Automatic Step-Size Control for Minimization Iterations

Prof. W. Kahan (Retired)
University of California
Berkeley CA., USA

Abstract:

The "Training" of "Deep Learning" for "Artificial Intelligence" is a process that minimizes a "Loss Function" $f(\mathbf{w})$ subject to memoryconstraints that allow the computation of $f(\mathbf{w})$ and Gradients $\mathbf{G}(\mathbf{w}) := \mathrm{d}f(\mathbf{w})/\mathrm{d}\mathbf{w}$` but not the Hessian $\mathrm{d}^2 f(\mathbf{w})/\mathrm{d}\mathbf{w}^2$ nor estimates of it from many stored pairs $\{\mathbf{G}(\mathbf{w}), \mathbf{w}\}$. Therefore the process is iterative using "Gradient Descent" or an accelerated modification of it like "Gradient Descent Plus Momentum". These iterations require choices of one or two scalar "Hyper-Parameters" which cause divergence if chosen badly. Fastest convergence requires choices derived from two of the Hessian's attributes, its "Norm" and "Condition Number", that can almost never be known in advance. This retards Training,– severely if the Condition Number is big. A new scheme chooses Gradient Descent's Hyper-Parameter, a step-size called "the Learning Rate", automatically without any prior information about the Hessian; and yet that scheme has been observed always to converge *ultimately* almost as fast as could any accelerated version of Gradient Descent with optimally chosen Hyper-Parameters. Alas, a mathematical proof of that new scheme's efficacy has not been found yet.

This document is posted at people.eecs.berkeley.edu/~wkahan/26Sept19.pdf
Details are posted at people.eecs.berkeley.edu/~wkahan/STEPSIZE.pdf

Automatic Step-Size Control for Minimization Iterations

In a space of high dimension, we seek vectors **x** to minimize a "smooth" scalar function $f(\mathbf{x})$, We may have to find more than one minimum to select one we prefer.

Memory limitations restrict our choices of algorithms.

Their speeds depend upon "**H**yper-**P**arameters" we must choose.

Two Regimes -- Two Strategies

Ignorance obstructs our ability to choose **H-P**s well..

How fast could agorithms go with the (unknown) best **H-P**s ?

How can our programs go almost that fast despite our ignorance?

"... if a machine is expected to be infallible,
it cannot also be intelligent."

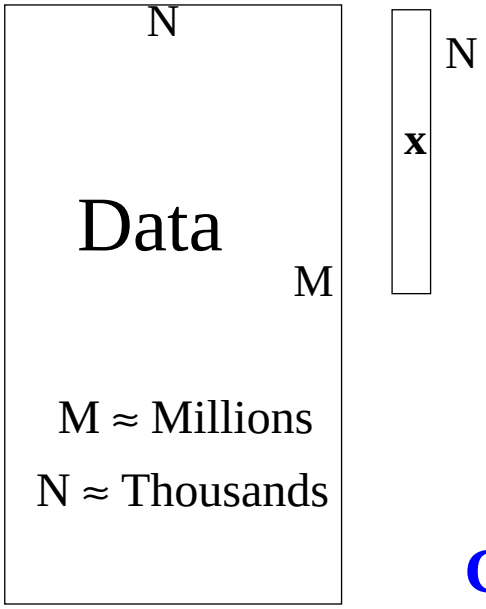
A.M. Turing, 1947

Intelligence entails curiosity, adaptability to unpredictable and predictable changes, and an ability to learn from mistakes *i.a.*

"Deep Learning" does not deserve to be called "Learning"
unless it can learn also from mistakes.
This entails REtraining; it must not be too slow.

How fast can (Re)Training algorithms be made to go?

Memory limitations restrict our choices of algorithms.



We seek **x** to minimize $f(\mathbf{x})$.

Gradient $\mathbf{G}(\mathbf{x}) := \partial f(\mathbf{x})/\partial \mathbf{x}'$; a few are computable: M·N work, N storage.

Hessian $\mathbf{H}(\mathbf{x}) := \partial^2 f(\mathbf{x})/\partial \mathbf{x}^2$ is NOT computable; $M \cdot N^2$ work, $N^2/2$ storage

Our chosen algorithm must come from ...

Gradient-Based Iterations

e.g., **GD**: Gradient Descent:

new $\mathbf{x} := \mathbf{x} - \Delta\tau \cdot \mathbf{G}(\mathbf{x})$

(AI calls Step-Size $\Delta\tau$ "the Learning Rate")

GD+M: Gradient Descent + Momentum:

new $\mathbf{x} := \mathbf{x} - \alpha \cdot \mathbf{G}(\mathbf{x}) + \beta \cdot (\mathbf{x} - \text{old } \mathbf{x})$

and variations.

They all require choices for scalar **H**yper-**P**arameters $\Delta\tau$, α, β.

How should these be chosen ?

Gradient-Based Iterations can be construed as **Discretizations** of **Differential Equations:**

GD: $d\mathbf{x}(\tau)/d\tau = -\mathbf{G}(\mathbf{x}(\tau)) = -\partial f(\mathbf{x}(\tau))/\partial \mathbf{x}'$.

f declines along each Trajectory $\mathbf{x}$ satisfying the Differential Equation:

$$df(\mathbf{x}(\tau))/d\tau = -\|\mathbf{G}(\mathbf{x}(\tau))\|^2 < 0 \quad \text{so long as} \quad \mathbf{G} \neq \mathbf{o} .$$

Each Trajectory terminates at a Stationary Point $\mathbb{x}$ where $\mathbf{G}(\mathbb{x}) = \mathbf{o}$.
$\mathbb{x}$ is almost always a (local) Minimum of f .

GD+M: $d\mathbf{x}(\tau)/d\tau = \mathbf{v}(\tau)$;

$d\mathbf{v}(\tau)/d\tau = -\mathbf{G}(\mathbf{x}(\tau)) - \mu\cdot\mathbf{v}(\tau)$ for some ***Drag*** $\mu > 0$.

