# Landscapes & Algorithms for Quantum Control

Sophie (Sonia) Schirmer

Dept of Applied Maths & Theoretical Physics
University of Cambridge, UK

April 15, 2011

# Control Landscapes: What do we really know?

**“Kinematic control landscapes” generally universally nice**

1. Pure-state transfer problems (finite-dim.)
   - Only global extrema, no saddles
2. Density-matrix/observable optimization problems
   - Saddles but no suboptimal extrema
3. Unitary operator optimization: depends on domain
   - $U(N)$: Critical manifolds but no traps
   - $SU(N)$: Beware of root-of-unity traps
   - $PU(N)$: Critical manifolds but no traps

**Actual control landscapes — things get messy**

# Kinematic vs Actual Control Landscape

**Idea:** Decompose map from control function space $\mathfrak{F} : L^2[0,T] \mapsto \mathbb{R}$ into parts

1. $U_f : L^2[0,T] \mapsto U(N)$ and
2. $\mathfrak{G} : U(N) \mapsto \mathbb{R}$ (easy problem)

If we are lucky, maybe we can study the control landscape of (2) — kinematic control landscape — and apply the results to the actual (hard) problem (1).

**More mathematically precisely:** If (1) is

(a) surjective (equivalent to controllability) and

(b) has maximum rank everywhere

then the actual control landscape looks like the kinematic one.

**But are we this lucky?**

# Assumption (a) – controllability

**Definition:** Given the control system $H = H_0 + f(t)H_1$ let $\mathfrak{L}_0$ be the set of commutator expression in $iH_0$ and $iH_1$ joined with $iH_1$; $\mathfrak{L} = \mathfrak{L}_0 \cup iH_0$. $G_0 = \exp(\mathfrak{L}_0)$ and $G = \exp(\mathfrak{L})$.

## Theorem (Controllability)

*If $\mathfrak{L} = \mathfrak{su}(N)$ or $\mathfrak{u}(N)$ then there exists a time $T_{\max}$ and neighborhood $\mathcal{N}$ of $(-iH_0, -iH_1)$ in $\mathfrak{L} \times \mathfrak{L}$ such that for all $s \in \mathcal{N}$ and $g \in G$ there is a control taking $s$ to $g$ in some time $T < T_{\max}$.*

## Theorem (Exact-time controllability)

*If $\mathfrak{L}_0 = \mathfrak{su}(N)$ or $\mathfrak{u}(N)$ then there is a critical time $T_c$ and neighborhood $\mathcal{N}$ of $(-iH_0, -iH_1) \in \mathfrak{L} \times \mathfrak{L}$ such that for all $s \in \mathcal{N}$, $g \in G$ and $T > T_c$ there is a control taking $s$ to $g$ in time $T$.*

**Very many quantum systems satisfy $\mathfrak{L}_0 = \mathfrak{su}(N)$ or $\mathfrak{u}(N)$.**

# Assumption (b) – no singular points

1. **Gate optimization:** assumption (b) never satisfied over any function space containing constant controls as all of these rank-deficient; many non-constant controls also fail this test.
2. **State transfer/observable optimization:** for every bilinear Hamiltonian control system there exist pairs of initial and target states for which there are singular controls.

**Failure of (b) implies that actual landscape need not resemble kinematic one — Further analysis needed!**

1. Counter-examples show that suboptimal extrema exist for all types of problems above [arXiv:1004.3492]
2. Many critical points with semi-definite Hessian 2nd-order potential traps [PRL 106,120402]

**Are potential traps always actual traps?**

# Control Landscapes: What is a Trap?

Not a trivial question!

1. Is it a local extremum for which fidelity (error) assumes value less than global maximum (minimum)?
2. Is it any point that can attract trajectories?

(2) includes saddles that have domains of attraction but in any neighborhood there are also points not attracted to saddle.

**3 Cases:** domain of attraction

1. contains a neighborhood of the point – the case for strict local extrema under the usual assumptions – typical traps
2. within any neighborhood of the point is open (has positive measure) but not everything
3. for the point is lower dimensional or has empty interior (measure-zero)

# Simple example: $f(x,y) = x^2 + \alpha y^n$, $n \geq 2$

$(0,0)$ **critical point:** $f_x(0,0) = 2x = 0$, $f_y = n\alpha y^{n-1} = 0$.

