critical condition and computed the fastest path of Figure 1. Morgan acknowledges NSF support.

2. Fastest Paths

Lemmas 1 and 2 provide existence and structure for the shortest path between two bases, given initial and final velocities. Proposition 1 considers the full baserunner problem with all four bases. We conclude by explaining our numerical solution of Figure 1.

Lemma 1. *There exists a fastest path from one point to another in the plane, given initial velocity, final velocity, and a bound $\sigma > 0$ on the acceleration. The minimum time is a lower-semicontinuous function of the initial and final positions and velocities.*

Remark. The minimum time is not continuous in the prescribed conditions. For example, for $\sigma = 1$, the fastest path from (0, 0) to $(v_0 + 1/2, 0)$ with initial velocity $v_0 > 0$ and final velocity $v_0 + 1 + \varepsilon$ is for $\varepsilon = 0$ simply forward motion for 1 second at unit acceleration, but for small $\varepsilon > 0$ one must decelerate for v_0 seconds to velocity 0 at $(.5\, v_0^2, 0)$, move backwards, accelerating and decelerating for another $\sqrt{2}\, v_0$ seconds to come to rest just left of (0, 0), and then move forward for a bit more than a second at unit acceleration, for total time a bit more than $1 + v_0(1+\sqrt{2})$ seconds, a huge discontinuity if v_0 is large. See Figure 3 for the case $v_0 = 1$. In summary, to increase the final velocity of a linear path with maximum acceleration involves backing up and a discontinuous increase in total time.

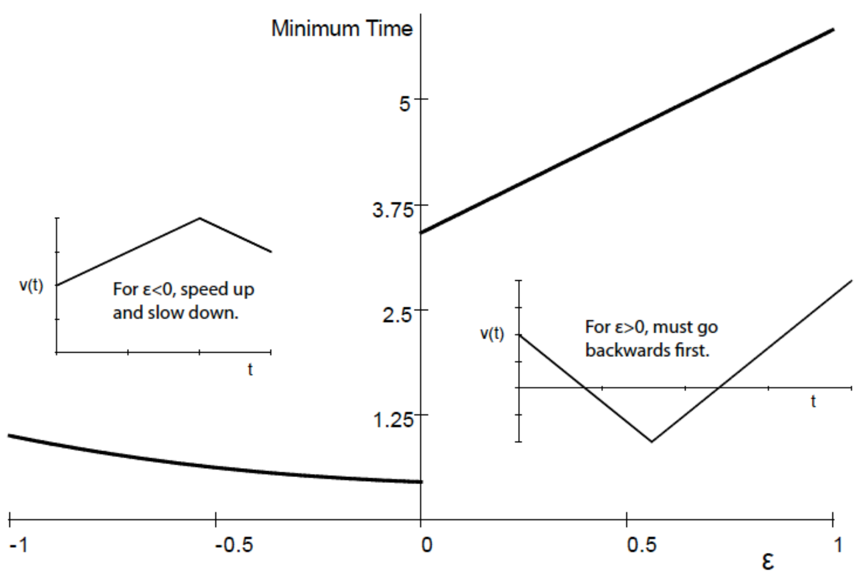


Figure 3. As the prescribed final velocity increases past that obtained by constant maximum acceleration, the fastest path has to back up, with a discontinuous increase in total time.

Proof of Lemma 1. First we note that there exists some path satisfying the conditions. If the given velocities are 0, this is obvious. Otherwise just follow the given initial direction with maximum negative acceleration until obtaining velocity 0, and similarly backwards from the terminal point, to reduce to the obvious case. This path bounds the minimum time and hence the positions and velocities. Except for the trivial case when the initial and final position and velocities coincide, there is also a lower bound on the total time.

To prove simultaneously existence and lower-semicontinuity in the prescribed conditions, consider a sequence of paths with conditions converging to the prescriptions and times T_i converging to the infimum T. We may assume that the velocities are bounded functions from $[0, T_i]$ into R^2 with Lipschitz constant at most σ and that $T_i \leq 2T$. Rescale time to change the domain to $[0, T]$. Now each velocity has Lipschitz constant at most $\sigma T_i/T \leq 2\sigma$ and the conditions still converge to the prescriptions. By the compactness of uniformly bounded Lipschitz functions, we may assume that the velocities and hence the paths converge; the limit has time T as desired.

Lemma 2. *For a fastest $C^{1,1}$ path from one point to another in the plane, given initial velocity, final velocity, and a bound $\sigma > 0$ on the magnitude of the acceleration a,*

$$a = \sigma \frac{At + B}{|At + B|}$$

for some constant vectors A, B. Such a fastest path is unique.

Remarks. By a translation in time, we may assume that $B \cdot A = 0$ and that A is a unit vector. The path is real-analytic in time unless $B = 0$ and $t = 0$, when a flips direction. In addition the path in space can have a singularity where the velocity vanishes, as in Figure 4b.

