Recent Developments in Large Scale Optimization

Panos Parpas

Department of Computing
Imperial College London
www.doc.ic.ac.uk/~pp500
p.parpas@imperial.ac.uk

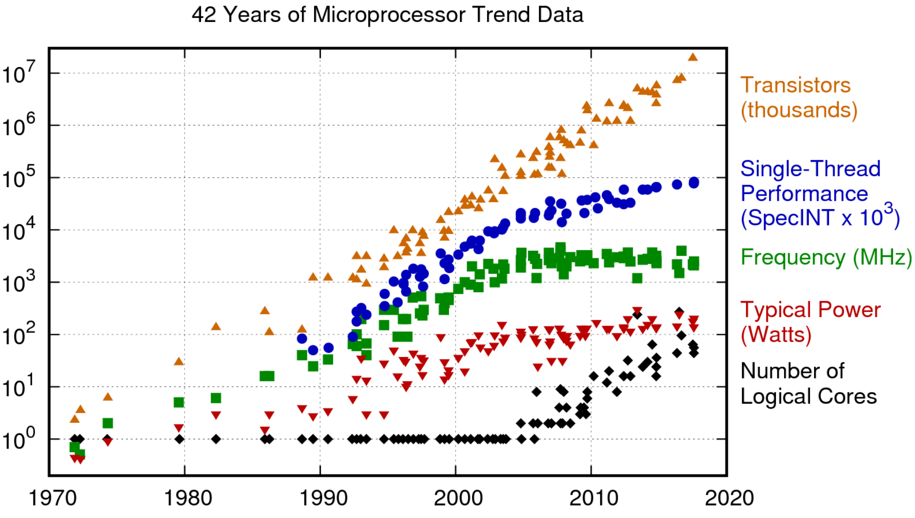
10-Jan-2022

Outline

- Numerical methods and computer architectures are tightly linked
- Optimization methods that will be useful in the future must make use of parallelism, minimize energy use & communication
- Present three algorithmic developments:
  - Multilevel methods
  - Distributed optimization
  - Stochastic representation of a deterministic problem

Joint work with: A. Borovykh, C.P Ho, A. V. Hovhannisyan, N. Kantas, T. Lelièvre, G.A. Pavliotis, J.C Salazar, N. Tsipanakis.

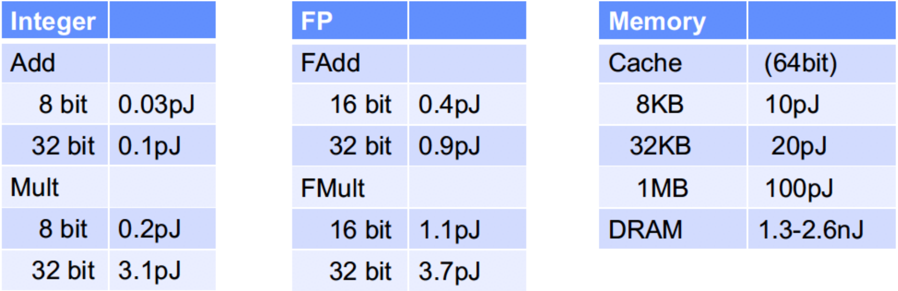
Microprocessor Trends



Original data up to the year 2010 collected and plotted by M. Horowitz, F. Labonte, O. Shacham, K. Olukotun, L. Hammond, and C. Batten
New plot and data collected for 2010-2017 by K. Rupp

Memory Energy Costs for Memory Access

40nm, 8-core processor with an 8MB last-level cache[†]



Integer	
Add	
8 bit	0.03pJ
32 bit	0.1pJ
Mult	
8 bit	0.2pJ
32 bit	3.1pJ

FP	
FAdd	
16 bit	0.4pJ
32 bit	0.9pJ
FMult	
16 bit	1.1pJ
32 bit	3.7pJ

Memory	
Cache	(64bit)
8KB	10pJ
32KB	20pJ
1MB	100pJ
DRAM	1.3-2.6nJ

[†] M. Horowitz, *Computing's Energy Problem (and what we can do about it)*, ISSCC 2014

Energy costs to **develop** a modern neural network[††]

Energy costs to **develop** a modern neural network[††]

- **750,000 CPU days**
- From Athens to Thessaloniki $\sim$ 500 times!

