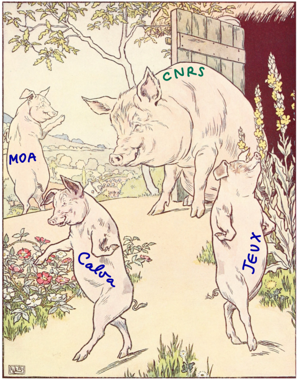
Jean-Baptiste Caillau
Université Côte d'Azur, CNRS, Inria, LJAD

Scientific day on optimisation
Paris, October 2024

Once upon a time...

... there were three little GDRs:

- GDR MOA (Mathématiques de l'Optimisation et Applications)
- GDR CalVa (Calcul des Variations et théorie géométrique de la mesure)
- GDR Jeux (Jeux : Modélisation Mathématique et Applications)



Once upon a time...

GdR MOA

GdR 3273 MOA

Mathématiques de l'Optimisation et Applications

cnrs

Accueil | Equipes | Activités | Mois de l'optimisation | Séminaire Français d'Optimisation | Historique | Liens | Contact

Accueil

Le GdR Mathématiques de l'Optimisation et Applications a pour vocation de regrouper un nombre important de chercheurs français, sur les aspects les plus divers de l'optimisation et de la programmation mathématique :

- optimisation numérique, algorithmes d'optimisation, pratique de l'optimisation numérique et développement logiciel
- analyse convexe et quasiconvexe
- analyse variationnelle et analyse non lisse
- calcul des variations
- commande optimale déterministe de systèmes de dimension finie ou infinie (dynamique de populations, équations aux dérivées partielles)
- inéquations variationnelles et problèmes de complémentarité
- systèmes dynamiques non-réguliers
- programmation stochastique et contrôle stochastique
- programmation robuste
- programmation semi-définie, programmation conique

et d'en développer les applications et les interactions, notamment dans les domaines suivants :

À l'affiche

- **Journées annuelles 2023 du GdR MOA**
  du 18 au 20 octobre 2023 midi, à Perpignan
- **Mini-cours**
  du 16 au 18 octobre 2023, à Perpignan

Bureau du GdR

Direction :

- Mounir Haddou, IRMAR - INSA, Rennes

Comité de direction :

- Nadia Oudjane, EDF R&D
- Serge Gratton, ENSEEIHT, Toulouse
- Florent Nacry, Université de Perpignan
- Edouard Pawels, IRIT Université de Toulouse 3

Comité scientifique international :

- René Henrion, Institut Weierstrass de Berlin
- Martine Labbé, Université Libre de Bruxelles
- Monique Laurent, Université de Tilburg, Hollande
- Claudia Sagastizabal, IMECC-Unicamp, Brésil
- Silvia Villa, Gênes, Italie

Once upon a time...

GDR CALVA

Calcul des variations et théorie géométrique de la mesure

PRÉSENTATION Thèmes Organisation Annonces

Le GDR "Calva" est un groupement de cherche du CNRS sur le thème du calcul des variations et de la théorie géométrique de la mesure. Il rassemble plus de 120 membres permanents répartis dans 38 institutions différences sur toute la france, avec une ouverture à l'internationnal. Il permet de fédérer la recherche française sur le thème du calcul des variations, d'organiser des rencontres et de procurer une aide financière pour les jeunes chercheurs (doctorants notamment).

Pour vous inscrire à la **liste de diffusion du GDR CALVA** vous pouvez le faire en utilsant ce lien.

Last update: 12/15/2023 10:40:07

Once upon a time...

Présentation

L'axe Jeux du Réseau thématique « Optimisation » réunit des chercheurs de différents domaines autour des thèmes transversaux que la théorie mathématique des jeux aborde. Il regroupe environ 180 chercheurs issus de 31 unités différentes, relevant des mathématiques, de l'informatique et de l'économie.