Associated with these Differential Equations is a ***Pseudo-Hamiltonian***

$$Æ(\mathbf{x}, \mathbf{v}) := f(\mathbf{x}) + \|\mathbf{v}\|^2/2 \qquad \text{(analogous to } \textit{Total Energy}\text{)}$$

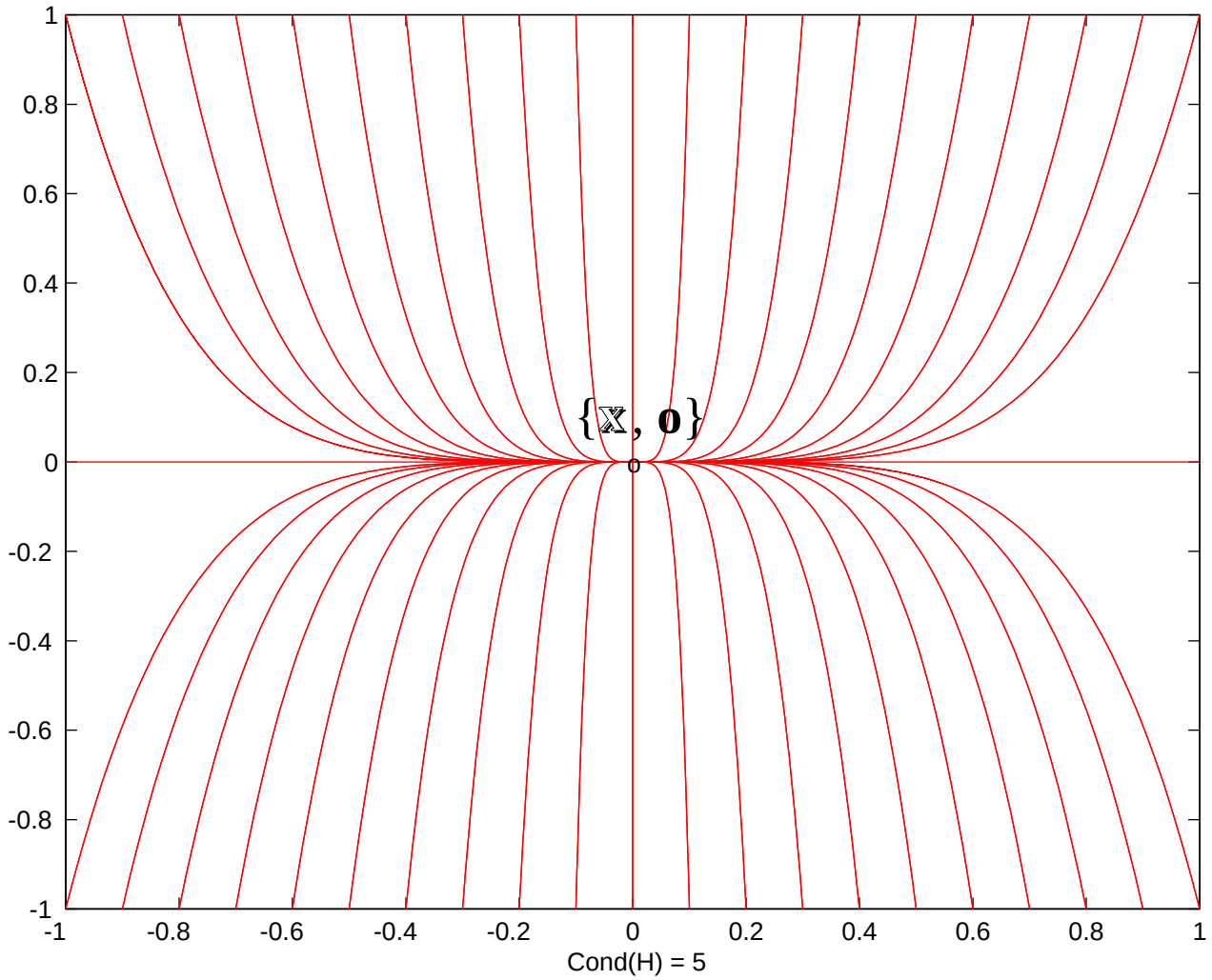
$Æ(\mathbf{x}, \mathbf{v})$ declines along each of the Differential Equations' Trajectories:

$$dÆ(\mathbf{x}(\tau), \mathbf{v}(\tau))/d\tau = -\mu\cdot\|\mathbf{v}(\tau)\|^2 < 0 \quad \text{so long as} \quad \mathbf{v}(\tau) \neq \mathrm{o} .$$

Each Trajectory ends at a Stationary Point $\{\mathbb{x}, \mathbb{v}\}$ where $\mathbf{G}(\mathbb{x}) = \mathbb{v} = \mathbf{o}$.
$\{\mathbb{x}, \mathbb{v}\}$ is almost always a (local) Minimum of f and of $Æ$.