**Hessian:**

$$H(x,y) = \begin{pmatrix} 2 & 0 \\ 0 & \alpha n(n-1)y^{n-2} \end{pmatrix}$$

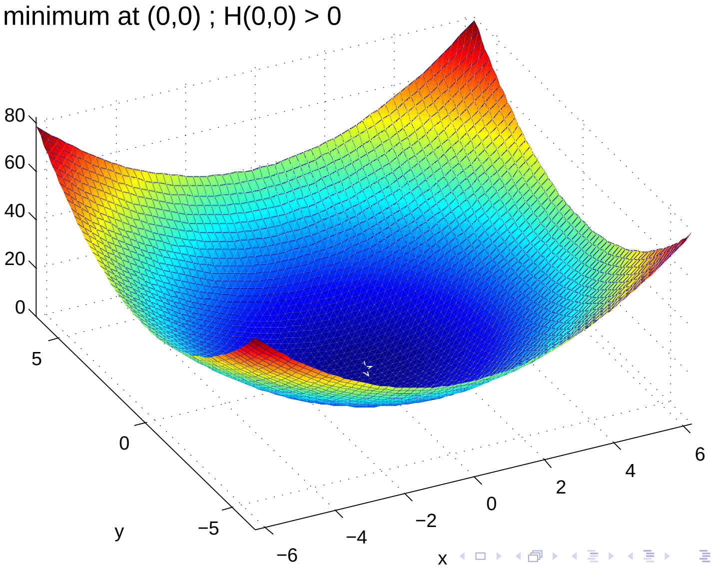
$H(0,0) > 0$ (positive definite) if $n = 2$ and $\alpha > 0$

$H(0,0) \geq 0$ (positive semi-definite) for $n > 2$.

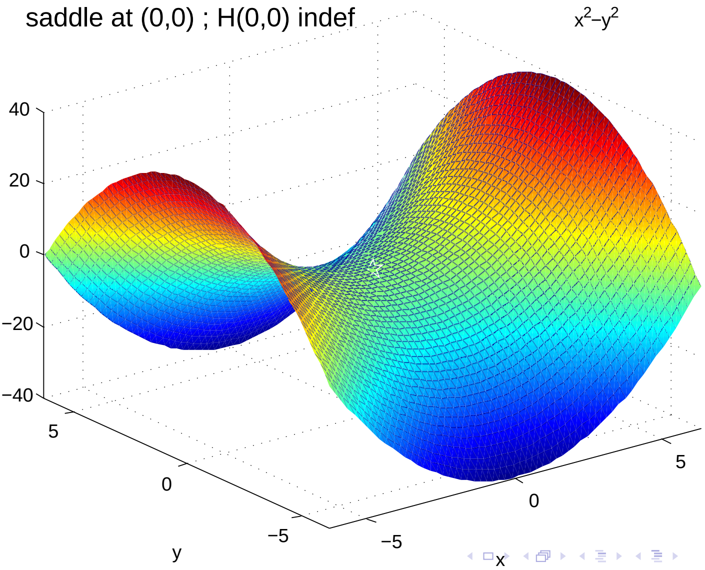
**Definition:** "Second-order" trap: Hessian $H(0,0) \geq 0$

