Inverse Problems and Environment:

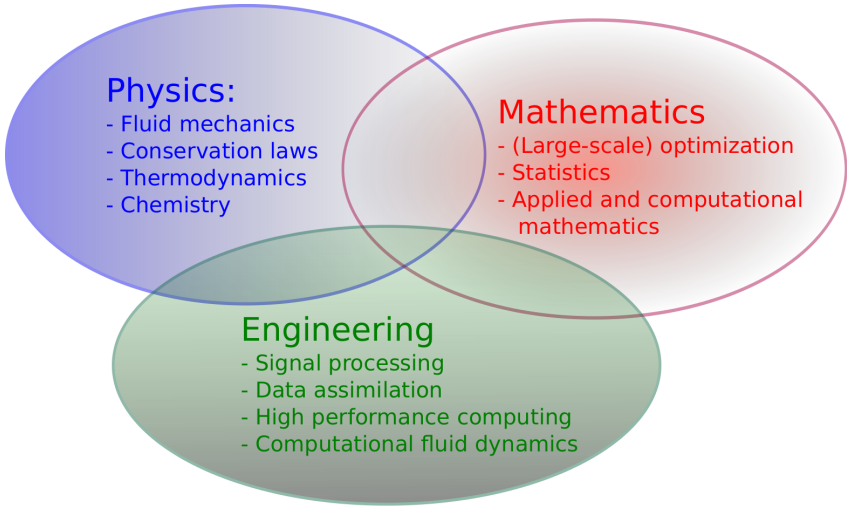
some new tools for atmospheric studies

Overview

Emmanuel WITRANT
emmanuel.witrant@ujf-grenoble.fr

PhD school “MATH et COMPLEX”, Department of Mathematics
UNamur, Belgium, March 10^{th}, 2014.

Inverse modeling: an interdisciplinary approach



This course (mostly): Conservation laws, Optimization/Computational mathematics, Data assimilation/Computational fluid dynamics.

Environment and atmospheric studies

- NASA video: progression of changing global surface temperatures anomalies from 1880 through 2012.
  `http://svs.gsfc.nasa.gov/vis/a000000/a004000/a004030/`
- Earth system = complex interconnected systems involving, e.g. oceans, polar ice sheets and atmosphere.
- Numerous proxies, e.g. temperature measurements in oceans, chemical measurements in atmosphere and firns/ice cores, radar measurements in ice sheet.

Classes overview (day 1)

Les.	Topic
1	**Introduction and basic modeling concepts (E. Witrant, 1h)** *Overview of inverse problems; Conservation laws and modeling with partial differential equations; Some computation issues.*
2	**The earth system: from proxies to decision making (S. Houweling, 2h)** *The climate system; Atmospheric transport models and feedback mechanisms; Measurements and proxies; The role of Methane and Carbon dioxyde; Assessing climate changes.*
3	**Optimization methods: problem formulation (E. Witrant, 1h30)** *State variables, Inverse problem formulation, Performance measure, Regularization.*
4	**Optimization methods: analytical and numerical solutions (E. Witrant, 1h30)** *Calculus of variations, Optimizing linear systems; Nonlinear programming and gradient descent; Numerical methods.*

Classes overview (day 2)

Les.	Topic
5	**Chemo-hydrodynamic patterns and instabilities (Anne De Wit, 1h30)** *Hydrodynamic instabilities in reactive systems; Effect of chemical reactions on density and viscous fingering; Convective instabilities in sea ice and during CO2 sequestration in porous media.*
6	**Inverse modeling from ice cores (E. Witrant, 1h30)** *Modeling heterogeneous transport of trace gas in a 1-D porous medium; Characterizing transport in ice cores using PDE optimization; Atmospheric reconstruction of trace gas from linear optimization and sparse data.*
7	**Inverse modeling of atmospheric emissions (M. Krol, 2h30)** *Measurements calibration and correlation; Handling uncertainties; Cost function and regularization; Qualitative analysis and methods revision, sensitivity analysis.* Hands-on simulation experience.