Up to rotation and translation in the plane and scaling in time and space, we may assume that

$$a = \frac{(1,t)}{\sqrt{1+t^2}} \ ,$$

$$v = (\text{arcsinh}\ t, \sqrt{1+t^2}\ ) + v_0\ ,$$

$$x = \left(t\ \text{arcsinh}\ t - \sqrt{1+t^2}\ , .5\ t\sqrt{1+t^2} + .5\ \text{arcsinh}\ t\right) + v_0\, t\ ,$$

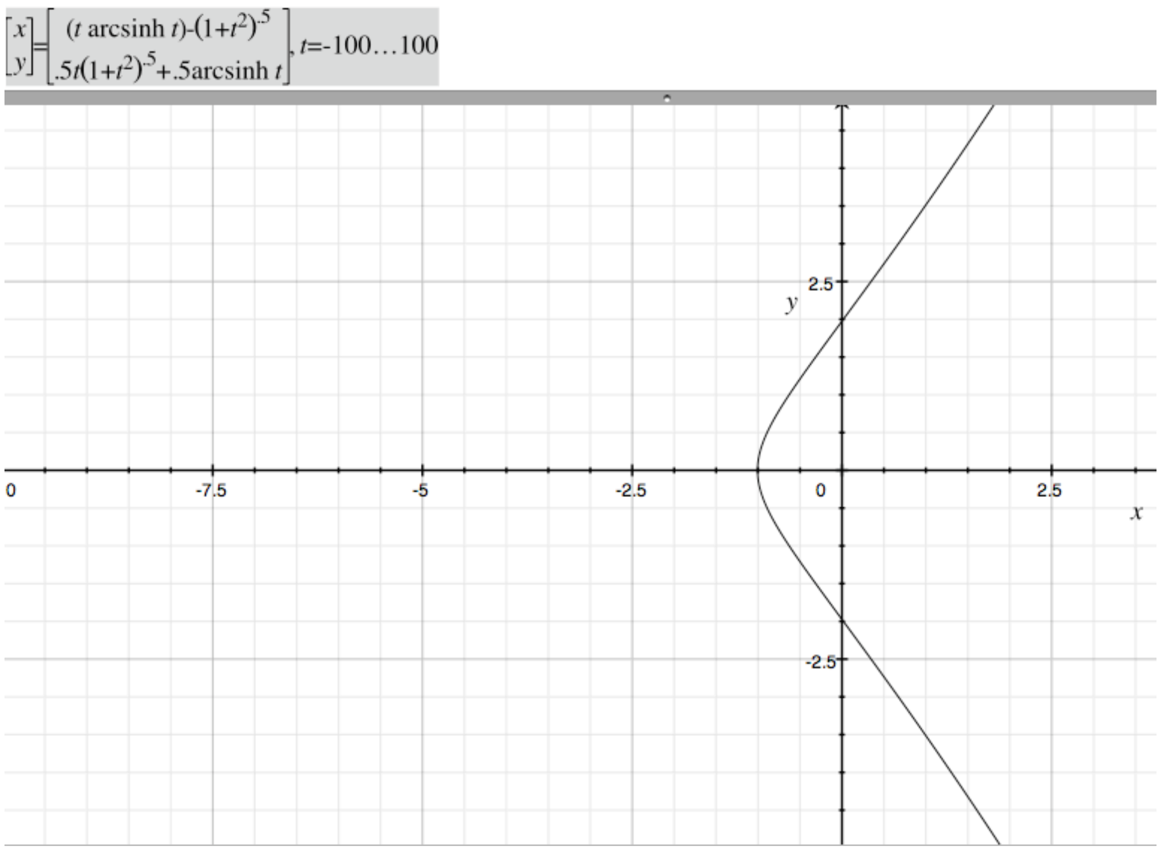
pictured for $v_0 = 0$, $v_0 = -(0,1)$, $v_0 = -(0,10)$, and $v_0 = -(1,0)$ in Figure 4; or in the degenerate case