**Which of the following are traps?**

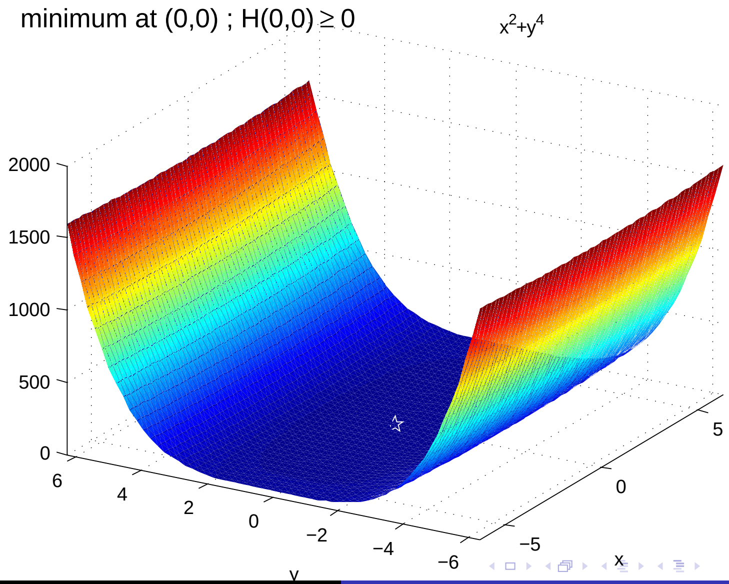
# Minimum, Hessian positive definite



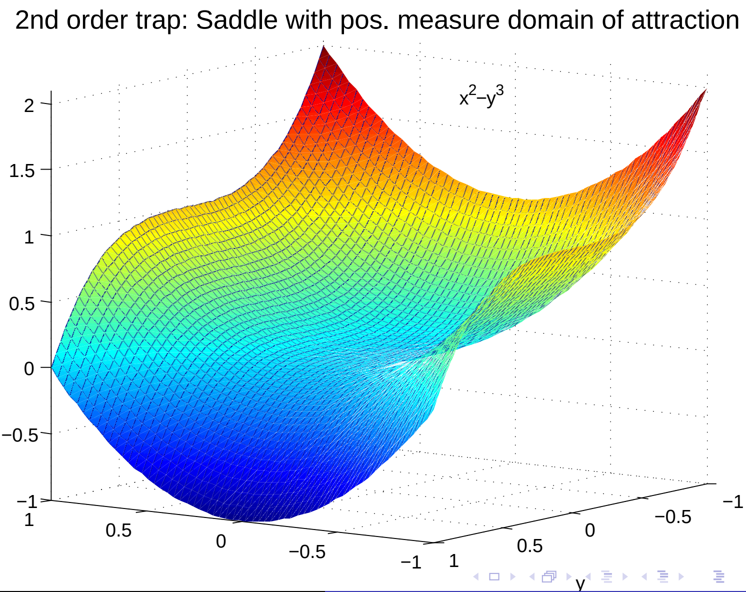
# Saddle, Hessian indefinite



# Second-order trap – minimum



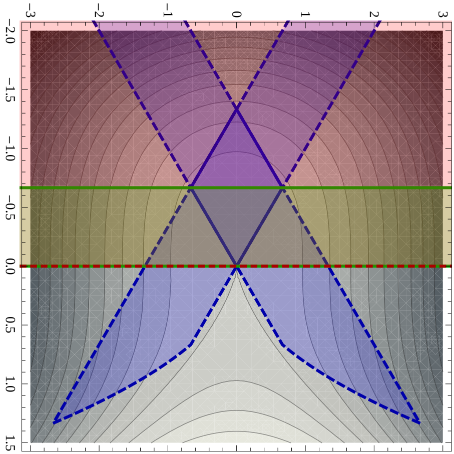
# 2nd-order trap – saddle / pos. measure attractive set



# So what is a trap now?

1. Not all "second-order" traps are extrema – many saddles.
   - Semi-definite Hessian not sufficient for extremum.
   - Second-order traps that are saddles exist for quantum control problem — see Example 2 in arXiv:1004.3492.
   - Could argue saddles are not strictly traps as in any neighborhood some points will not converge to the saddle.
2. **But** saddles have domains of attraction; second-order traps can have positive measure domains of attraction.
3. Domains of attraction depend on algorithm!
   - Local extrema traps for all local optimization algorithms – how many second-order traps are suboptimal extrema?
   - Convergence to saddles possible, especially if they have positive measure domains of attraction!

# Domains of attraction of a saddle




**Trapping regions** for different optimization algorithms:

green: steepest descent (concurrent)

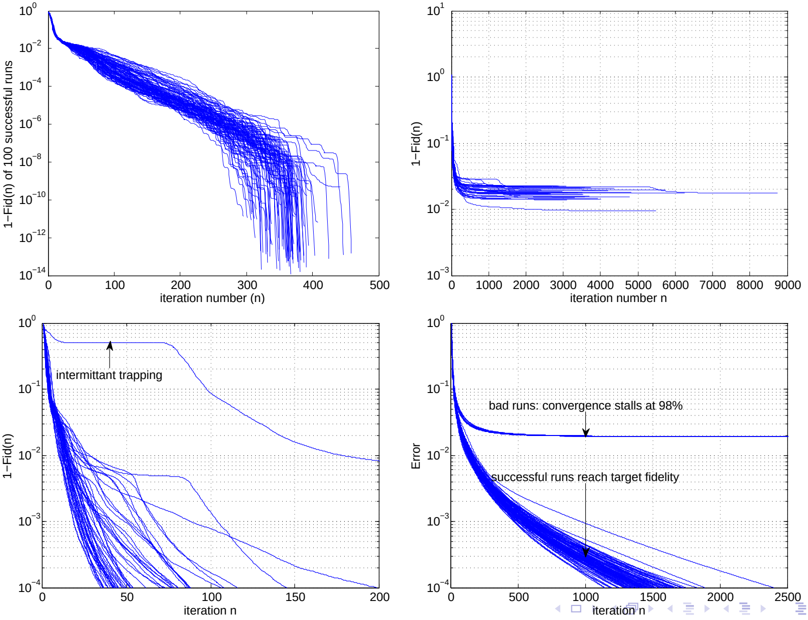
red: Newton (concurrent)