[††] H. Assi, J.C Duchi, *The importance of better models in stochastic optimization*

BULGARIA
MACEDONIA
ALB.
TURK.
Kavála
Thessaloníki
Alexandroúpolis
Mount Olympus
Kérkyra
Ioánnina
Lárisa
Igoumenítsa
Aegean
Sea
TURKEY
Chalkís
Chíos
ATHENS
Pátrai
Piraeus
Sámos
Ionian Sea
Corinth Canal
Náxos
Rhodes
Sea of Crete
Mediterranean Sea
Irákleion
Crete
0 50 100 km
0 50 100 mi
21
24
42
39
36
27

500 km

A General Optimization Model

$$\min_{v \in \mathcal{V}} f(v)$$

- f **objective**/loss function
- v design/**decision** variables
- $\mathcal{V}$ is the "design"/**constraint** space

Optimization Methods use Local Approximations

$$\boxed{v^\star \in \arg\min_{v \in \mathcal{V}} f(v)}$$

- Guess a solution v
- Find d **search direction**:

$$f(v+d) < f(v)$$
$$\|v+d-v^\star\| < \|v-v^\star\|$$

- Optimize **local** approximation:

$$f(v+d) \approx \underbrace{f(v) + \nabla f(v)^\top d +}_{\text{linear: } l_v(d)} \underbrace{\frac{1}{2} d^\top B d}_{\text{quadratic: } q_v(d)}$$

- **Update:**

$$\boxed{v \leftarrow v + d}$$

Why use a Quadratic Approximation?

- **Greedy/Pragmatic**

$$f(v+d) \approx \underbrace{f(v) + \nabla f(v)^\top d}_{\text{linear: } l_v(d)} + \underbrace{\frac{1}{2} d^\top B d}_{\text{quadratic: } q_v(d)}$$

Why use a Quadratic Approximation?

- **Greedy/Pragmatic**

$$f(v+d) \approx \underbrace{f(v) + \nabla f(v)^\top d +}_{\text{linear: } l_v(d)} \underbrace{\frac{1}{2} d^\top B d}_{\text{quadratic: } q_v(d)}$$

- Smoothness: $f(v+d) \leq l_v(d) + \frac{L}{2}\|d\|^2$
- Convexity: $f(v+d) \geq l_v(d)$
- Strong convexity:

$$l_v(d) + \frac{1}{2}\mu\|d\|^2 \leq f(v+d) \leq l_v(d) + \frac{L}{2}\|d\|^2$$

Why use a Quadratic Approximation?

- **Greedy/Pragmatic**

$$f(v+d) \approx \underbrace{f(v) + \nabla f(v)^\top d +}_{\text{linear: } l_v(d)} \underbrace{\frac{1}{2} d^\top B d}_{\text{quadratic: } q_v(d)}$$

- Smoothness: $f(v+d) \leq l_v(d) + \frac{L}{2}\|d\|^2$
- Convexity: $f(v+d) \geq l_v(d)$
- Strong convexity:

$$l_v(d) + \frac{1}{2}\mu\|d\|^2 \leq f(v+d) \leq l_v(d) + \frac{L}{2}\|d\|^2$$

First Order (B is typically diagonal), Gradient Descent: Stochastic, Proximal, Accelerated, Block Coordinate, ...
Second Order($B = \nabla^2 f$): Newton Method, Quasi-Newton, Sketched, Subsampled ...

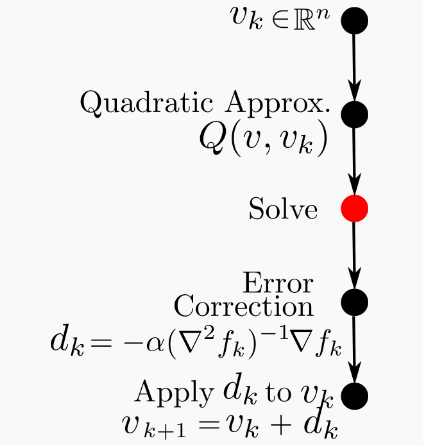
Multi-level/resolution Algorithms

$$\boxed{\min_{v \in \mathbb{R}^n} f(v)}$$

Multi-level/resolution Algorithms

$$Q(v, v_k) = f(v_k) + \langle \nabla f_k, v - v_k \rangle + \frac{1}{2\alpha_k} \langle v - v_k, \nabla^2 f_k (v - v_k) \rangle$$