Introduction and basic modeling
E. Witrant

Classes overview (day 3)

Les.	Topic
8	**Optimization techniques in data assimilation for oceanography and weather forecasting (A. Sartenaer, 1h30)** *The variational approach and its 4D-Var formulation; Preconditioning techniques; Derivative-free approaches; Multilevel optimization.*
9	**Glaciers, ice sheets and ice shelves (F. Pattyn, 2h)** *Ice-sheet modelling ; Common approximations of the Stokes equations; Numerical solutions; Initialization problems and boundary condition estimates using inverse modeling..*
10	**Analysing and simulating large-scale systems (S. Vandewalle, 2h)** *Methods from numerical analysis for complex (possibly nonlinear) systems described by partial differential equations, multigrid approaches.*

Inverse Problems and Environment:

some new tools for atmospheric studies

Lesson 1: Introduction to inverse problems and basic modeling concepts

Emmanuel WITRANT
emmanuel.witrant@ujf-grenoble.fr

PhD school "MATH et COMPLEX", Department of Mathematics
UNamur, Belgium, March 10^{th}, 2014.

Introduction and basic modeling
E. Witrant
Inverse problems
Mappings
Solutions
Inversion scheme
Statistical methods
Approximation methods
Kalman filtering
Optimization
Conservation laws
Convection-diffusion
Euler and Navier-Stokes
Computation issues
From distributed to lumped dynamics
Green's function
Conclusions

Introduction and basic modeling
E. Witrant
Inverse problems
Mappings
Solutions
Inversion scheme
Statistical methods
Approximation methods
Kalman filtering
Optimization
Conservation laws
Convection-diffusion
Euler and Navier-Stokes
Computation issues
From distributed to lumped dynamics
Green's function
Conclusions

Inverse problems (IP)

Overview

- Introduced by Viktor Ambartsumian (Soviet-Armenian physicist, 1908-1996) for the Sturm-Liouville pb in 1929

Wikipedia "An IP is a general framework that is used to convert observed measurements into information about a physical object or system"

- Widely used, e.g. in computer vision, natural language processing, machine learning, statistics, statistical inference, geophysics, medical imaging (such as computed axial tomography and EEG/ERP), remote sensing, ocean acoustic tomography, nondestructive testing, astronomy, physics, environment...
- Examples:

Physical system	**Governing equations**	**Physical quantity**	**Observed data**
Earth's grav. field	Newton's law of gravity	Density	Gravitational field
Earth's mag. field	Maxwell's equations	Magnetic susceptibility	Magnetic field
Seismic waves	Wave equation	Wave-speed (density)	Particle velocity

Introduction and basic modeling
E. Witrant
Inverse problems
Mappings
Solutions
Inversion scheme
Statistical methods
Approximation methods
Kalman filtering
Optimization
Conservation laws
Convection-diffusion
Euler and Navier-Stokes
Computation issues
From distributed to lumped dynamics
Green's function
Conclusions

Definition

- Consider a particular (physical) model structure $\mathcal{M}$ parameterized using $p \in \mathcal{D}_{\mathcal{M}} \subset \mathbb{R}^{n_p}$:
  $\mathcal{M}^* = \{\mathcal{M}(p) \,|\, p \in \mathcal{D}_{\mathcal{M}}\}$:
  - knowing $\mathcal{M}$ and p, we can predict the observed data $\hat{y}$
    $\Rightarrow$ forward model, e.g. $\mathcal{M} : p \mapsto \hat{y}$ and $\hat{y} = M(p)$
  - knowing $\mathcal{M}$ and observed data y, we can estimate the unknown parameter $\hat{p}$
    $\Rightarrow$ inverse model, e.g. $\mathcal{M}^{-1} : y \mapsto \hat{p}$ and $\hat{p} = M^{-1}(y)$
- e.g. inverse solution for the simplest algebraic case where $\mathcal{M} : y = Mp$ where $p \in \mathbb{R}^{n_p}$, $y \in \mathbb{R}^{n_d}$ and M a known matrix $\in \mathbb{R}^{n_d \times n_p}$, for $n_p = n_d$, $n_p > n_d$ and $n_p < n_d$?