blue: sequential

What is the probability of trapping?

How can we detect when an optimization run is going badly?

# Convergence: The Good, The Bad & The Ugly



# Optimization Algorithms: Which perform best?

Numerical optimization **expensive** hence **efficiency** is key.

Many algorithms, many choices — **How do we choose?**

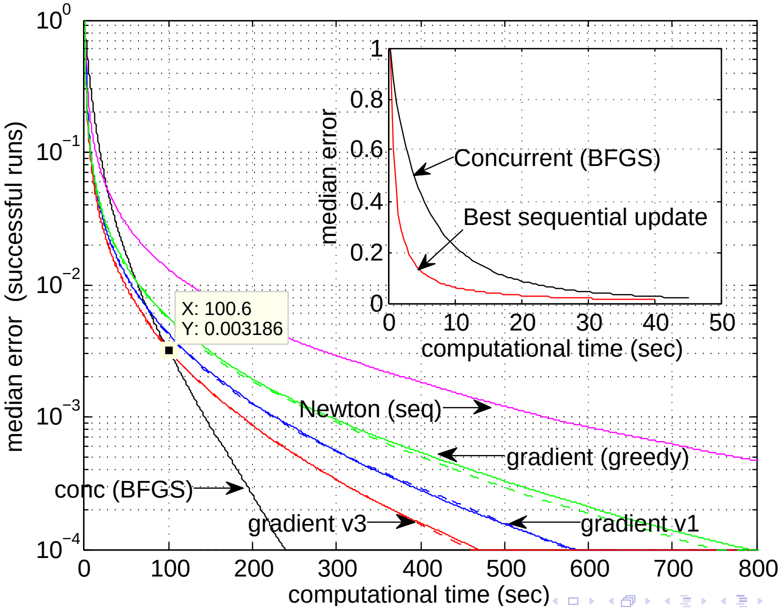
1. **Global vs Local Optimization:**
   - Optimization algorithms designed to find local extrema far more efficient than global optimization strategies
   - Local optimization algorithms preferable if probability of trapping low.
2. **Sequential vs concurrent update:**
   - Sequential update (Krotov) inspired by dynamical systems: discrete version of continuous flow (Krotov)
   - Concurrent update motivated by conventional optimization
3. **How to choose effective update rules?**
   - Gradient vs higher-order methods (Newton/quasi-Newton)
   - Search length adjustment (line search)
4. **Discretization:** How to parametrize the controls?

# Preliminary observations

1. Local optimization algorithms outperform global ones in general — trapping probability low for suitable initial fields
2. Far from a global optimum sequential update algorithms generally help you escape faster — but only up to a point.

   Sequential update allow larger local changes of the fields but rapid initial gains may be paid for in terms of decreased asymptotic rate of convergence.
3. Close to the top concurrent update has clear edge as it can exploit nonlocal temporal correlations of the fields.
4. Discretization must be considered in analysis — affects
   - Gradient accuracy — don't rely on continuous limit
   - Effective line search strategies — use quadratic model
   - Rate of convergence — don't be too greedy
   - Choice of penalty terms — regularizing penalty terms unnecessary for finite time steps

# Algorithm Comparison



# Summary and Conclusions

1. **Control Landscape:**
   Actual dynamic landscape appears to be far richer than kinematic analysis suggests — many questions
2. **Numerical Algorithms** for solving opt. control problems

   Progress in understanding convergence behaviour, etc. but significant room for improvement!

   Many questions — e.g., are some algorithms more likely to get trapped in (higher-order) saddles?
3. **Parametrization of controls/discretization crucial**
   Analysis of infinite dimensional control problem may not actually tell us much about what happens in discrete case.
4. **Robustness:** Optimal control solutions naturally robust but some perturbations that are more detrimental than others.