Quadratic Approximation



$v_k \in \mathbb{R}^n$

Quadratic Approx. $Q(v, v_k)$

Solve

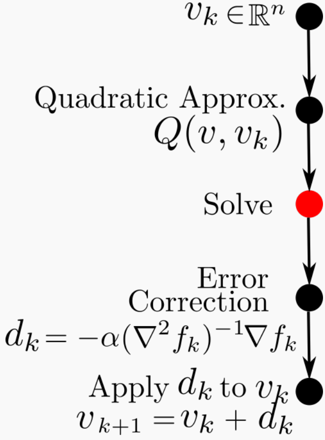
Error Correction $d_k = -\alpha(\nabla^2 f_k)^{-1} \nabla f_k$

Apply d_k to v_k $v_{k+1} = v_k + d_k$

Multi-level/resolution Algorithms

Use a low resolution problem with *favorable characteristics*

Quadratic Approximation



$v_k \in \mathbb{R}^n$

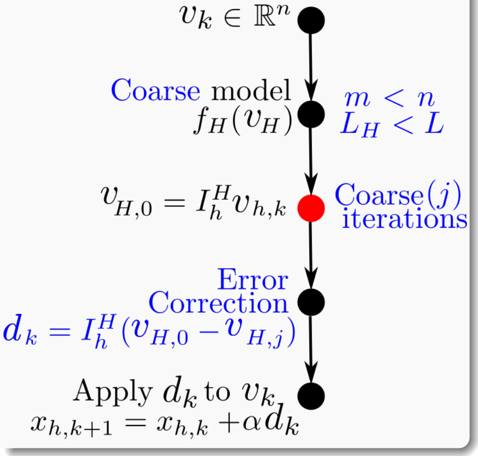
Quadratic Approx.
$Q(v, v_k)$

Solve

Error
Correction
$d_k = -\alpha(\nabla^2 f_k)^{-1}\nabla f_k$

Apply d_k to v_k
$v_{k+1} = v_k + d_k$

Coarse Approximation



$v_k \in \mathbb{R}^n$

Coarse model
$f_H(v_H)$

$m < n$
$L_H < L$

$v_{H,0} = I_h^H v_{h,k}$

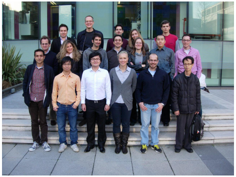
Coarse(j)
iterations

Error
Correction
$d_k = I_h^H(v_{H,0} - v_{H,j})$

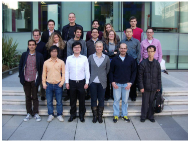
Apply d_k to v_k
$x_{h,k+1} = x_{h,k} + \alpha d_k$

The Three Main Ingredients of Multilevel Algorithms

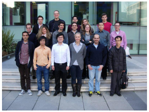
1. **Define Coarse Model**



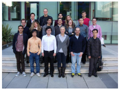
$1280 \times 1024 = 1310720$



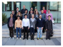
$1024 \times 768 = 786432$



$800 \times 600 = 480000$



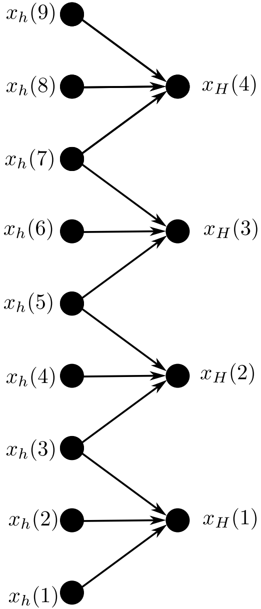
$640 \times 480 = 307200$



$320 \times 240 = 76800$

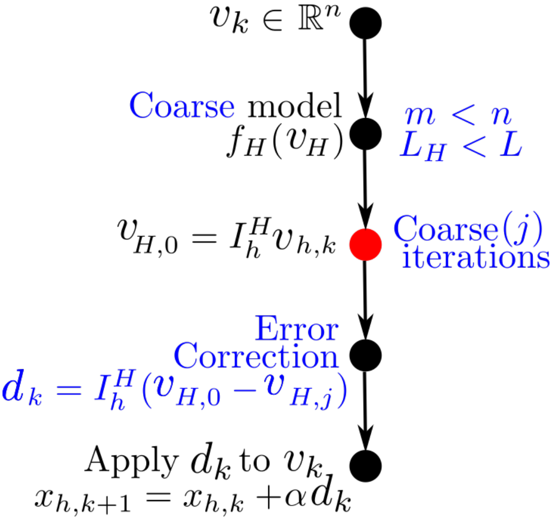
The Three Main Ingredients of Multilevel Algorithms