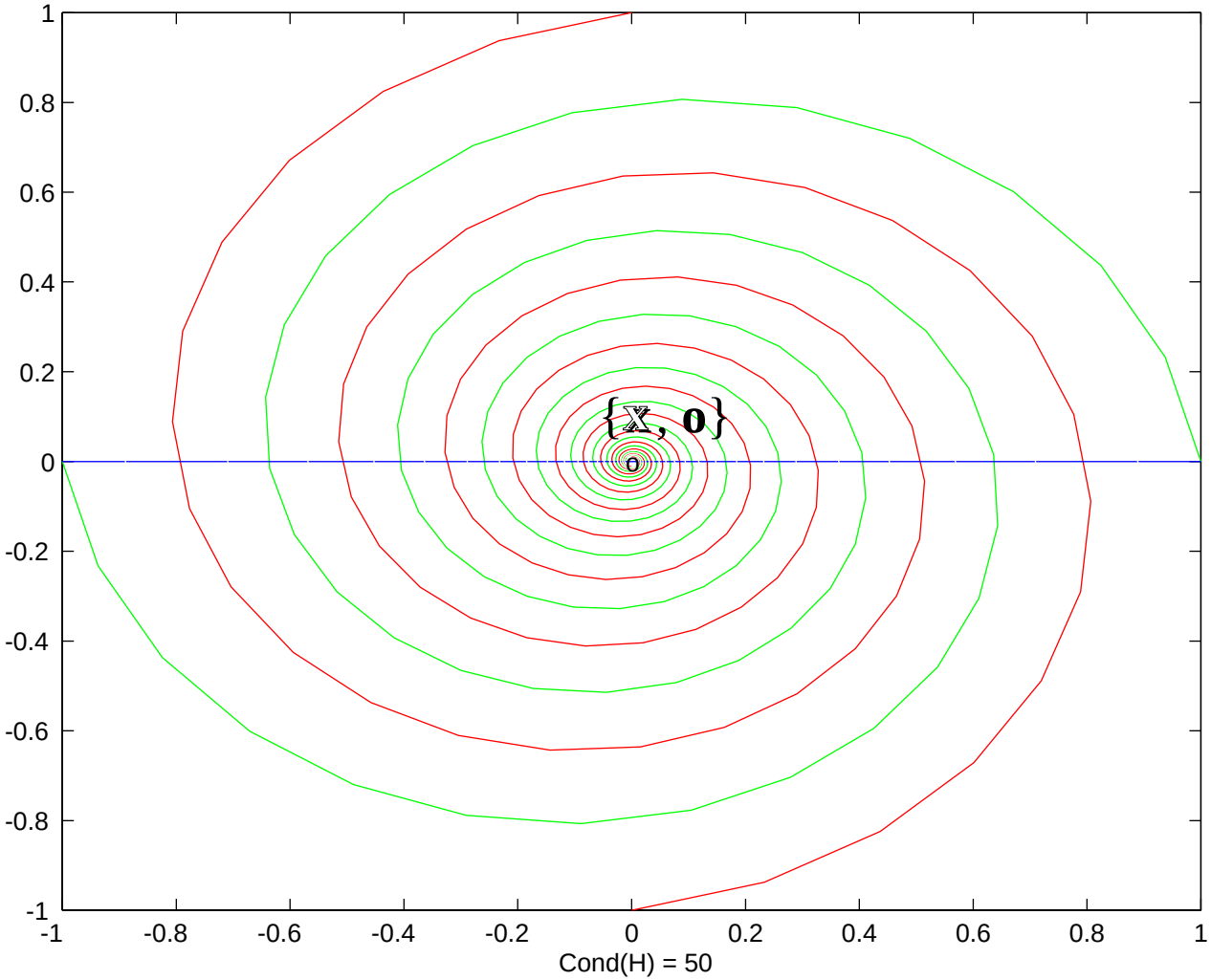
Trajectories fill the space. What do they look like?

Near a Minimum, for **GD**, and for **GD+M** with high Drag μ



Visualize this in a space of high dimension N or 2N ; twisted "*L*"

Near a Minimum, for GD+M with low Drag μ



Visualize this in a space of high dimension 2N ; Corkscrews & twisted *L*

Two Regimes -- Two Strategies

Regime #0: Iterates **x** are not near enough to a Minimum of f ;
Curvature ($\mathbf{H}(\mathbf{x})$) varies -- Hyperbolic, Ellipsoidal.
Hyper-Parameters influence iterations unpredictably.

Strategy #0: Choose Step-Sizes $\Delta\tau$ (Learning Rates) small enough that computed iterates **x** follow a trajectory closely enough that they approach a minimum $\mathbb{X}$ determined by the initial **x** rather than by Hyper-Parameters or computational accidents.

If f has more than one minimum, some preferable to others, prudence obliges us to seek them by starting searches from several initial points.

Even if initial points are distributed partly randomly,
bigger step-sizes $\Delta\tau$ may increase the likelihood that
some minima $\mathbb{X}$ get rediscovered repeatedly
while others go unnoticed.

Two Regimes -- Two Strategies

Regime #1: Iterates **x** are near enough to a Minimum $\mathbb{x}$ that the Curvature ($\mathbf{H}(\mathbf{x})$) is Ellipsoidal and varies relatively little. Hyper-Parameters determine iterations' *ultimate* convergence rate from two attributes of $\mathbf{H}(\mathbb{x})$:

its Norm $\|\mathbf{H}\|$ and its Condition Number $Ç := \|\mathbf{H}\|\cdot\|\mathbf{H}^{-1}\|$.

Strategy #1: When $Ç >> 1$ and Hyper-Parameters are optimal, step-sizes $\Delta\tau$ are big enough that iterates ***Ricochet*** wildly while converging as fast as possible. Their behavior is practically indistinguishable from substantially slower convergence or divergence due to slightly sub-optimal Hyper-Parameters.

But $\|\mathbf{H}\|$ and $Ç$ can almost never be known in advance.

How can good Hyper-Parameters be chosen despite our ignorance?

How good is good enough?

How fast could agorithms go with the (unknown) best **H-P**s ?

Because our iterates can Ricochet, we measure an iteration's *ultimate* rate of convergence $-\log(\rho)$ by finding the ratio ρ^n by which a *very* big number n of iterations reduces $\|\mathbf{G}(\mathbf{x})\|$ in Regime #1. In other words,
the convergence ratio ρ is the ***average*** factor by which
each of vastly many iteration-steps ("*Epochs*") reduces $\|\mathbf{G}(\mathbf{x})\|$
starting from almost every initial iterate **x** in Regime #1.

A 1960s theorem by V. Samanskii says
every Gradient-Based Iteration (all our iterations are among them)
must have a convergence ratio
$$\rho \geq 1 - 2/(1 + \sqrt{Ç})$$
no matter how the iteration's Hyper-Parameters were chosen.
And that inequality can becomes equality for
some Gradient-Based Iterations in Regime #1.

In particular, $\rho = 1 - 2/(1 + \sqrt{Ç})$ for **Optimally Chosen**
constant Hyper-Parameters for
some versions of Gradient Descent Plus Momentum
running in Regime #1.