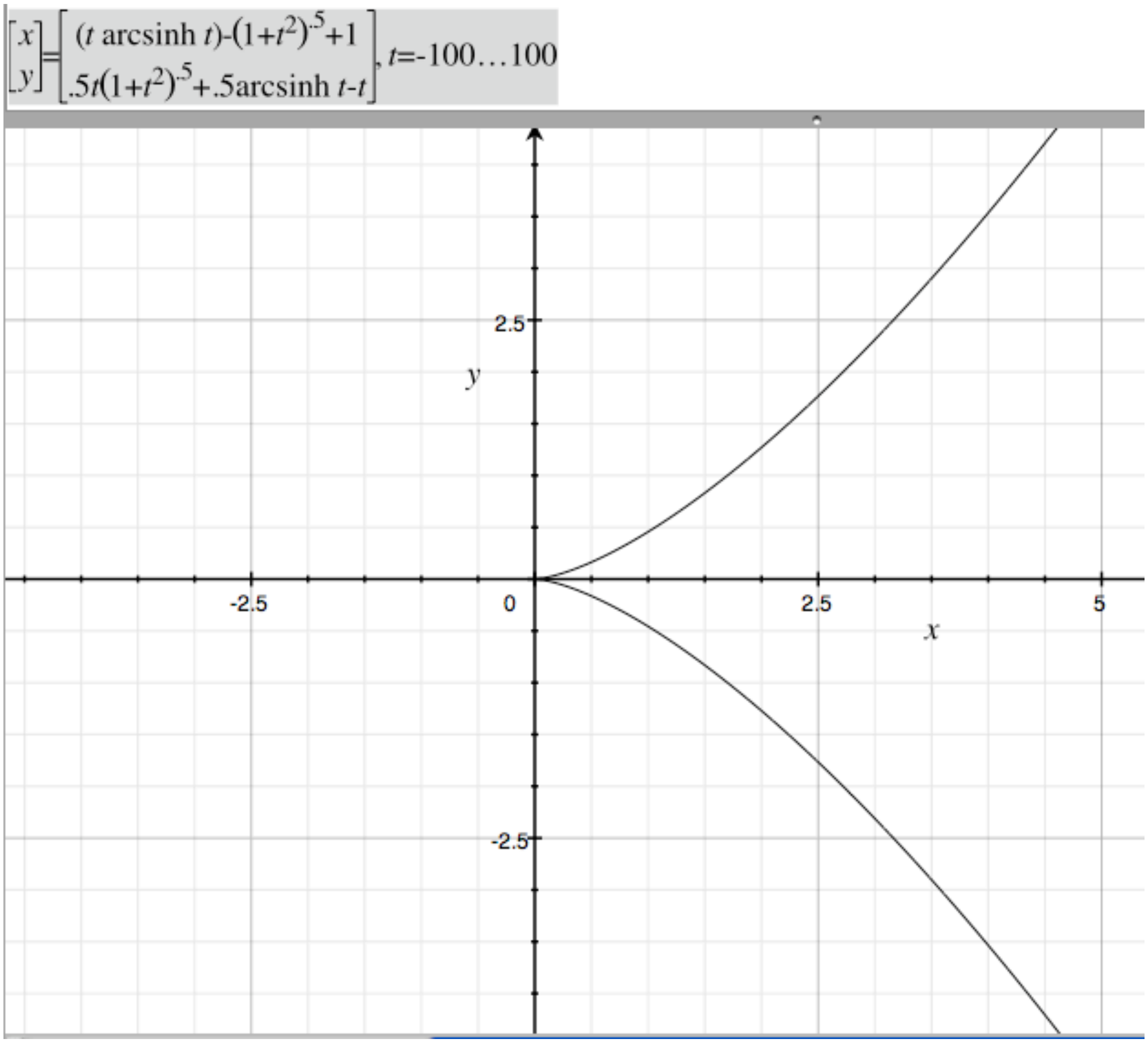
$$a = (\text{sign}\ t = \pm 1, 0)\ ,$$

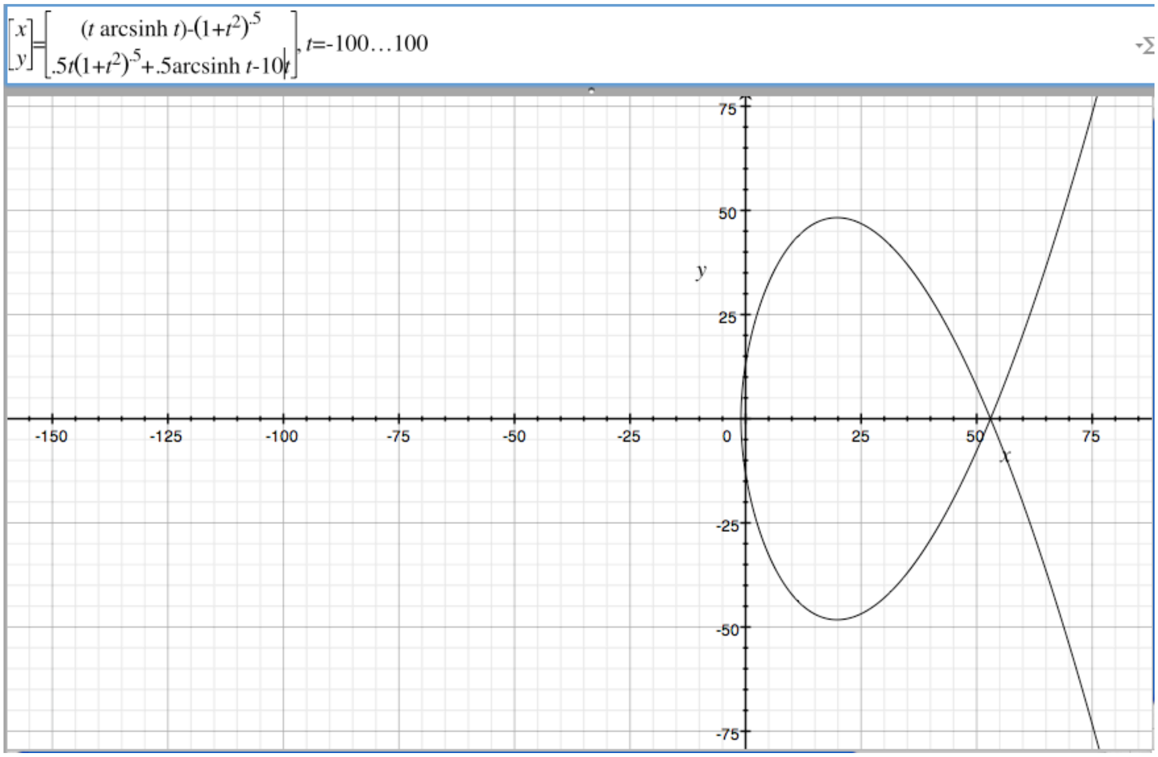
$$v = (-1 \pm t, 0) + v_0\ ,$$

$$x = (-t \pm .5t^2,\ 0) + v_0 t,$$

pictured for $v_0 = -(0,1)$ in Figure 5.