Mappings

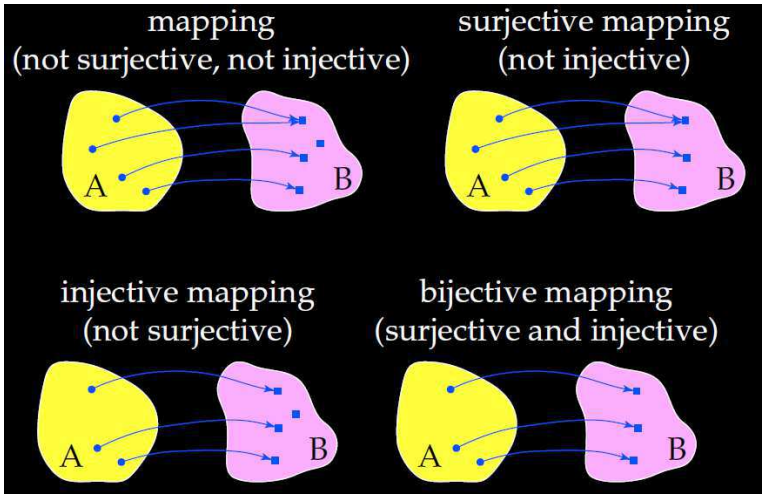


Figure courtesy of A. Tarantola, Course on Inverse Problems 2007

→ Expressed in terms of under (e.g. $n_p > n_d$) or over (e.g. $n_d > n_p$) constrained IP

Introduction and basic modeling
E. Witrant
Inverse problems
Mappings
Solutions
Inversion scheme
Statistical methods
Approximation methods
Kalman filtering
Optimization
Conservation laws
Convection-diffusion
Euler and Navier-Stokes
Computation issues
From distributed to lumped dynamics
Green's function
Conclusions

- "Far better an approximate answer to the right question, which is often vague,
  than an exact answer to the wrong question,
  which can always be made precise."
  John W. Tukey, 1962

→ Handle the under/over contrained issues in the optimization problem formulation, expressing the trade-offs between conflicting objectives

- Physical problems are always underconstrained: continuous (infinite dimensional) parameter estimated from discrete (sampled) measurements

Introduction and basic modeling
E. Witrant
Inverse problems
Mappings
Solutions
Inversion scheme
Statistical methods
Approximation methods
Kalman filtering
Optimization
Conservation laws
Convection-diffusion
Euler and Navier-Stokes
Computation issues
From distributed to lumped dynamics
Green's function
Conclusions

Solutions and well-posedness

Jacques S. Hadamard
(French mathematician
1865-1963)

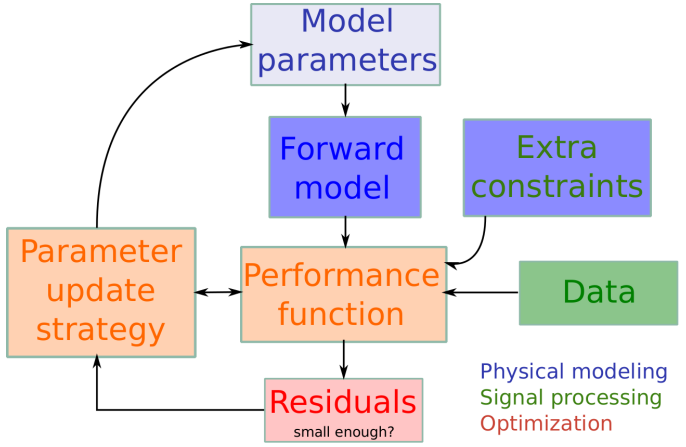
For a well-posed problem [J. Hadamard]:

- a solution exists
- the solution is unique
- the solution's behavior changes continuously with the initial conditions

Inverse problems are typically ill-posed: need for extra constraints (e.g. regularity of the solution, coherence between model and data variability, etc.)

Introduction and basic modeling
E. Witrant
Inverse problems
Mappings
Solutions
Inversion scheme
Statistical methods
Approximation methods
Kalman filtering
Optimization
Conservation laws
Convection-diffusion
Euler and Navier-Stokes
Computation issues
From distributed to lumped dynamics
Green's function
Conclusions

A general inversion scheme



Main inversion strategies

Statistical methods

- Uncertainties/prior information modeled as random → statistical characterization
- Models complex uncertainties from simulation model, data measurements and prior information
- Aim of the methods: achieve statistical description with an acceptable computation cost
- Require a stochastic model of the data and constraints on the possible state of the world
- Tools:
  - parameters: numerical properties of state
  - estimators: quantities computed from data without knowing the state
  - risk functions: quantify the expected 'cost' to compare estimators