1. Define Coarse Model
2. **Information Transfer**



The Three Main Ingredients of Multilevel Algorithms

1. Define Coarse Model
2. Information Transfer
3. **Exploit the Coarse Model**



$v_k \in \mathbb{R}^n$

Coarse model $f_H(v_H)$

$m < n$
$L_H < L$

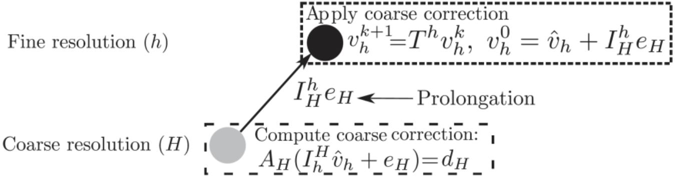
$v_{H,0} = I_h^H v_{h,k}$

Coarse(j) iterations

Error Correction
$d_k = I_h^H(v_{H,0} - v_{H,j})$

Apply d_k to v_k
$x_{h,k+1} = x_{h,k} + \alpha d_k$

Example I: **Markov Decision Processes**



Apply coarse correction

$v_h^{k+1} = T^h v_h^k, \; v_h^0 = \hat{v}_h + I_H^h e_H$

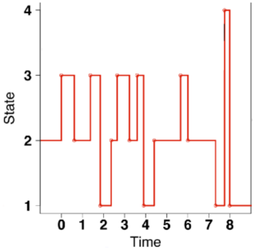
Fine resolution (h)

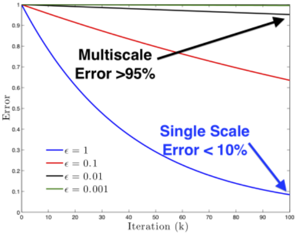
$I_H^h e_H$ ← Prolongation

Coarse resolution (H)

Compute coarse correction:

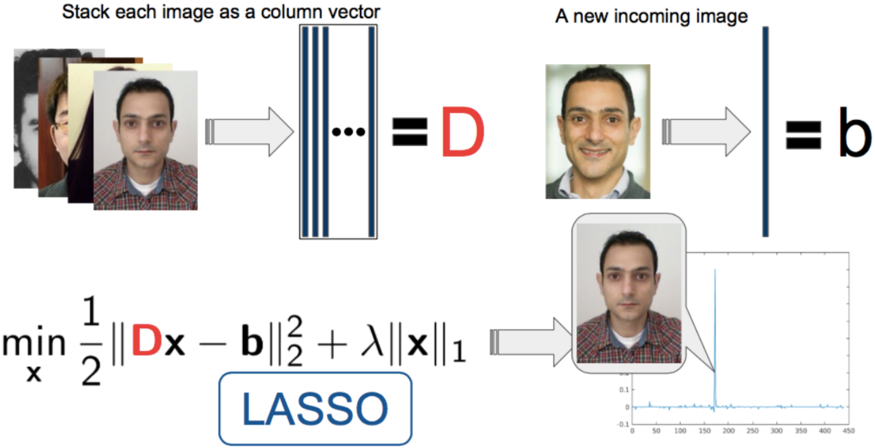
$A_H(I_h^H \hat{v}_h + e_H) = d_H$





C.P Ho, P.P. *Singularly Perturbed Markov Decision Processes: A Multiresolution Algorithm*, SIAM Journal on Control and Optimization, 52(6), 3854-3886, 2014.