[x; y] = [(t arcsinh t)-(1+t^2)^.5+1; .5t(1+t^2)^.5+.5arcsinh t-t], t=-100…100
2.5
y
-2.5
0
2.5
5
x
-2.5


[x; y] = [(t arcsinh t)-(1+t^2)^.5 ; .5t(1+t^2)^.5+.5arcsinh t-10t], t=-100…100
75
50
25
0
-25
-50
-75
-150
-125
-100
-75
-50
-25
25
50
75
x
y

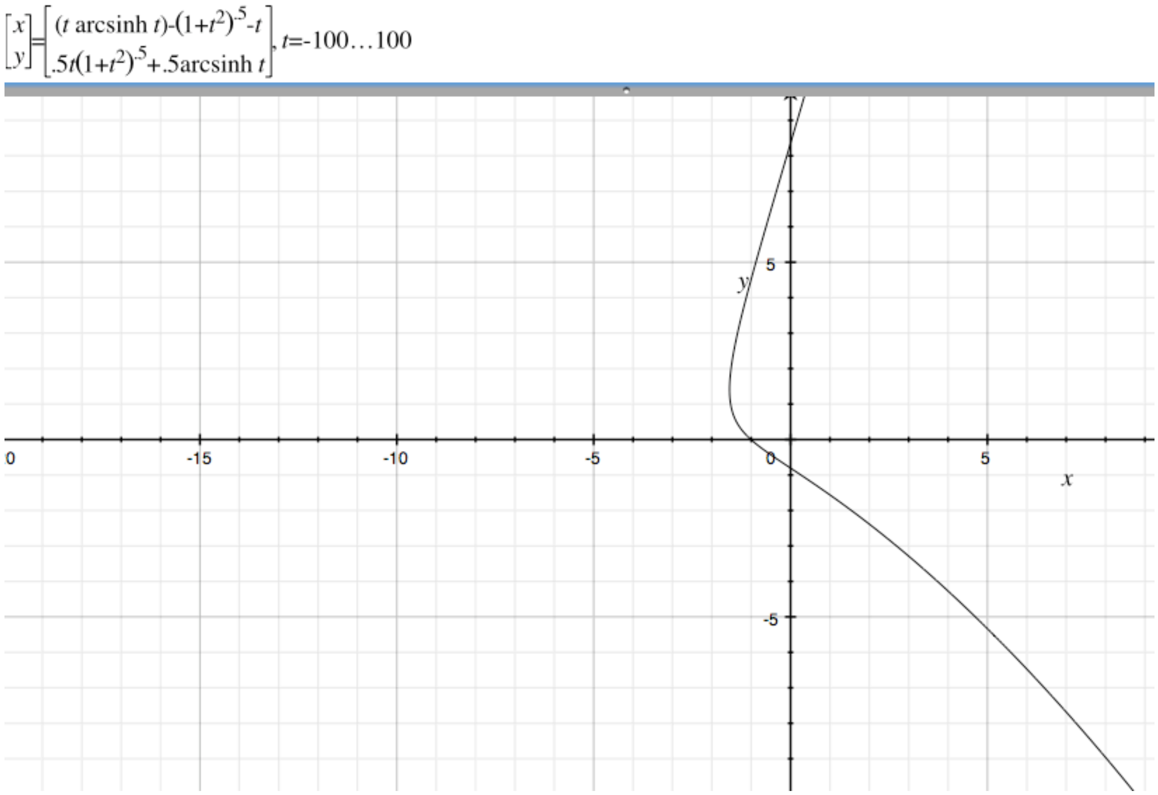


Figure 4. (A,B,C,D) Some critical paths with acceleration $At+B$ normalized.

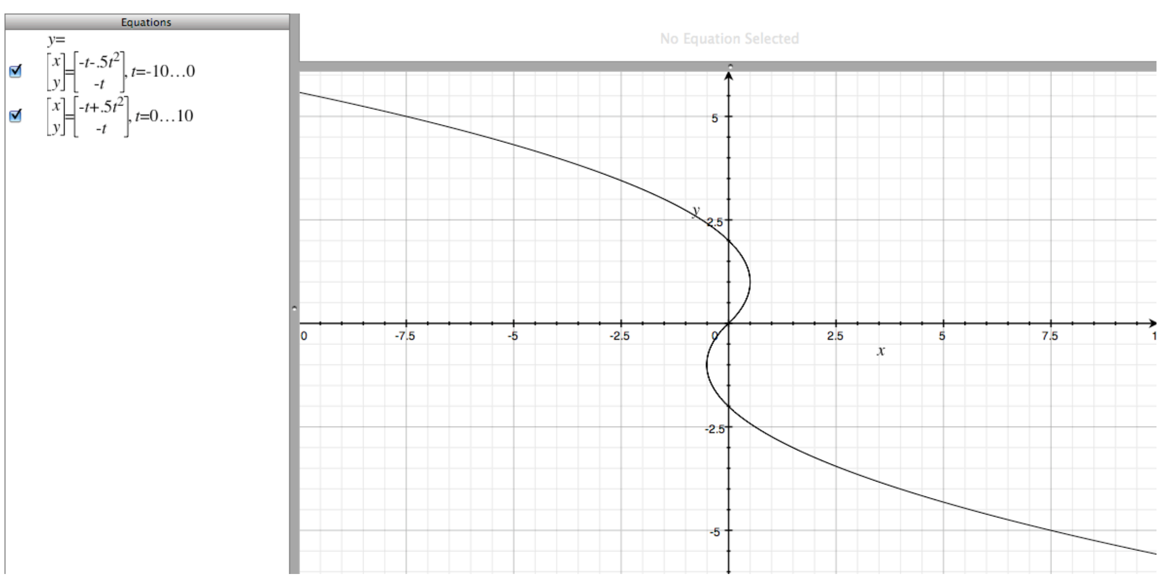


Figure 5. A symmetric critical path with acceleration $\pm(0,1)$, which is the special case $A = (0,1)$, $B = 0$.