Introduction and basic modeling
E. Witrant
Inverse problems
Mappings
Solutions
Inversion scheme
Statistical methods
Approximation methods
Kalman filtering
Optimization
Conservation laws
Convection-diffusion
Euler and Navier-Stokes
Computation issues
From distributed to lumped dynamics
Green's function
Conclusions

Statistical methods (2)

- Frequentist methods:
  - frequency interpretation of probability: any given experiment can be considered as one of an infinite sequence of possible repetitions of the same experiment, each capable of producing statistically independent results
  - parameters: fixed but unknown values, not random var.
  - result = true/false, or confidence interval
  - e.g. minmax estimation: minimize the maximum risk over all states satisfying the constraints
- Bayesian inference:
  - min. the expected risk when the state is drawn at random according to the constraints
  - Bayes' rule (relates odds of event A_1 to event A_2 before and after conditioning on another event B) is used to update the probability estimate for a hypothesis as additional evidence is acquired
  - model parameters and constraints as prior probability distribution
  - result = probability distribution

Introduction and basic modeling
E. Witrant
Inverse problems
Mappings
Solutions
Inversion scheme
Statistical methods
Approximation methods
Kalman filtering
Optimization
Conservation laws
Convection-diffusion
Euler and Navier-Stokes
Computation issues
From distributed to lumped dynamics
Green's function
Conclusions

Approximation methods

- Many (thousands of) iterations of the forward model are often necessary
- Critical for large-scale (e.g. PDE) problems → reduce computation cost
- Advanced linear solver and preconditioning techniques + parallelization: often not sufficient

⇒ Replace the forward model by an inexpensive surrogate, e.g. ↘ input space and improve sampling, mutligrid approaches

- Then combined with statistical or deterministic optimization strategies

Introduction and basic modeling
E. Witrant
Inverse problems
Mappings
Solutions
Inversion scheme
Statistical methods
Approximation methods
Kalman filtering
Optimization
Conservation laws
Convection-diffusion
Euler and Navier-Stokes
Computation issues
From distributed to lumped dynamics
Green's function
Conclusions

Kalman filtering (Rudolf Emil Kálmán, Hungarian EE engineer 1930-)

- Ensemble Kalman filters:
  - recursive filter suitable for a large number of parameters (e.g. discretized PDEs)
  - seek the solution in the space spanned by a collection of ensembles
  - compared to the classical KF (1960): replace covariance matrix by samples covariance
  - supposes Gaussian probability distributions
- Assume that the variability of the parameters can be well approximated by a small number of nodes
- Causal: use only the data up to time t to estimate $p(t)$ → family of identification/estimation techniques

Optimization

- Formulate the inverse problem as an optimization problem
- Based on (large-scale) deterministic optimization methods
- Full use of the physical knowledge of the system
- Reduce the statistical flexibility
- Design according to the model architecture
- May provide qualitative, analytical results → new insights on the model property

Conservation laws

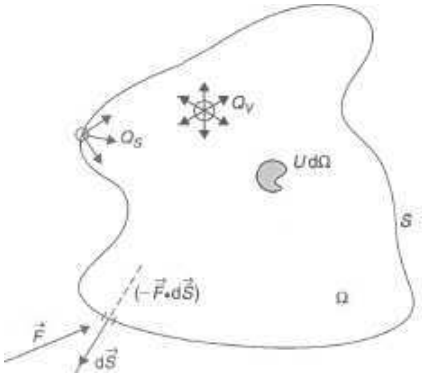
General form

- Conservation: the variation of a conserved (intensive) flow quantity U in a given volume results from internal sources and the quantity, the *flux*, crossing the boundary
- Fluxes and sources depend on space-time coordinates, + on the fluid motion
- Not all flow quantities obey conservation laws. Fluid flows fully described by the conservation of
  1. mass
  2. momentum (3-D vector)
  3. energy

  ⇒ 5 equations
- Other quantities can be used but will not take the form of a conservation law

Scalar conservation law

Consider:
- a scalar quantity per unit volume U,
- an arbitrary volume Ω fixed in space (control volume) bounded by
- a closed surface S (control surface) crossed by the fluid flow