TA-DA

rt-optimisation.math.cnrs.fr

- 1 RT (F. Santambrogio), 3 axes : MOA (M. Haddou), CalVa (M. Goldman), Jeux (C. Rainer)
- Started 1st January 2024 (4 years), 50 Keuros/year
- 300+ researchers, 50+ labs, 800+ mailing list subscribers (`rt-optimisation@listes.math.cnrs.fr`)
- Mostly maths, also computer science + economics
- Funding events on optimisation

SMAI-MODE group on optimisation

Journées SMAI MODE 2024

27–29 Mar 2024
Lyon
Europe/Paris timezone

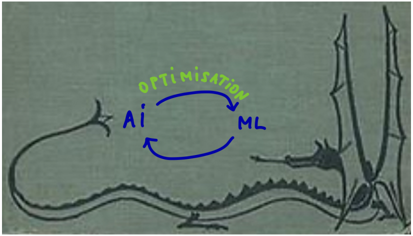
Enter your search term

- Accueil
- Dates clés
- Programme global
- Programme détaillé
- Registration
- Tarifs
- Prix Dodu et prix poster
- Informations pratiques
- Comités
- Partenaires
- Photos
- Affiche
- Mention d'information

Accueil

Two more things

- AI: there and back again (a journey into optimisation and control)
- Efficient optimisation: algorithms, (sparse) numerical linear algebra and Automatic Differentiation (AD)



ResNets as discretised linear control systems

After Agrachev (SISSA), Sarychev (Florence), Scagliotti (TUM) *et al*:

- Agrachev, A. A.; Sarychev, A. V. Control in the Spaces of Ensembles of Points. *SIAM J. Control Optim.* **58** (2020), no. 3, 1579–1596
- Agrachev, A. A.; Sarychev, A. V. Control on the Manifolds of Mappings with a View to the Deep Learning. *J. Dyn. Control Syst.* **28** (2021), 989–1008
- A Scagliotti. Deep Learning approximation of diffeomorphisms via linear-control systems. *MCRF* **13** (2023), no. 3, 1226-1257

ResNets are compositions of nonlinear mappings

$$\Phi = \Phi_M \circ \cdots \circ \Phi_1$$

where M is the depth of the neural network and where each layer is of the form (additional term *wrt.* non-residual networks)

$$\Phi_\ell(x) = x + \sigma(W_\ell + b_\ell).$$

ResNets as discretised linear control systems

For the approximation properties of such compositions see, *e.g.*,

- Yarotsky, D. Error bounds for approximations with deep ReLU networks. *Neural Networks* **94** (2017), 103-114

in the case of non-residual networks. This composition can also be interpretated as the explicit Euler discretisation of the *Neural ODE*

$$\dot{x}(t) = \sigma(W(t)x(t) + b(t)),$$

where W and b are now functions of time (continuum of layers), *controls*. Point of view developed, *e.g.*, in Tabuada & Gharesifard:

- Tabuada, P.; Gharesifard, B. Universal Approximation Power of Deep Neural Networks via Nonlinear Control Theory. arXiv:007.06007 (2020)

See also *constructive* approach for Lipschitz (ReLU-like) activation function in references below:

- Li, Q.; Lin, T; Shen, Z. Deep learning via dynamical systems: An approximation perspective. *J. Eur. Math. Soc.* **25** (2023), 1671–1709
- Ruiz-Balet, D.; Zuazua, E. Neural ode control for classification, approximation and transport. *SIAM Review* **65** (2022), no. 3, 735-773

ResNets as discretised linear control systems

Alternative point of view. Nonlinear in the data x but *linear* in the parameters:

$$\Phi_\ell(x) = x + G(x)u_\ell.$$

Composition now interpretated as the discretisation of

$$\dot{x}(t) = G(x(t))u(t) = \sum_{i=1}^{m} u_i(t)F_i(x(t))$$

where the *smooth* vector fields $F_1, \ldots, F_m$ are the columns of the nonlinear function G.

Ability to learn data (finite or *continuum*) = controllability properties of the control system for *ensembles*.

Ensembles

Definition. For Θ compact subset of $\mathbf{R}^n$ (set of possibly infinite indices of the data), define an ensemble as a continuous *injective* map from Θ to $\mathbf{R}^n$. Denote $\mathscr{E}_\Theta$ the set of ensembles.