Two Versions of Gradient Descent Plus Momentum:

GD+M: new $\mathbf{v} := \mathbf{v} - \Delta\tau\cdot(\mathbf{G}(\mathbf{x}) + \mu\cdot\mathbf{v})$; new $\mathbf{x} := \mathbf{x} + \Delta\tau\cdot(\text{new } \mathbf{v})$;

i.e. new $\mathbf{x} := \mathbf{x} - \alpha\cdot\mathbf{G}(\mathbf{x}) + \beta\cdot(\mathbf{x} - \text{old } \mathbf{x})$;

when $\alpha \equiv \Delta\tau^2$, $\beta \equiv 1 - \mu\cdot\Delta\tau$, and $\mathbf{x} - \text{old } \mathbf{x} \equiv \Delta\tau\cdot\mathbf{v}$.

Converges in Regime #1 if & only if $\|\mathbf{H}\|\cdot\alpha < 2 + 2\beta < 4$.

Optimal choices: $\alpha = (4/\|\mathbf{H}\|)/(1 + 1/\sqrt{Ç})^2$, $\beta = ((\sqrt{Ç}-1)/(1+\sqrt{Ç}))^2$.

Best Convergence Ratio: $\rho = 1 - 2/(1 + \sqrt{Ç}) = \sqrt{\beta}$ is Samanskii's best.

AGD: Anadromic Gradient Descent:

$\mathbf{y} := \mathbf{x} + \mathbf{v}\cdot\Delta\tau/2$;

new $\mathbf{v} := \mathbf{v} - (\mathbf{G}(\mathbf{y}) + \mu\cdot\mathbf{v})\cdot\Delta\tau/(1 + \mu\cdot\Delta\tau/2)$;

new $\mathbf{x} := \mathbf{y} + (\text{new } \mathbf{v})\cdot\Delta\tau/2$.

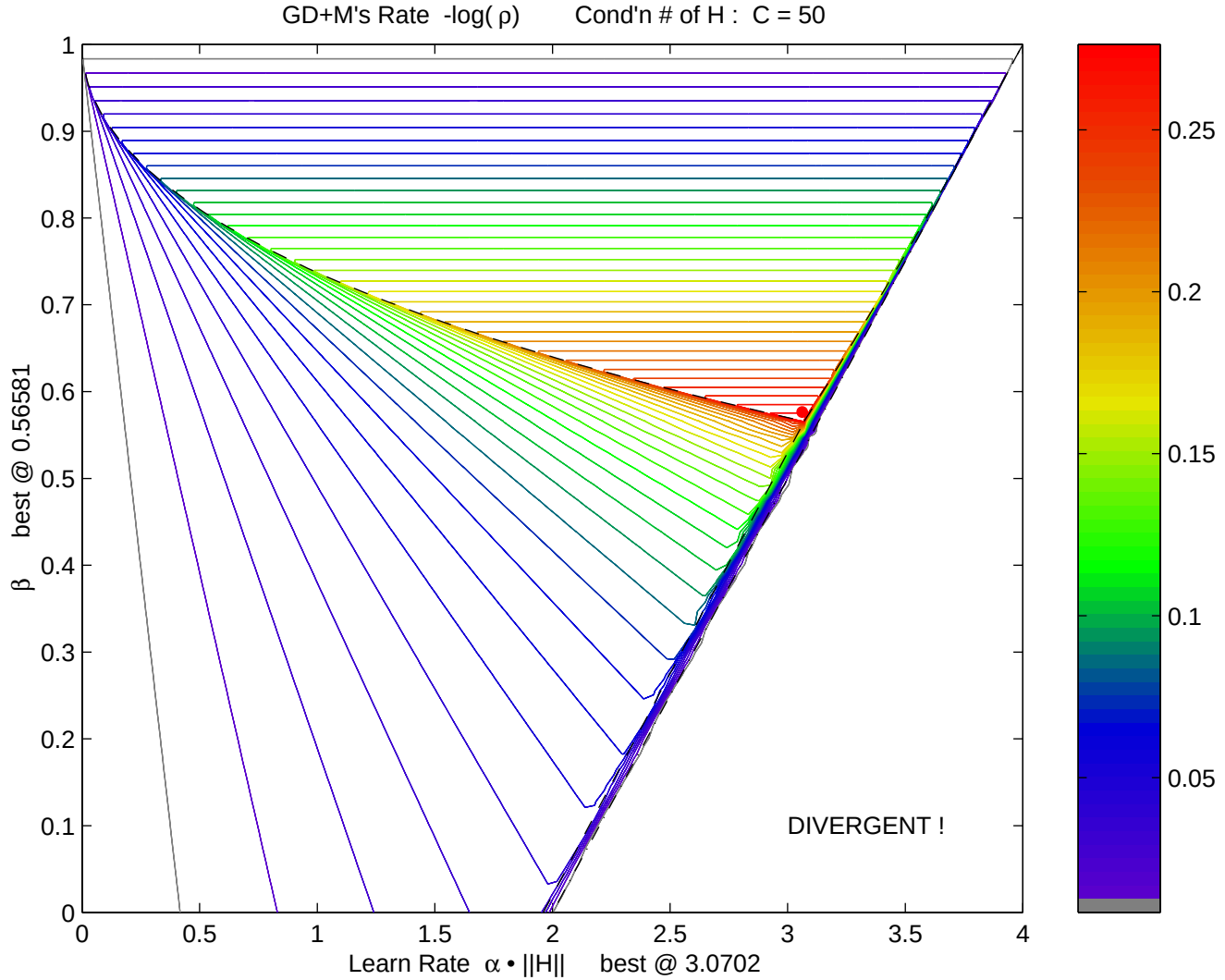
Converges in Regime #1 if & only if $\|\mathbf{H}\|\cdot\Delta\tau^2 < 4$.

Optimal choices: $\Delta\tau = (2/\sqrt{\|\mathbf{H}\|})/\sqrt{1 + 1/Ç}$, $\mu\cdot\Delta\tau = 4\sqrt{Ç}/(1 + Ç)$.