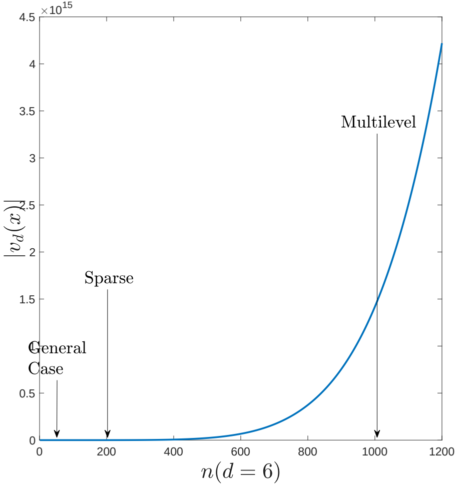
Example II: **Machine Learning**



$$\min_{\mathbf{x}} \frac{1}{2}\|\mathbf{D}\mathbf{x}-\mathbf{b}\|_2^2+\lambda\|\mathbf{x}\|_1$$

V. Hovhannisyan, P.P, and S. Zafeiriou. *MAGMA: Multi-level accelerated gradient mirror descent algorithm for large-scale convex composite minimization*, SIAM Journal on Imaging Sciences, 2016.

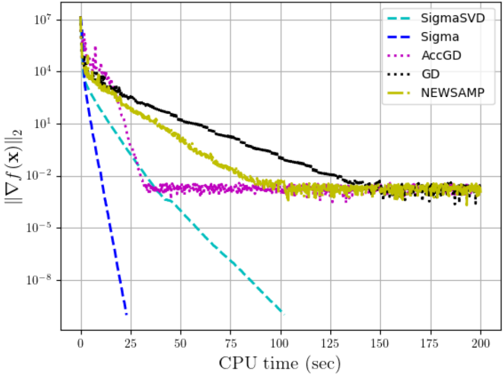
Example III: **Semi-definite Programming**

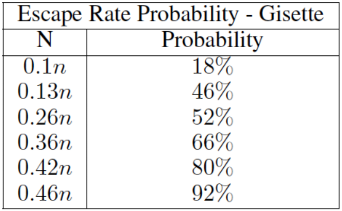


J.C Salazar, P.P. *A Multigrid approach to SDP relaxations of sparse polynomial optimization problems* , SIAM Journal on Optimization, 2018.

K., Igor, V. Magron, J. Volčič.*Optimization over trace polynomials.* A. H. Poincaré, 2022.

Example IV: **Second Order Algorithms in ML**





Escape Rate Probability - Gisette	
N	Probability
$0.1n$	18%
$0.13n$	46%
$0.26n$	52%
$0.36n$	66%
$0.42n$	80%
$0.46n$	92%

N. Tsipinakis,P. Tigas, P.P, *A Multilevel Low-Rank Newton Method with Super-linear Convergence Rate and its Application to Non-convex Problems* , submitted , 2022.

Related Work

- Nash, S. G. A multigrid approach to discretized optimization problems. *Optimization Methods and Software, 2000*
- Gratton, S., Sartenaer, A., Toint, P. L. . Recursive trust-region methods for multiscale nonlinear optimization. *SIAM Journal on Optimization, 2008*
- W., Zaiwen, and D. Goldfarb. A line search multigrid method for large-scale nonlinear optimization. *SIAM Journal on Optimization, 2009*
- A. Borzi, On the convergence of the mg/opt method. *PAMM, 5(1):735-736, 2005.*
- A. Borzi and V. Schulz. Multigrid methods for pde optimization. *SIAM review, 51(2):361 395, 2009.*
- S. Gratton, M. Moue, A. Sartenaer, P.L Toint, and D. Tomanos. Numerical experience with a recursive trust-region method for multilevel nonlinear bound-constrained optimization. *Optimization Methods & Software, 25(3):359–386, 2010.*
- S.G Nash. Properties of a class of multilevel optimization algorithms for equality- constrained problems. *Optimization Methods and Software, 29, 2014.*
- S.G. Nash and R.M Lewis. Assessing the performance of an optimization-based multilevel method. *Optimization Methods and Software, 26(4-5):693–717, 2011.*

Exploiting Model Geometry

Optimization over simplex: $\min_{x \in \Delta} f(x)$, $\Delta = \{x_i \geq 0, \sum_{i=1}^{d} x_i = 1\}$.
Example application: Quantum state tomography i.e. estimating the state of qubits given measurements

Two possible algorithms:

Exploiting Model Geometry

Optimization over simplex: $\min_{x\in\Delta} f(x)$, $\Delta = \{x_i \geq 0, \sum_{i=1}^{d} x_i = 1\}$. Example application: Quantum state tomography i.e. estimating the state of qubits given measurements

Two possible algorithms:

1. Gradient Descent: $x_{k+1} = \Pi_\Delta[x_k - \tau \nabla f(x_k)]$
where $\Pi_\Delta[\cdot]$ orthogonal projection onto Δ