Some such critical paths are not minimizing. Indeed, the translation in velocity space of a minimizer need not be minimizing. For example, for $\sigma = 1$, the following path P is minimizing, but its translation P' by $v_0 = (1, 0)$ is not. The path P starts at (0, 0), accelerates left for 1 second to (–1/2, 0), decelerates for 1 second to (–1, 0), and then accelerates to the right for 2 seconds, ending up at (1, 0) with velocity (2, 0). Its translation P' starts at (0, 0) with velocity (1, 0), decelerates for 1 second to (1/2, 0) and velocity (0, 0), and then accelerates for 3 seconds ending up at (5, 0) with velocity (3, 0), for a total time of 4 seconds. A minimizer P'' accelerates for $\sqrt{10} - 1$ seconds and then decelerates for $\sqrt{10} - 3$ seconds for a total time of $2\sqrt{10} - 4 \approx 2.32$ seconds. In summary, the translation P' of a backtracking minimizer P may decelerate unnecessarily and fail to be minimizing.

Note that up to translation, rotation, and scaling, the path in velocity space is the famous catenary $v = \cosh u$ or in the degenerate case a line. It is well known that such paths minimize energy $\int v\, dt = \Delta y$ for given length $\sigma \Delta t$.

There are relative minima which are not absolute minima. Consider given velocities (–1, 1), (1, 1) and change in position $(0, \Delta y)$ vertical. Possible paths in velocity space are catenaries (or horizontal lines), a 1-parameter family. A horizontal line yields minimum time, but a catenary $v = a \cosh (u/a)$ yields local minimum Δy. Rotating such a catenary about the u-axis generates the famous minimal catenoid surface, with area 2π times the potential energy $\int v\, dt = \Delta y$ of the catenary. It is well known that for two relatively close congruent vertical circles about the same horizontal axis there are two catenoids, a slightly bowed area minimum and a deeply bowed unstable one [N]. Bowing upward from the catenary generator of the stable catenoid, Δy increases; time decreases to the horizontal line, then increases. Downward, time increases; Δy decreases to the generator of the stable catenoid, then increases to the generator of the unstable catenoid, then decreases, eventually going very negative. So Δy values between the generators of the two catenoids are obtained three times, with times $T_1 < T_2 < T_3$. All have the same σ. The first is the global minimum. The third is a local minimum, since by the energy-minimizing property of the catenary, decreasing time requires increasing Δy.

All three paths minimize $max\ |a|$ for given time, because if you could reduce $max\ |a|$ for given time, you could rescale to reduce time and Δy in the same proportion instead, but for reduced time, the minimum Δy is the catenary in velocity space, for which Δy (the potential energy of the catenary) is reduced less than proportionately, because the average value of velocity increases.

Proof of Lemma 2. For variable position $x(t)$ in $C^{1,1}$ and variable acceleration $a(t)$ in L^∞, we want to minimize

$$\int_0^T dt$$

subject to the constraints $\ddot{x} = a$ (a.e.) and $|a| \leq \sigma$. Since T is smooth in a, for some Lagrange multiplier $\lambda(t)$, a minimizer is a critical point for $\int H\, dt$ where

$$H = 1 + \lambda \bullet (a - \ddot{x})\,.$$

The Euler conditions of vanishing first variation (see e.g. [M, 29.2]) say first that weakly

$$0 = \frac{\partial H}{\partial x} - \frac{d}{dt}\frac{\partial H}{\partial \dot{x}} + \frac{d^2}{dt^2}\frac{\partial H}{\partial \ddot{x}} = 0 - 0 - \ddot{\lambda}\,,$$

so that $-\lambda = At + B$ for constant vectors A, B and second that $0 \leq \partial H/\partial a$. Since a is constrained to lie in the disc of radius σ, this second condition just says that

$$a = -\sigma\lambda/|\lambda| = \sigma(At+B)/|At+B|\,.$$

Suppose that there were two fastest paths $x_1(t)$, $x_2(t)$. Then their average $x_3(t)$ would also be a fastest path. Since the acceleration $a_3(t)$ must like a_1 and a_2 have constant length σ, $a_1 = a_2$ and $x_1 = x_2$.