Best Convergence Ratio: $\rho = 1 - 2/(1 + \sqrt{Ç})$ is Samanskii's best.

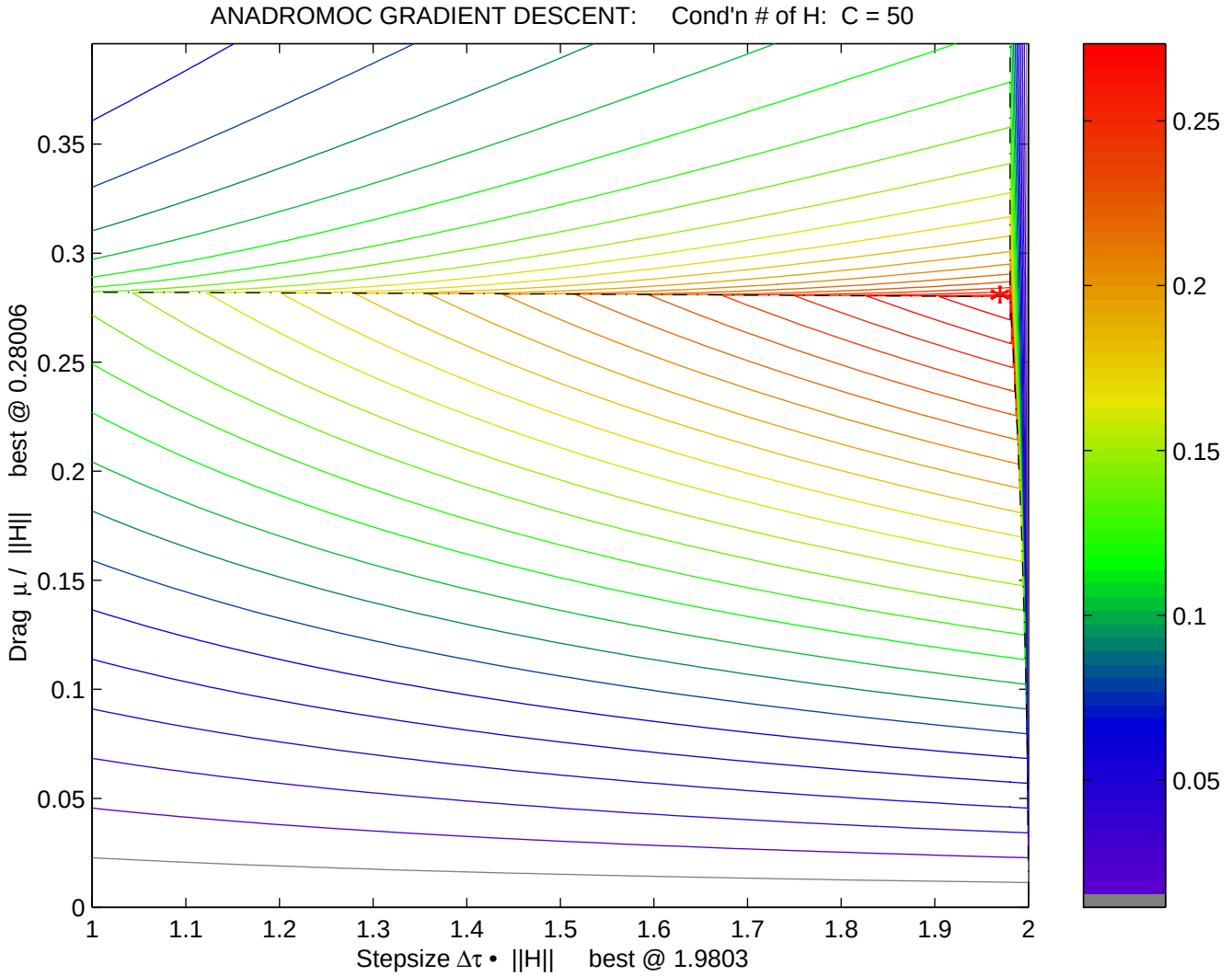
AGD tolerates bigger step-sizes $\Delta\tau$ than **GD+M** and *any* Drag $\mu > 0$.

GD+M's Convergence Rate –log(ρ) for **H**'s Condition No. = 50



The Rate plummets for slightly sub-optimal learning rate α or β .
Divergence occurs if α is slightly too big or β slightly too small..

AGD's Convergence Rate $-\log(\rho)$ for **H**'s Condition No. = 50



Sub-optimal step-sizes $\Delta\tau$ and drag μ are tolerated well; but the Rate plummets if μ is slightly too big. $\Delta\tau$ slightly too big causes divergence.

But $\|\mathbf{H}\|$ and Ç can almost never be known in advance.

Does this put Samanskii's best ρ beyond reach? Maybe not:

Ordinary un-accelerated **G**radient **D**escent's

$$\text{new } \mathbf{x} := \mathbf{x} - \Delta\tau\cdot\mathbf{G}(\mathbf{x})$$

changes $f(\mathbf{x})$ to

$$f(\text{new } \mathbf{x}) \approx f(\mathbf{x}) - \Delta\tau\cdot\|\mathbf{G}(\mathbf{x})\|^2 + \Delta\tau^2\cdot\mathbf{G}(\mathbf{x})'\bullet\mathbf{H}(\mathbf{x})\bullet\mathbf{G}(\mathbf{x})/2 \pm \boldsymbol{O}(\Delta\tau\cdot\|\mathbf{G}(\mathbf{x})\|)^3$$

whose last term "$\pm\boldsymbol{O}$..." becomes relatively negligible as iterate $\mathbf{x}$ goes deeper into Regime #1 approaching $\mathbb{X}$ where $\mathbf{G}(\mathbb{X}) = \mathbf{o}$.