Exploiting Model Geometry

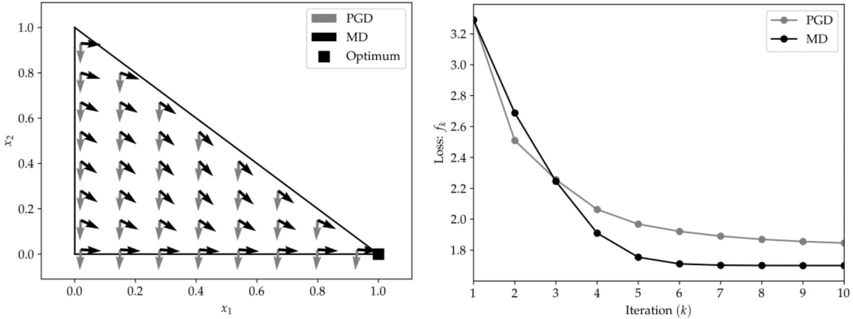
Optimization over simplex: $\min_{x \in \Delta} f(x)$, $\Delta = \{x_i \geq 0, \sum_{i=1}^{d} x_i = 1\}$.
Example application: Quantum state tomography i.e. estimating the state of qubits given measurements

Two possible algorithms:

1. Gradient Descent: $x_{k+1} = \Pi_\Delta[x_k - \tau \nabla f(x_k)]$
where $\Pi_\Delta[\cdot]$ orthogonal projection onto Δ

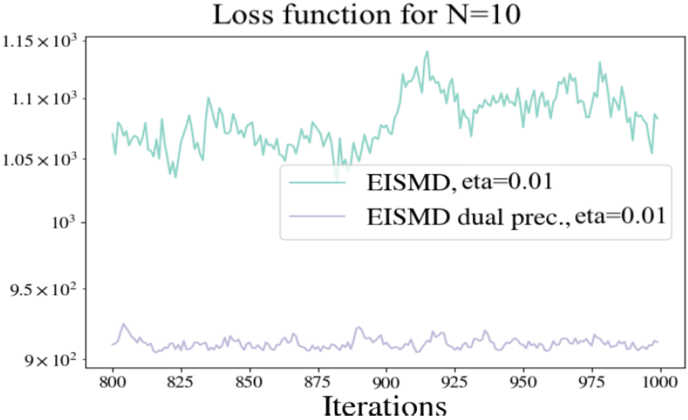
2. Mirror Descent: $x_{k+1} = \arg\min \tau \nabla f(x_k)^\top (x - x_k) + D_\Phi(x, x_k)$
where Φ is a Bregman divergence chosen to reflect the geometry of the simplex.

Exploiting Model Geometry



Stochastic Mirror Descent for Convex Optimization with Consensus Constraints, A. Borovykh, N. Kantas, P.P, G.A. Pavliotis, submitted 2022

Exploiting Model Geometry



Stochastic Mirror Descent for Convex Optimization with Consensus Constraints, A. Borovykh, N. Kantas, P.P, G.A. Pavliotis, submitted 2022

Computing Index-1 Saddle Points

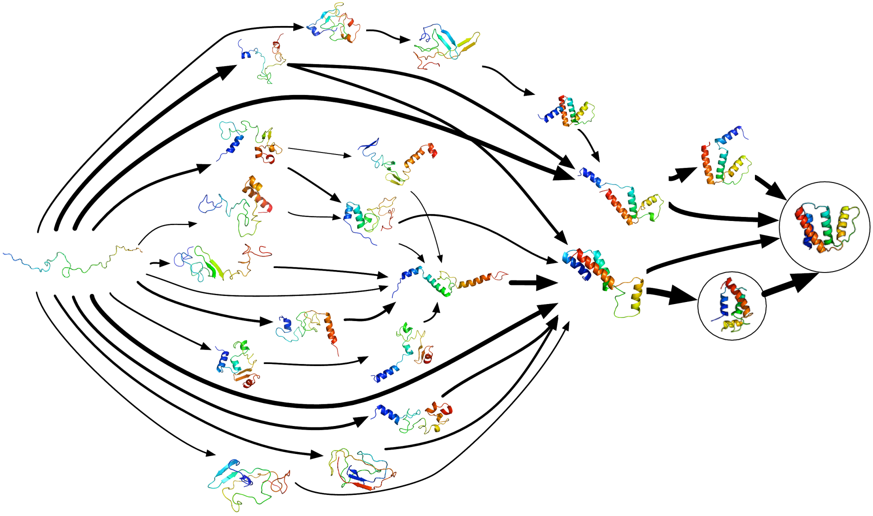
Problem Statement: Given a function $f : \mathbb{R}^d \to \mathbb{R}$ compute (possibly all) index-1 saddle points.