Proposition 1. *Given $\sigma > 0$ and points $x_1, x_2, \ldots, x_n$ in R^2 and optionally velocities v_1, v_n, there is a fastest path from x_1 to x_n passing in order through $x_2, \ldots, x_{n-1}$ with initial velocity v_0, final velocity v_n, and acceleration bounded by σ. The acceleration is continuous of magnitude σ, with at most one possible exception from x_k to x_{k+1}: it may flip direction between x_k and x_{k+1} or it may change discontinuously at x_k or x_{k+1}; the former can occur only if a is otherwise constant along the segment (as it is on the last segment), the latter only if a is constant along both incident segments. (If no velocities are prescribed, we must assume that the points do not lie in order along a line, the one case in which arbitrarily small time is possible.)*

Proof. Since the set of all possible velocities at the points x_i is compact, existence follows from Lemma 1. Lemma 2 implies the asserted regularity except at the points $x_2, \ldots, x_{n-1}$. Free velocity at x_i adds a boundary term

$$\lambda \bullet \delta v \Big]_{x_i -}^{x_i +}$$

to the first variation, so that the Lagrange multiplier λ is continuous at x_i. Therefore the acceleration $a = \lambda/|\lambda|$ is continuous at x_i, unless $\lambda(x_i) = 0$, in which case a is constant on both incident segments. At x_n, $\lambda = 0$, so on the last segment $\lambda = B(t-t_n)$ (see Remarks after Lemma 2), and a is constant on the last segment, except possibly for a flip.

The fastest path (see Figure 1). Computing the fastest path proceeds in two steps. First, for prescribed velocities at two sequential bases, we use Lemma 2, a finite difference boundary value method [F, §14.2], and multidimensional Newton's method [F, §7.1] to find a solution with velocities that match the prescriptions. This problem can be highly non-linear, and requires close guesses for Newton's method to converge, which we achieved by deforming an easily computed symmetric path. Second we minimize total time over varied choices of prescribed velocities at the bases as in the proof of Proposition 1, which we achieve with a gradient descent method [F, §7.2]. Since there is no general uniqueness result for relative minima, we cannot be sure that our solution reflects a global minimum. Our MATLAB code is given in the Appendix.

Figure 6 shows the fastest path to 2nd base for a double, taking 10.4 seconds for $\sigma = 10$ ft/sec^2, as compared to 12 seconds along the baseline, coming to a full stop at 1st and 2nd base. The runner slows down a bit before rounding 1st base.

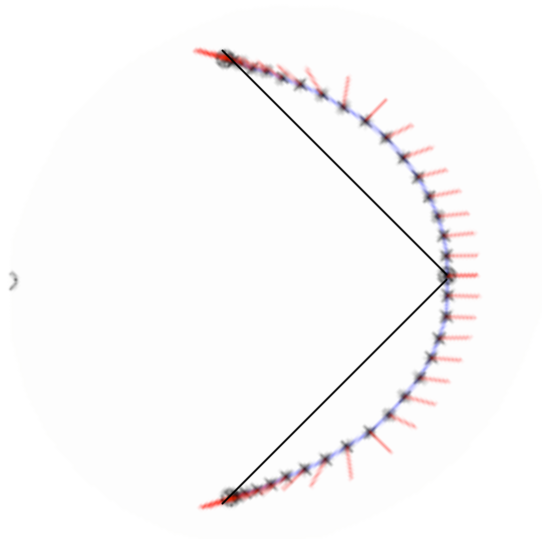


Figure 6. The fastest path to 2nd base.