- Total amount of U inside Ω: $\int_\Omega U d\Omega$ with variation per unit time $\frac{\partial}{\partial t}\int_\Omega U d\Omega$
- Flux = amount of U crossing S per unit time: $F_n dS = \vec{F}\cdot d\vec{S}$ with $d\vec{S}$ outward normal, and net total contribution $-\oint_S \vec{F}\cdot d\vec{S}$ ($\vec{F} > 0$ when entering the domain)
- Contribution of volume and surface sources: $\int_\Omega Q_V d\Omega + \oint_S \vec{Q}_S \cdot d\vec{S}$

Scalar conservation law (2)

Provides the integral conservation form for quantity U:

$$\frac{\partial}{\partial t}\int_\Omega U d\Omega + \oint_S \vec{F}\cdot d\vec{S} = \int_\Omega Q_V d\Omega + \oint_S \vec{Q}_S \cdot d\vec{S}$$

- valid ∀ fixed S and Ω, and any point in flow domain
- internal variation of U depends only of fluxes through S, not inside
- no derivative/gradient of F: may be discontinuous and admit shock waves

⇒ relate to conservative numerical scheme at the discrete level (e.g. conserve mass)

Differential form of a conservation law

Obtained using Gauss' theorem $\oint_S \vec{F} \cdot d\vec{S} = \int_\Omega \vec{\nabla} \cdot \vec{F} d\Omega$ as:

$$\frac{\partial U}{\partial t} + \vec{\nabla} \cdot \vec{F} = Q_V + \vec{\nabla} \cdot \vec{Q}_S \Leftrightarrow \frac{\partial U}{\partial t} + \vec{\nabla} \cdot (\vec{F} - \vec{Q}_S) = Q_V$$

- the effective flux $(\vec{F} - \vec{Q}_S)$ appear exclusively under the gradient operator $\Rightarrow$ way to recognize conservation laws
- more restrictive than the integral form as the flux has to be differentiable (excludes shocks)
- fluxes and source definition provided by the quantity U considered

Introduction and basic modeling
E. Witrant
Inverse problems
Mappings
Solutions
Inversion scheme
Statistical methods
Approximation methods
Kalman filtering
Optimization
Conservation laws
Convection-diffusion
Euler and Navier-Stokes
Computation issues
From distributed to lumped dynamics
Green's function
Conclusions

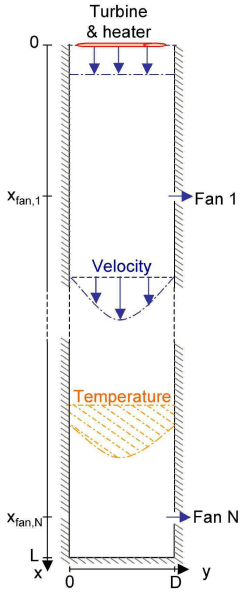
Convection-diffusion form of a convection law

Flux = convective transport + molecular agitation (even at rest)

- Convective flux:
  - amount of U carried away or transported by the flow (velocity $\vec{v}$): $\vec{F}_C = U\vec{v}$
  - for fluid density $U = \rho$, local flux through $d\vec{S}$ is the local mass flow rate: $\rho\vec{v} \cdot d\vec{S} = d\vec{m}$ (kg/s)
  - for $U = \rho u$ (u the quantity per unit mass), $\vec{F}_C \cdot d\vec{S} = \rho u \vec{v} \cdot d\vec{S} = u d\vec{m}$
- Diffusive flux:
  - macroscopic effect of molecular thermal agitation
  - from high to low concentration, in all directions, proportional to the concentration difference
  - Fick's law: $\vec{F}_D = -\kappa\rho\vec{\nabla}u$, where κ is the diffusion coefficient (m^2/s)
- Provides the transport equation:
  $\frac{\partial \rho u}{\partial t} + \vec{\nabla} \cdot (\rho \vec{v} u) = \vec{\nabla} \cdot (\kappa\rho\vec{\nabla}u) + Q_V + \vec{\nabla} \cdot \vec{Q}_S$
  $\Rightarrow$ *Backbone of all mathematical modeling of fluid flow phenomena*

Introduction and basic modeling
E. Witrant
Inverse problems
Mappings
Solutions
Inversion scheme
Statistical methods
Approximation methods
Kalman filtering
Optimization
Conservation laws
Convection-diffusion
Euler and Navier-Stokes
Computation issues
From distributed to lumped dynamics
Green's function
Conclusions