Index-1 saddle point: $\nabla f(x^\star) = 0$, $\nabla^2 f(x^\star)$ has one negative eigenvalue and the rest positive.

Applications: Material Science, Chemical Physics, a very challenging problem that cannot be 'tractably' written as an optimization problem.

Using Witten Laplacians to locate index-1 saddle points, T. Lelièvre, P.P, https://arxiv.org/abs/2212.10135, 2022

Protein folding & misfolding



Link between Sampling and Optimization

$$dX_t = -\nabla f(X_t)dt + \sqrt{\beta^{-1}}dB(t),$$

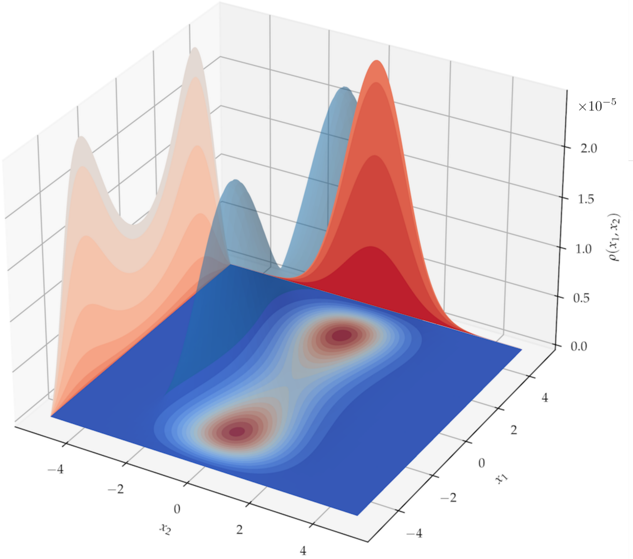
where B is the standard Brownian motion and $\beta > 0$ is the so called 'temperature' parameter

$(X_t)_{t\geq 0}$ is ergodic with respect to Boltzmann-Gibbs measure $Z^{-1}\exp(-\beta V(x))\,dx$

The p.d.f of the SDE above satisfies the Fokker-Planck PDE:

$$\frac{\partial \rho}{\partial t} = \operatorname{div}\left(\rho \nabla V\right) + \beta^{-1}\Delta\rho.$$

Example on the two-well potential: $V(x_1, x_2) = (x_1^4 - x_1^2) + x_2^2$,
Fokker Planck PDE concentrates on local minima



A stochastic representation of the Witten PDE

It is known that the Witten PDE:

$$\partial_t \phi_i = \operatorname{div}_x(\nabla V(x)\phi_i) + \beta^{-1}\Delta_x \phi_i - [\nabla^2 V(x)\phi]_i, \quad i = 1, \dots d$$

Concentrates on index-1 saddle points.
Consider the system:

$$\begin{aligned} dX_t &= -\nabla V(X_t)\,dt + \sqrt{2\beta^{-1}}\,dB_t, \\ dY_t &= -\nabla^2 V(X_t) Y_t\,dt, \end{aligned}$$

The Fokker-Planck equation associated with the above is:

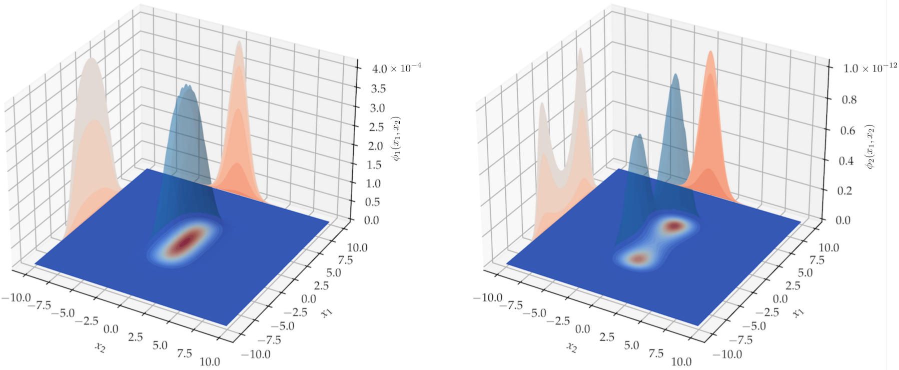
$$\frac{\partial \kappa}{\partial t} = \operatorname{div}_x(\nabla V(x)\kappa + \beta^{-1}\nabla_x \kappa) + \operatorname{div}_y(\nabla^2 V(x) y \kappa)$$