Appendix

MATLAB code for computing the fastest path around the bases, with segments and acceleration both normalized to 1.

```
% compute and display optimal path around bases
% Stewart Johnson 2009



%% Main baserunner function

function baserunner()

% Set up variables
clear
global N L FLCoords1 FLCoords2 FLCoords3

% resolution
N= 120;
```

```
%ODE Matrix
M= -2*diag(ones(N-1,1))+diag(ones(N-2,1),1)+diag(ones(N-2,1),-1);
L=inv(M);

% Initial guess for velocities at bases
Varray= [0.5807 0.4725 0.6520 0.6089 0.3733 0.6152];

% Initial guess for the values of A=(a1 a2), B=(b1,1), and T
FLCoords1 =[ 0.6349 -0.4667 -0.9490 1.6870];
FLCoords2 =[-0.2672  0.9950 -1.1814 1.3642];
FLCoords3 =[ 0.5039  0.9695 -1.9015 1.4726];

% Gradient descent on total time

for iter=1:100
   iter
   del= .0001;

   Grad(1)= (TargetsToTime(Varray + [del 0 0 0 0 0])- TargetsToTime(Varray - [del 0 0
0 0 0]))/(2*del);
   Grad(2)= (TargetsToTime(Varray + [0 del 0 0 0 0])- TargetsToTime(Varray - [0 del 0
0 0 0]))/(2*del);
   Grad(3)= (TargetsToTime(Varray + [0 0 del 0 0 0])- TargetsToTime(Varray - [0 0 del
0 0 0]))/(2*del);
   Grad(4)= (TargetsToTime(Varray + [0 0 0 del 0 0])- TargetsToTime(Varray - [0 0 0
del 0 0]))/(2*del);
   Grad(5)= (TargetsToTime(Varray + [0 0 0 0 del 0])- TargetsToTime(Varray - [0 0 0 0
del 0]))/(2*del);
   Grad(6)= (TargetsToTime(Varray + [0 0 0 0 0 del])- TargetsToTime(Varray - [0 0 0 0
0 del]))/(2*del);

   Varray= Varray - .01*Grad;

end

% Display results
Report(Varray)

% Done
return


%% Velocity targets to total time function.

function TotalTime=TargetsToTime(V)
```

```
global FLCoords1 FLCoords2 FLCoords3

A1=V(1);
B1=V(2);
A2=V(3);
B2=V(4);
A3=V(5);
B3=V(6);

%%% Home to First %%%%%%%%%%%%%%%%%%%%%%
Target1= [0.0,0.0,A1,B1];
FLCoords1= MatchCoords(FLCoords1,Target1,20);
T1= FLCoords1(4);

%%% First to Second %%%%%%%%%%%%%%%%%%%
Target2= [B1,-A1,A2,B2];
FLCoords2= MatchCoords(FLCoords2,Target2,20);
T2= FLCoords2(4);

%%% Second to Home %%%%%%%%%%%%%%%%%%%%%
Target3= [B2,-A2,A3,B3];
FLCoords3= MatchCoords(FLCoords3,Target3,20);
T3= FLCoords3(4);

%%%Third to Home %%%%%%%%%%%%%%%%%%%%%%%
Phi=  B3;
Mu=  -A3;
R=roots([.25 0 -Mu^2-Phi^2 2*Phi -1]);
T4=0;
for i=1:length(R)
   r= R(i);
   if imag(r)==0 && r>0 && r<2
      T4=r;
   end
end

% Done
TotalTime= T1+T2+T3+T4;
return

%% Graphics & Report

function Report(V)

global N FLCoords1 FLCoords2 FLCoords3
```

```
figure
hold on

%%% Graph H->1 %%%%%%%%%%%%%%%%%%%%%%%%%%%
a= FLCoords1(1);
b= FLCoords1(2);
c= FLCoords1(3);
d=1.0;
%plot path
[alph1, bet1, alph2, bet2, X, Y]=abcSfun(FLCoords1);
T1= FLCoords1(4);
plot(1-X,1-Y)
%plot acceleration
T= linspace(0,T1,N+1);
D= sqrt((a*T+b).^2 + (c*T+d).^2);
F= (c*T+d)./D;
G= (a*T+b)./D;
for i=1:N+1
   plot([1-X(i),1-X(i)+.1*F(i)],[1-Y(i),1-Y(i)+.1*G(i)],'r')
end
plot(1-X,1-Y,'k*')
% compute speed
for i=1:N
   Sp(i)= sqrt( (X(i+1)-X(i))^2 + (Y(i+1)-Y(i))^2 )*N/T1;
   Tm(i)= T(i);
end

%%% Graph 1->2 %%%%%%%%%%%%%%%%%%%%%%%%%%%
a= FLCoords2(1);
b= FLCoords2(2);
c= FLCoords2(3);
d=1.0;
%plot path
[alph1, bet1, alph2, bet2, X, Y]=abcSfun(FLCoords2);
T2= FLCoords2(4);
plot(Y,1-X)
%plot acceleration
T= linspace(0,T2,N+1);
D= sqrt((a*T+b).^2 + (c*T+d).^2);
F= (c*T+d)./D;
G= (a*T+b)./D;
for i=1:N+1
   plot([Y(i),Y(i)-.1*G(i)],[1-X(i),1-X(i)+.1*F(i)],'r')
end
plot(Y,1-X,'k*')
```

```
%compute speed
for i=1:N
   Sp(i+N)= sqrt( (X(i+1)-X(i))^2 + (Y(i+1)-Y(i))^2 )*N/T2;
   Tm(i+N)= T(i) + Tm(N);
end

%%% Graph 2->3 %%%%%%%%%%%%%%%%%%%%%%%%%
a= FLCoords3(1);
b= FLCoords3(2);
c= FLCoords3(3);
d=1.0;
%plot path
[alph1, bet1, alph2, bet2, X, Y]=abcSfun(FLCoords3);
T3= FLCoords3(4);
plot(X,Y)
%plot acceleration
T= linspace(0,T3,N+1);
D= sqrt((a*T+b).^2 + (c*T+d).^2);
F= (c*T+d)./D;
G= (a*T+b)./D;
for i=1:N+1
   plot([X(i),X(i)-.1*F(i)],[Y(i),Y(i)-.1*G(i)],'r')
end
plot(X,Y,'k*')
%compute speed
for i=1:N
   Sp(i+2*N)= sqrt( (X(i+1)-X(i))^2 + (Y(i+1)-Y(i))^2 )*N/T3;
   Tm(i+2*N)= T(i) + Tm(2*N);
end

%%% Graph 3->h %%%%%%%%%%%%%%%%%%%%%%%%%
Phi=  V(6);
Mu=  -V(5);
R=roots([.25 0 -Mu^2-Phi^2 2*Phi -1]);
T4=0;
for i=1:length(R)
   r= R(i);
   if imag(r)==0 && r>0 && r<2
      T4=r;
   end
end
%plot path
T= linspace(0,T4,N+1);
X= (T/T4).^2*(-Phi*T4+1)+Phi*T;
Y= (T/T4).^2*(-Mu*T4)   +Mu*T;
plot(1-Y,X)
```

```
plot(1-Y,X,'k*')
%plot acceleration
d1= 2*(-Phi*T4+1)/T4^2;
d2= 2*(-Mu*T4)/T4^2;
for i=1:N+1
   plot([1-Y(i),1-Y(i)+.1*d2],[X(i),X(i)-.1*d1],'m')
end
%compute speed
for i=1:N
   Sp(i+3*N)= sqrt( (X(i+1)-X(i))^2 + (Y(i+1)-Y(i))^2 )*N/T4;
   Tm(i+3*N)= T(i) + Tm(3*N);
end

%%% Plot bases & preen %%%%%%%%%%%%%%%%%%
plot(1,0,'ko');
plot(1,1,'ko');
plot(0,0,'ko');
plot(0,1,'ko');
axis([-.6 1.6 -.6 1.6]);
axis equal
hold off

%%% Plot speed graph %%%%%%%%%%%%%%%%%%%%
figure
hold on
plot(Tm,Sp);
plot([Tm(N) Tm(2*N) Tm(3*N)],[Sp(N) Sp(2*N) Sp(3*N)],'bo');
hold off

% Done
return

%% abcSfun() for boundary value problem
% Given At+B and time S, solve ODE for path from (0,0) to (1,0),
% and report start and end velocities

function [alph1, bet1, alph2, bet2, X, Y ]= abcSfun(FLC)

global N L

a= FLC(1);
b= FLC(2);
c= FLC(3);
d=1.0;

S= FLC(4);
```

```
T= linspace(0,S,N+1);
D= sqrt((a*T+b).^2 + (c*T+d).^2);

F= (S/N)^2*(c*T+d)./D;
F(N)= F(N)-1;

G= (S/N)^2*(a*T+b)./D;

X(1)= 0;
X(N+1)= 1;
X(2:N)= L * F(2:N)';

Y(1)= 0;
Y(N+1)= 0;
Y(2:N)= L * G(2:N)';

alph1= (X(2)-X(1))*N/S;
alph2= (X(N+1)-X(N))*N/S;

bet1= (Y(2)-Y(1))*N/S;
bet2= (Y(N+1)-Y(N))*N/S;

return

%% Compute jacobian for abcSfun()

function J= abcSjako(FLC)

del= .1;

[alph1a bet1a alph2a bet2a]= abcSfun(FLC-[del 0 0 0]);
[alph1c bet1c alph2c bet2c]= abcSfun(FLC+[del 0 0 0]);
J(1:4,1)=([alph1c bet1c alph2c bet2c]-[alph1a bet1a alph2a bet2a])/(2*del);

[alph1a bet1a alph2a bet2a]= abcSfun(FLC-[0 del 0 0]);
[alph1c bet1c alph2c bet2c]= abcSfun(FLC+[0 del 0 0]);
J(1:4,2)=([alph1c bet1c alph2c bet2c]-[alph1a bet1a alph2a bet2a])/(2*del);

[alph1a bet1a alph2a bet2a]= abcSfun(FLC-[0 0 del 0]);
[alph1c bet1c alph2c bet2c]= abcSfun(FLC+[0 0 del 0]);
J(1:4,3)=([alph1c bet1c alph2c bet2c]-[alph1a bet1a alph2a bet2a])/(2*del);

[alph1a bet1a alph2a bet2a]= abcSfun(FLC-[0 0 0 del]);
[alph1c bet1c alph2c bet2c]= abcSfun(FLC+[0 0 0 del]);
J(1:4,4)=([alph1c bet1c alph2c bet2c]-[alph1a bet1a alph2a bet2a])/(2*del);
```

```
return

%% Newtons method for matching FLCoords to velocities
% Highly nonlinear problem. Need close guess.

function FLCoordsMatch= MatchCoords(FLCoords,Target,iter)

for i=1:iter
   [alph1, bet1, alph2, bet2]=abcSfun(FLCoords);
   Del= abcSjako(FLCoords)\[Target(1)-alph1; Target(2)-bet1; Target(3)-alph2;
Target(4)-bet2];
   FLCoords = FLCoords + Del';
end

FLCoordsMatch= FLCoords;

return
```