To minimize $f(\text{new } \mathbf{x})$ (while ignoring "$\pm\boldsymbol{O}$...") we would have chosen a step-size

$$\Delta\tau/\big(\,2 + 2(f(\text{new } \mathbf{x}) - f(\mathbf{x}))/(\Delta\tau\cdot\|\mathbf{G}(\mathbf{x})\|^2)\,\big)$$

$$\approx \mathbf{G}(\mathbf{x})'\bullet\mathbf{G}(\mathbf{x})/\big(\,\mathbf{G}(\mathbf{x})'\bullet\mathbf{H}(\mathbf{x})\bullet\mathbf{G}(\mathbf{x}) \pm \boldsymbol{O}(\Delta\tau\cdot\|\mathbf{G}(\mathbf{x})\|)^3\,\big)$$

had we known what we know now.

Too late now; but better late than never.

When iterate $\mathbf{x}$ is deep in Regime #1, after executing a **GD**-step

$$\text{new } \mathbf{x} := \mathbf{x} - \Delta\tau\cdot\mathbf{G}(\mathbf{x}) \ ,$$

let us compute

$$\text{new } \Delta\tau := \Delta\tau/\max\{\,0.25,\ \ 2 + 2(f(\text{new } \mathbf{x}) - f(\mathbf{x}))/(\Delta\tau\cdot\|\mathbf{G}(\mathbf{x})\|^2)\,\}$$

to be used as the step-size for the next **GD**-step. The "max{ 0.25, ...}" prevents new $\Delta\tau$ from going wild because of a computational accident like roundoff.

But if consecutively new $\Delta\tau = 4\Delta\tau$ much too often, suspicions should arise that either Ç is enormous or $\mathbf{x}$ is not deep in Regime #1 after all.

Computed from the formula above, step-sizes new $\Delta\tau$ can fluctuate chaotically between $1/\|\mathbf{H}\|$ and $Ç/\|\mathbf{H}\|$; and then this **GD** iteration has been observed to converge rapidly as if its ultimate average convergence ratio were nearly Samanskii's best $\boldsymbol{\rho} = 1 - 2/(1 + \sqrt{Ç})$.

Why that scheme above works so well has not yet been explained to my mathematical satisfaction, partly because it doesn't always work so well.

That scheme has a failure mode.

When iterate $\mathbf{x}$ is deep in Regime #1, after executing a **GD**-step

$$\text{new } \mathbf{x} := \mathbf{x} - \Delta\tau\cdot\mathbf{G}(\mathbf{x}) ,$$

we compute

$$\text{new } \Delta\tau := \Delta\tau/\max\{\,0.25,\;\; 2 + 2(f(\text{new } \mathbf{x}) - f(\mathbf{x}))/(\Delta\tau\cdot\|\mathbf{G}(\mathbf{x})\|^2)\,\}$$

as the step-size for the next **GD**-step except occasionally, at random,

$$\text{new } \Delta\tau := \Delta\tau/2 .$$

This thwarts a failure mode when, for peculiar initial $\mathbf{x}$ and $\Delta\tau$, every new $\Delta\tau \approx (2/\|\mathbf{H}\|)/(1 + Ç)$ and the *ultimate* average convergence ratio is $1 - 2/(1 + Ç)$, much worse than Samanskii's best $\rho = 1 - 2/(1 + \sqrt{Ç})$ when $Ç >> 1$. The failure mode resembles the scheme's behavior when $Ç$ exceeds 1 only a little, in which case **GD** converges rapidly; but we wish not to iterate long enough to measure **GD**'s *ultimate* rate.

Occasionally computing new $\Delta\tau := \Delta\tau/2$ won't change the rate much.

Does the scheme above have another failure mode?

I have failed to find one after trying many examples.

The scheme always converged as fast as I expected.

Escaping from Regime #0

Though we seek an $\mathbb{x}$ where $\mathbf{G}(\mathbb{x}) = \mathbf{o}$, Gradient-Based Iterations do not always reduce $\|\mathbf{G}\|$, especially in Regime #0 where the curvature ($\mathbf{H}(\mathbf{x})$) can be hyperbolic. Instead we can gauge an iteration's progress by how much its every step diminishes ...

$f(\mathbf{x})$ by **GD**,
Æ$(\mathbf{x}, \mathbf{v})$ by **GD+M** and **AGD**.

If $\Delta\tau$ is small enough, each **GD**-step new $\mathbf{x} := \mathbf{x} - \Delta\tau\cdot\mathbf{G}(\mathbf{x})$ gets ...

$$f(\text{new } \mathbf{x}) \approx f(\mathbf{x}) - \Delta\tau\cdot(4\|\mathbf{G}(\mathbf{x})\|^2 + \|\mathbf{G}(\mathbf{x}) + \mathbf{G}(\text{new } \mathbf{x})\|^2)/8 + \boldsymbol{O}(\Delta\tau^3)$$

in which the coefficient of $\Delta\tau$ is the rate at which $f(\text{new } \mathbf{x})$ decreases for very small step-sizes. *It's not obvious*. $f(\text{new } \mathbf{x})$ and $\mathbf{G}(\text{new } \mathbf{x})$ will be computed in the course of preparing for the next step. But the next step will not be taken if $f(\text{new } \mathbf{x}) \geq f(\mathbf{x})$. Instead that new $\mathbf{x}$ must be discarded and recomputed from the saved $\mathbf{x}$ and $\mathbf{G}(\mathbf{x})$, but now with a new step-size $\delta\tau$ smaller than $\Delta\tau$.

How much smaller ?

$$\text{new } \mathbf{x} := \mathbf{x} - \Delta\tau\cdot\mathbf{G}(\mathbf{x})$$

$$f(\text{new } \mathbf{x}) \approx f(\mathbf{x}) - \Delta\tau\cdot(4\|\mathbf{G}(\mathbf{x})\|^2 + \|\mathbf{G}(\mathbf{x}) + \mathbf{G}(\text{new } \mathbf{x})\|^2)/8 \; + \boldsymbol{O}(\Delta\tau^3)$$

If $f(\text{new } \mathbf{x}) \geq f(\mathbf{x})$, **then** new $\mathbf{x}$ must be recomputed, but with a new step-size $\delta\tau$ smaller than $\Delta\tau$. ***How much smaller ?***

A rough answer comes from approximating the term "$\boldsymbol{O}(\Delta\tau^3)$" above:

$$\boldsymbol{O}(\Delta\tau^3) \approx ¥(\mathbf{x}, \Delta\tau)\cdot\|\mathbf{G}(\mathbf{x})\|^2\cdot\Delta\tau^3$$

where $¥(\mathbf{x}, \Delta\tau)$ varies very slowly with $\Delta\tau$, we hope, and usually > 0.