We show that

$$\phi(t,x) = \int_{\mathbb{R}^d} y\kappa(t,x,y)\,dy.$$

solves the Witten PDE.

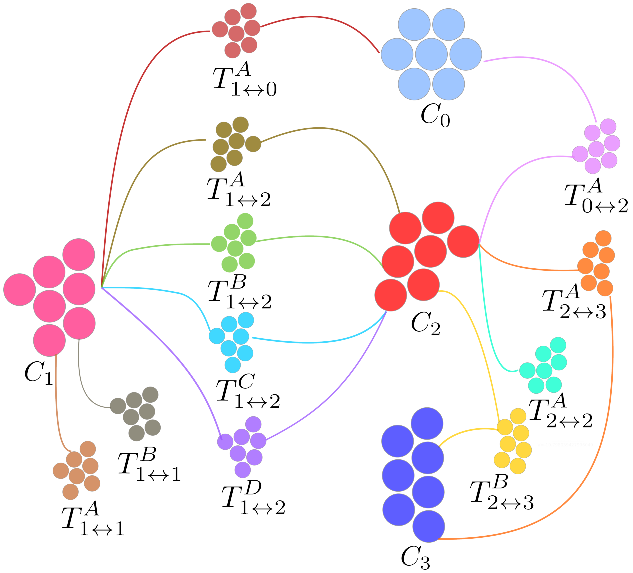
Example on the two-well potential: $V(x_1, x_2) = (x_1^4 - x_1^2) + x_2^2$,
Witten PDE concentrates on saddle points



Simulating the SDE - Animation

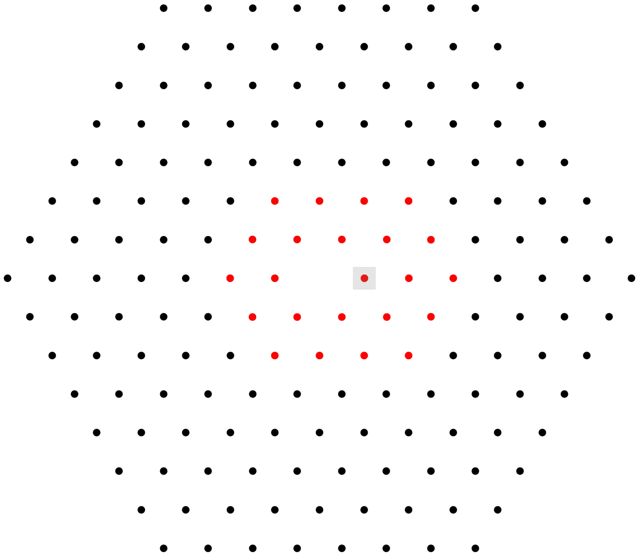
2ddoublewell.mp4

Lennard Jones 7 atoms in 2d



$T^A_{1\leftrightarrow 0}$
C_0
$T^A_{1\leftrightarrow 2}$
$T^A_{0\leftrightarrow 2}$
$T^B_{1\leftrightarrow 2}$
C_2
$T^A_{2\leftrightarrow 3}$
C_1
$T^C_{1\leftrightarrow 2}$
$T^A_{2\leftrightarrow 2}$
$T^B_{1\leftrightarrow 1}$
$T^D_{1\leftrightarrow 2}$
$T^B_{2\leftrightarrow 3}$
$T^A_{1\leftrightarrow 1}$
C_3

Vacancy Diffusion in 2d



Vacancy Diffusion - Scaling with dimension

Dimension	CPU-Time (s)	k
18	0.85	500
46	1.68	700
138	3.84	300
202	5.59	1000
278	8.21	700

Conclusions

- Numerical methods and computer architectures are tightly linked
- Three algorithmic developments motivated by computational considerations:
    - Multilevel methods
    - Distributed optimization
    - Stochastic representation of a deterministic problem