Example. For $|\Theta| = N < \infty$ finite, ensemble $=$ open subset of $(\mathbf{R}^n)^N$ made of pairwise distinct vectors: $(x_1, \dots, x_N) \in (\mathbf{R}^n)^{(N)}$.

ICERM

Home | Programs | Your Visit | Videos | About | Support ICERM

Interacting Particle Systems: Analysis, Control, Learning and Computation

May 6 - 10, 2024

Workshop Overview | Workshop Participants | Workshop Schedule | Reimburse Me

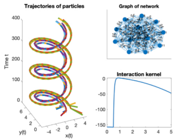
Organizing Committee

- Jose Carrillo
  University of Oxford
- Fei Lu
  Johns Hopkins University
- Katy Craig
  UC Santa Barbara
- Mauro Maggioni
  Johns Hopkins University
- Massimo Fornasier
  Technical University of Munich
- Kavita Ramanan
  Brown University

Abstract

Systems of interacting particles or agents are studied across many scientific disciplines. They are used as effective models in a wide variety of sciences and applications, to represent the dynamics of particles in physics, cells in biology, people in urban mobility studies, but also, more abstractly in the context of mathematics, as sample particles in Monte Carlo simulations or parameters of neural networks in machine learning.

This workshop aims at bringing together researchers in analysis, computation, inference, control and applications, to facilitate cross-fertilization and collaborations.



Exact controllability

For an admissible control u in $L^2([0,1],\mathbf{R}^m)$ (+ growth conditions on vector fields), define the time 1 flow Φ_u of the controlled system

$$\dot{x}(t) = \sum_{i=1}^{m} u_i(t) F_i(x(t)) \qquad (1)$$

mapping an initial condition x_0 to $\Phi_u(x_0) := x(1,u,x_0)$.

Definition. The control system (1) is said to be controllable on $\mathscr{E}_\Theta$ if, for any ensembles γ_0, γ_f, there exists an admissible control u *s.t.*

$$\Phi_u \circ \gamma_0 = \gamma_f.$$

Example. For $|\Theta| = N$ finite and any ensembles $(x_1^0,\dots,x_N^0)$, $(x_1^f,\dots,x_N^f)$ in $(\mathbf{R}^n)^{(N)}$, controllability means that there exists an admissible control u *s.t.*

$$\Phi_u(x_j^0) = x_j^f, \quad j = 1,\dots,N.$$

Theorem. (Agrachev-Sarychev'2020) For $m \geq 2$ controls, any $N \geq 1$ and k sufficiently large, the set of vector fields $F_1,\dots,F_m$ *s.t.* controllability holds on $\mathscr{E}_\Theta$ with $|\Theta| = N$, is residual in $\mathscr{C}^k(\mathbf{R}^n)$.

Approximate reachability

Definition. Let γ_0 and γ_f be two ensembles in $\mathscr{E}_\Theta$. Then γ_f is said to be $\mathscr{C}^0$-*approximately reachable* from γ_0 by control system (1) if, for any $\varepsilon > 0$, there exists an admissible control u *s.t.*

$$\sup_{\theta \in \Theta} |\Phi_u(\gamma_0(\theta)) - \gamma_f(\theta)| \leq \varepsilon.$$

Remark. As a flow, any such Φ_u is diffeotopic to identity since

$$x(0, u, \cdot) = \mathrm{Id}, \quad x(1, u, \cdot) = \Phi_u.$$

So if γ_f is reachable from γ_0, the two ensembles must be diffeotopic.

Approximate reachability

Definition. The family $F_1, \ldots, F_m$ satisfies the *(Lie algebra) strong approximation property* if there exists $k \geq 1$ *s.t.*, for any $\mathscr{C}^k$ vector field X and for any compact $K \subset \mathbf{R}^n$, there is $\delta > 0$ *s.t.*

$$\inf\{\max_{x \in K} |X(x) - Y(x)| \text{ with } Y \in \mathrm{Lie}\{F_1, \ldots, F_m\},\ \|Y\|_{1,K} \leq \delta\} = 0.$$

Remark. The strong approximation property implies that, for every $N \geq 1$, the folds of $F_1, \ldots, F_m$ are bracket generating on $(\mathbf{R}^n)^{(N)}$. So (exact) controllability holds for finite sets Θ.