References

[F] Laurene V. Fausett, *Applied Numerical Analysis using MATLAB,* 2nd ed, Prentice-Hall, 2008.

[G] Guiness World Records, http://www.baseball-almanac.com/recbooks/rb_guin.shtml

[M] Frank Morgan, *Real Analysis with Applications,* Amer. Math. Soc., 2008.

[S] http://speedendurance.com/2009/08/19/usain-bolt-10-meter-splits-fastest-top-speed-2008-vs-2009/

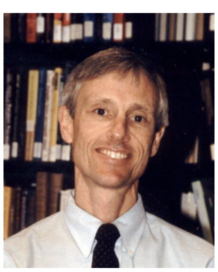
[TF] Dao Trong Thi and A. T. Fomenko, *Minimal Surfaces, Stratified Multivarifolds, and the Plateau Problem,* Amer. Math. Soc., 1991.

DAVIDE CAROZZA graduated in 2009 form Williams College, where he presented his senior colloquium (a requirement for all senior math majors) on some preliminary analysis of optimal base-running. In his present job he teaches Algebra I.

St. Albans School
Washington, DC 20016
09dgc@williams.edu

STEWART JOHNSON is a professor working in dynamics, optimal control, mathematical biology, game theory, differential equatIons, and statistics. He is also the Quantitative Skills Coordinator for Williams College, and as such he give tutorials and course for developing quantitative skills.

Department of Mathematics and Statistics
Williams College
Williamstown, MA 01267
Stewart.Johnson@williams.edu
http://williams.edu/Mathematics/sjohnson/



FRANK MORGAN is a specialist in minimal surfaces. One of his books is “The Math Chat Book”, based on the call-in TV show he used to run and his column at MathCaht.org. He is founder of Williams’s “SMALL” Undergraduate Research Project.

As Vice-President of the American Mathematical Society, he launched the blog http://mathgradblog.williams.edu by and for mathematics graduate students.

Department of Mathematics and Statistics
Williams College
Williamstown, MA 01267
Frank.Morgan@williams.edu
http://math.williams.edu/morgan