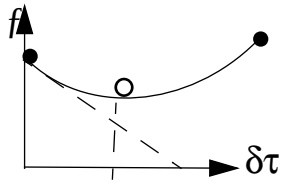
If the variation of $¥(\mathbf{x}, \Delta\tau)$ and of $\mathbf{G}(\text{new } \mathbf{x})$ with $\Delta\tau$ is ignored, we can predict that

$$\text{New } \mathbf{x} := \mathbf{x} - \delta\tau\cdot\mathbf{G}(\mathbf{x})$$

will produce $f(\text{New } \mathbf{x})$ hopefully well-approximated by this expression:

$$f(\mathbf{x}) - \delta\tau\cdot(4\|\mathbf{G}(\mathbf{x})\|^2 + \|\mathbf{G}(\mathbf{x}) + \mathbf{G}(\text{new } \mathbf{x})\|^2)/8 \; + ¥(\mathbf{x}, \Delta\tau)\cdot\|\mathbf{G}(\mathbf{x})\|^2\cdot\delta\tau^3.$$

Let's choose $\delta\tau$ to minimize this expression after obtaining $¥(\mathbf{x}, \Delta\tau)$ from the expression for $f(\text{new } \mathbf{x})$ above. The resulting $\delta\tau$ is below ...



$$\text{new } \mathbf{x} := \mathbf{x} - \Delta\tau\cdot\mathbf{G}(\mathbf{x})$$

$$f(\text{new } \mathbf{x}) \approx f(\mathbf{x}) - \Delta\tau\cdot(4\|\mathbf{G}(\mathbf{x})\|^2 + \|\mathbf{G}(\mathbf{x}) + \mathbf{G}(\text{new } \mathbf{x})\|^2)/8 \; + \boldsymbol{O}(\Delta\tau^3)$$

If $f(\text{new } \mathbf{x}) \geq f(\mathbf{x})$, **then** new $\mathbf{x}$ must be recomputed, but with a new step-size $\delta\tau$ smaller than $\Delta\tau$. Here is how to compute $\delta\tau$:

$$\text{Compute } \Delta f := f(\text{new } \mathbf{x}) - f(\mathbf{x}) ; \qquad (\text{now } \Delta f \geq 0)$$
$$V^2 := 4\|\mathbf{G}(\mathbf{x})\|^2 + \|\mathbf{G}(\mathbf{x}) + \mathbf{G}(\text{new } \mathbf{x})\|^2 ;$$
$$\delta\tau := \Delta\tau/\sqrt{\max}\{\, 0.8, \;\; 3 + 24\Delta f/(\Delta\tau\cdot V^2)\,\} \; .$$

And then New $\mathbf{x} := \mathbf{x} - \delta\tau\cdot\mathbf{G}(\mathbf{x})$ over-writes new $\mathbf{x}$; *etc*.

The "max{ 0.8, ...}" is explained below; it makes no difference yet.

If instead $f(\text{new } \mathbf{x}) < f(\mathbf{x})$ **then** retain new $\mathbf{x}$, $\mathbf{G}(\text{new } \mathbf{x})$ and $f(\text{new } \mathbf{x})$ but compute $\delta\tau$ as above for the next iteration-step's step-size in the expectation that it will nearly minimize the next computed value of f .

Now "max{ 0.8, ...}" acts to impose a bound $1.12 > \delta\tau/\Delta\tau$ to restrict step-sizes' growth-rate lest they get too big too soon and cause too many iterates new $\mathbf{x}$ to be recomputed.

In Regime #0 this scheme goes at least about as fast as any other tried.

When have iterates escaped from Regime #0 into Regime #1 ?

A symptom is that ...
recomputations of new **x** become infrequent, and
each of the past several **GD** iteration-steps diminishes $\|\mathbf{G}(\mathbf{x})\|$.

It's also a symptom of a rare approach to a (Hyperbolic) ***Saddle-Point.***
Both that approach and its departure can cost many iteration-steps.

Sometimes (rarely) that approach is foreshadowed by this inequality:

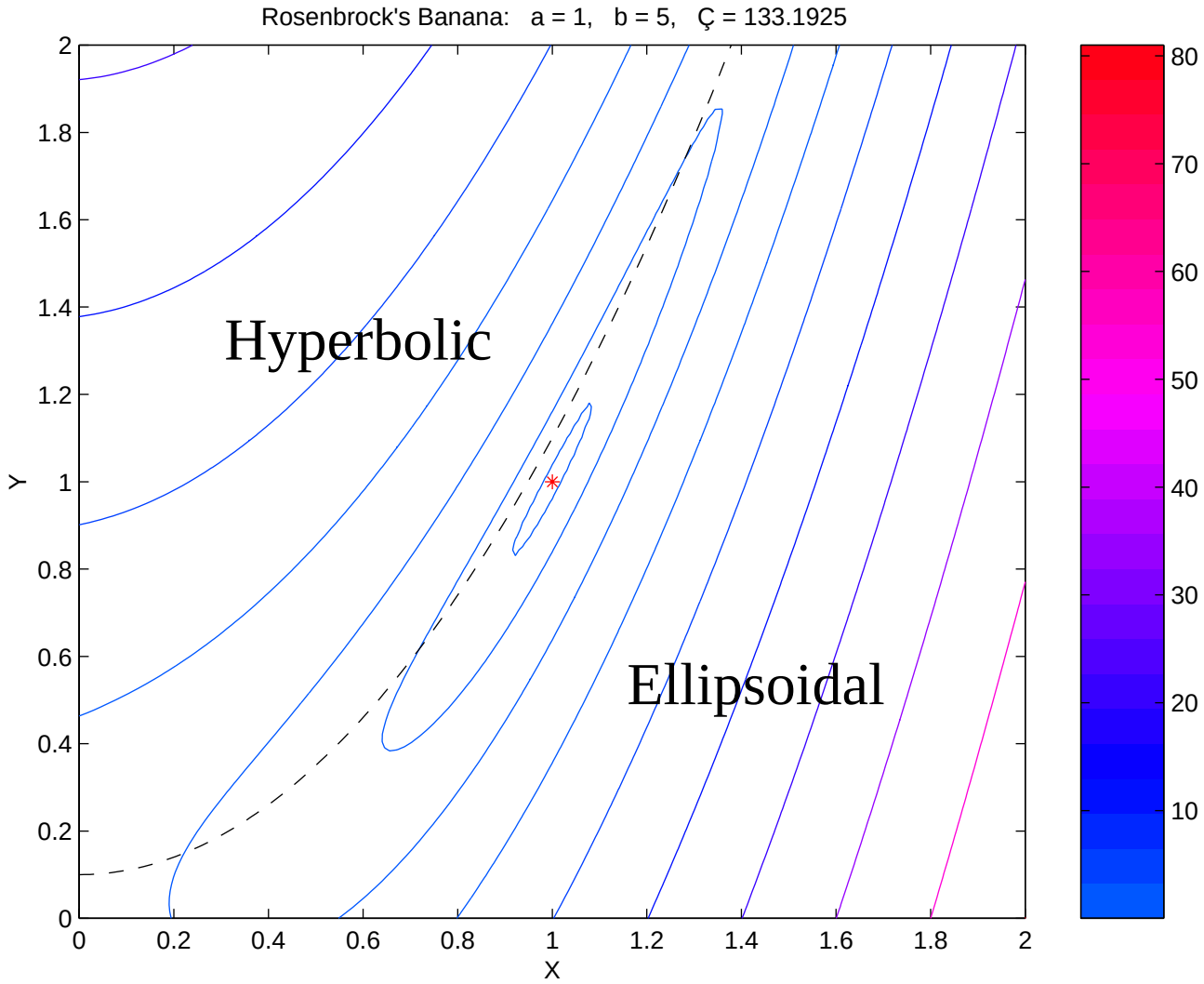
$$0 \geq \mathbf{G}(\mathbf{x})' \bullet (\mathbf{G}(\mathbf{x}) - \mathbf{G}(\text{new } \mathbf{x}))$$
$$\approx \Delta\tau \cdot \mathbf{G}(\mathbf{x})' \bullet (\mathbf{H}(\mathbf{x}) + \mathbf{H}(\text{new } \mathbf{x})) \bullet \mathbf{G}(\mathbf{x})/2 \pm O(\Delta\tau \cdot \|\mathbf{G}(\mathbf{x})\|)^3$$

because $\mathbf{G}' \bullet \mathbf{H} \bullet \mathbf{G} < 0$ whenever **G** enters a cone surrounding all the eigenvectors of **H** belonging to its negative eigenvalues. If ever that happens, or if $\|\mathbf{G}\|$ increases, stay or go back in Regime #0 .

To stimulate the detection of and escape from a Saddle-Point,
occasionally add to new **x** a very small random perturbation
orthogonal to both $\mathbf{G}(\mathbf{x})$ and $\mathbf{G}(\text{new } \mathbf{x})$ while
$\|\mathbf{G}(\text{new } \mathbf{x})\|$ is small but not yet tiny enough to disregard.

... BUT there may be no escape from Regime #0 .

H.H. Rosenbrock's ***Banana***: Its Minimum lies almost in Regime #0 .



The black parabola, separating hyperbolic from ellipsoidal curvatures, runs so near the Minimum * that iterates converging to it often straddle the parabola.

SUMMARY

With no prior knowledge of
the Hessian's Norm $\|\mathbf{H}\|$ nor Condition Number Ç ,

Gradient-Based iterations are performed,
computing f and its gradient $\mathbf{G}$ usually just once per iteration-step,
and computing step-sizes $\Delta\tau$ from by-products of f and $\mathbf{G}$,

that **seem** ***ultimately*** to achieve convergence at rates
roughly as good as the best that any other Gradient-Based iteration,
like **GD**, **AGD**, and every version of **GD+M**, could achieve

as if all their Hyper-Parameters were constants computed optimally
from rarely known valies of $\|\mathbf{H}\|$ and Ç .

Alas, **"seem"** is not a mathematical proof.

And as M. Keynes said, "In the long run, we are all dead."

Questions for Another Occasion:

Might **AGD** escape from Regime #0 faster than **GD** can?
(**AGD** is a 2nd Order Discretization; **GD** is a 1st Order.)

What happens to examples, —— lots of them?

Is f not smooth? A self-inflicted wound?

How accurately should we try to compute a minimizing $\mathbb{x}$?

When can the iteration be stopped with a satisfactory $\mathbb{x}$??

What makes one computed minimum $\mathbb{x}$ better than another?

What's wrong with "Stochastic Gradient Descent" ?

How are valid inferences drawn from fictitious hypotheses?
(Actually $\Delta\tau$ is ***not*** small, nor is $\|\mathbf{G}(\mathbf{x})\|$ until near the end.)

Supporting Equipment for which I owe Thanks:

Explorations were conducted on an ancient but very reliable
Intel 302 running MS DOS 6.2, PE2,
MATLAB 3.5 and DERIVE 4.11 .

This document was produced on a cranky old
Apple Power-Mac G4 running OS 9.2 ,
MATLAB 5.2 and FrameMaker 5 .

This document is posted at people.eecs.berkeley.edu/~wkahan/26Sept19.pdf
Details are posted at people.eecs.berkeley.edu/~wkahan/STEPSIZE.pdf
Earlier thoughts at people.eecs.berkeley.edu/~wkahan/7Nov18.pdf