Example. The family below satisfies the strong approximation property ($n \geq 2$):

$$F_i(x) = \frac{\partial}{\partial x_i} \quad G_i(x) = e^{-|x|^2} \frac{\partial}{\partial x_i} \quad i = 1, \ldots, n.$$

Theorem. (Agrachev-Sarychev'2021) Let γ_0 and γ_f be two *diffeotopic* ensembles in $\mathscr{E}_\Theta$. Under the strong approximation property, γ_f is $\mathscr{C}^0$*-approximately reachable* from γ_0 by control system (1).

A glimpse of AD

- Good optimisation needs good gradients (+ Hessians)
- More or less built-in with modern languages / optimisation modellers
- Work at LLVM level (Enzyme)
- Sparse AD: graph coloring techniques

8th International Conference on Algorithmic Differentiation

SEPTEMBER 16–19, 2024, CHICAGO, ILLINOIS, U.S.

Home Call for Papers Submissions Important Dates Attendance Program Committees

Image by Sea Cow, CC BY-SA 4.0 DEED

AD 2024

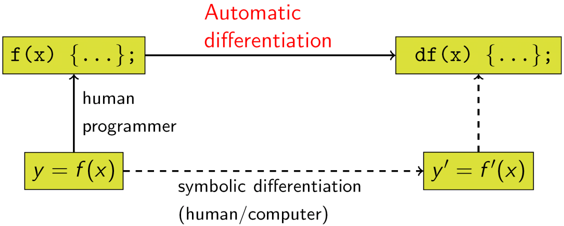
The International Conference on Algorithmic Differentiation 2024 (AD 2024) provides a top-tier forum for presenting original research on the theoretical foundations, implementation, and application of algorithmic differentiation, also known as automatic differentiation. AD 2024 in Chicago, United States, will be the eighth edition in this series of international AD conferences whose record of previous editions is available at www.autodiff.org.

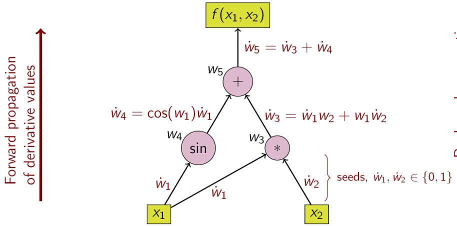
Invited Speakers
The following list is arranged in alphabetical order.

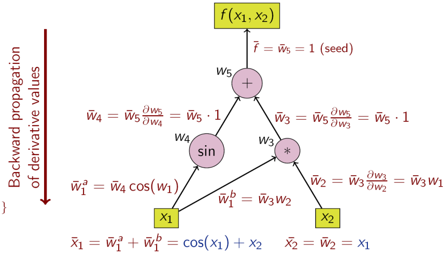
- Anshu Dubey, Argonne National Laboratory, USA, on Software Design for Sustainability and Differentiability
- Ralf Giering, FastOpt, Germany, on Challenges of Adjoint Code Construction
- Nathan Killoran, Xanadu, Canada, on Bringing Automatic Differentiation to Quantum Computing

A glimpse of AD

$$y = f(x_1, x_2) = \sin x_1 + x_1 x_2$$







Source: Wikipedia on AD

Optimisation modellers: finite dim

JUMP

Search docs (Ctrl + /)

Installation Guide

Tutorials

Getting started

Transitioning

Linear programs

Nonlinear programs

Introduction

Simple examples

User-defined operators with vector

```
julia> using JuMP, Ipopt

julia> function solve_constrained_least_squares_regression(A::Matrix, b::Vector)
           m, n = size(A)
           model = Model(Ipopt.Optimizer)
           set_silent(model)
           @variable(model, x[1:n])
           @variable(model, residuals[1:m])
           @constraint(model, residuals == A * x - b)
           @constraint(model, sum(x) == 1)
           @objective(model, Min, sum(residuals.^2))
           optimize!(model)
           return value.(x)
       end
solve_constrained_least_squares_regression (generic function with 1 method)

julia> A, b = rand(10, 3), rand(10);

julia> x = solve_constrained_least_squares_regression(A, b)
3-element Vector{Float64}:
 0.4137624719002825
 0.09707679853084578
 0.48916072956887174
```

Optimisation modellers: finite dim

ExaModels.jl

Search docs (Ctrl + /)

- **Introduction**
- **Mathematical Abstraction**
- **Tutorial**
  - Getting Started
  - Performance Tips
  - Accelerations
  - Developing Extensions
  - Example: Quadrotor
  - Example: Distillation Column
  - Example: Optimal Power Flow
- **JuMP Interface (experimental)**
- **API Manual**

```julia
    c = ExaCore(; backend = backend)

    x = variable(c, 1:N+1, 1:n)
    u = variable(c, 1:N, 1:p)

    constraint(c, x[1, i] - x0 for (i, x0) in x0s)
    constraint(c, -x[i+1, 1] + x[i, 1] + (x[i, 2]) * dt for i = 1:N)
    constraint(
        c,
        -x[i+1, 2] +
        x[i, 2] +
        (
            u[i, 1] * cos(x[i, 7]) * sin(x[i, 8]) * cos(x[i, 9]) +
            u[i, 1] * sin(x[i, 7]) * sin(x[i, 9])
        ) * dt for i = 1:N
    )
    constraint(c, -x[i+1, 3] + x[i, 3] + (x[i, 4]) * dt for i = 1:N)
    constraint(
        c,
        -x[i+1, 4] +
        x[i, 4] +
        (
            u[i, 1] * cos(x[i, 7]) * sin(x[i, 8]) * sin(x[i, 9]) -
            u[i, 1] * sin(x[i, 7]) * cos(x[i, 9])
        ) * dt for i = 1:N
    )
    constraint(c, -x[i+1, 5] + x[i, 5] + (x[i, 6]) * dt for i = 1:N)
    constraint(
        c,
        -x[i+1, 6] + x[i, 6] + (u[i, 1] * cos(x[i, 7]) * cos(x[i, 8]) - 9.8) * dt for
        i = 1:N
    )
```

Optimisation modellers: infinite dim

OptimalControl.jl

Search docs (Ctrl + /)

Optimal control problem

Let us define the problem

```
ocp = @def begin

    tf ∈ R,          variable
    t ∈ [ 0, tf ],   time
    x = (q, v) ∈ R², state
    u ∈ R,           control

    tf ≥ 0
    -1 ≤ u(t) ≤ 1

    q(0)  == 1
    v(0)  == 2
    q(tf) == 0
    v(tf) == 0

    -5 ≤ q(t) ≤ 5,          (1)
    -3 ≤ v(t) ≤ 3,          (2)

    ẋ(t) == [ v(t), u(t) ]

    tf → min

end
```

PEPR IA

PDE-AI

MEMBERS EVENTS LOCAL EVENTS JOB OFFERS PUBLICATIONS

PEPR PDE-AI

Partial Differential Equations for Artificial Intelligence: numerical analysis, optimal control and optimal transport

PDE-AI is a PEPR project funded by the ANR, which gathers ten major French institutions involved in developing the mathematical analysis of AI, the study of optimization in machine learning, as well as in developing machine learning for numerical analysis and scientific computing. The institutions are Univ. Paris-Dauphine (PSL), Univ. Paris-Cité, Sorbonne Univ., Univ. Paris-Saclay, Univ. Toulouse, Univ. Lyon (CNRS), Univ. Bordeaux, Univ. Côte d'Azur, CREST (ENSAE/Institut Polytechnique de Paris) and Univ. Strasbourg. The project started in September 2023 and will last until 31 August 2027. The project is supported by the "France 2030" programme.

- pde-ai.math.cnrs.fr