Euler and Navier-Stokes equations



- From the conservation of mass, momentum and energy:

$$\frac{\partial}{\partial t}\begin{bmatrix} \rho \\ \rho\vec{v} \\ \rho E \end{bmatrix} + \nabla \cdot \begin{bmatrix} \rho\vec{v} \\ \rho\vec{v}^T \otimes \vec{v} + p\mathbf{I} - \tau \\ \rho\vec{v}H - \tau \cdot \vec{v} - k\nabla T \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \dot{q} \end{bmatrix},$$

with shear stress (Navier-Stokes only)

$$\begin{bmatrix} \tau_{xx} \\ \tau_{xy} \\ \tau_{yy} \end{bmatrix} = \begin{bmatrix} \lambda \\ \mu \\ \lambda \end{bmatrix} (\nabla \cdot \vec{v}) + 2\mu \begin{bmatrix} u_x \\ 0 \\ v_y \end{bmatrix}$$

and viscosity [*Stokes* & *Sutherland*]

$$\lambda = -\frac{2}{3}\mu \quad \text{and} \quad \frac{\mu}{\mu_{sl}} = \left(\frac{T}{T_{sl}}\right)^{3/2} \frac{T_{sl} + 110}{T + 110}.$$

- Discrete boundary conditions (potential numerical instabilities).

Introduction and basic modeling
E. Witrant
Inverse problems
Mappings
Solutions
Inversion scheme
Statistical methods
Approximation methods
Kalman filtering
Optimization
Conservation laws
Convection-diffusion
Euler and Navier-Stokes
Computation issues
From distributed to lumped dynamics
Green's function
Conclusions

Some computation issues

- Need for a computable (i.e. discretized) model
- Wide family of discretization strategies
- Main issues: complexity, accuracy, stability, width (number of surrounding points), mesh definition
- E.g. using Taylor's series:

$$f(x + \Delta x, t) = f(x, t) + \Delta x f'(x, t) + \frac{\Delta x^2}{2} f''(x, t) + O(3)$$

implies at time $j\delta t$:

$$f_{i+1,j} = f_{i,j} + \delta x_{i+1} f'(x_i, t) + \frac{\delta x_{i+1}^2}{2} f''(x_i, t) + O(3)$$

$$f_{i-1,j} = f_{i,j} - \delta x_i f'(x_i, t) + \frac{\delta x_i^2}{2} f''(x_i, t) + O(3).$$

then f' is obtained from $f_{i+1,j} - f_{i-1,j}$ and f'' from $f_{i+1,j} + f_{i-1,j}$

- Example: gas diffusion in a tube [handout]

From distributed to lumped dynamics

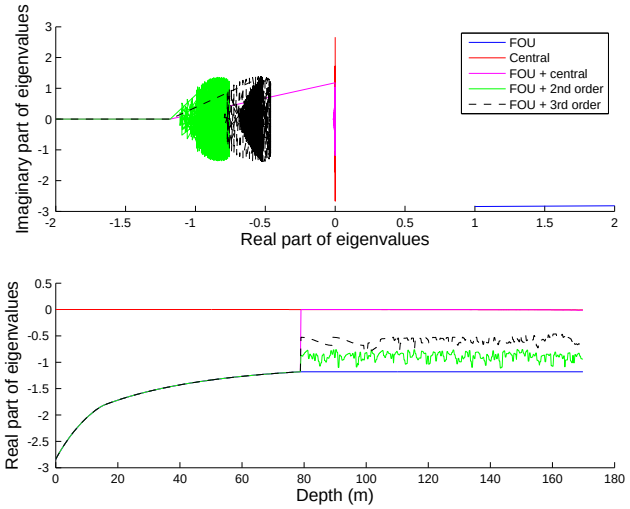
- Consider a quantity q transported in 1D by a flux $u = qv$ with a source term s ($t \in [0,\ T]$, $z \in [0,\ z_f]$):

$$\frac{\partial q}{\partial t} + \frac{\partial}{\partial z}[q\, v(z,t)] = s(z,t), \text{ with } \begin{cases} q(0,t) = 0 \\ q(x,0) = q_0(x) \end{cases}$$

  where $s(z,t) \neq 0$ for $z < z_1 < z_f$ and $s = 0$ for $z_1 < z < z_f$.
- Approximate $\partial[qv]/\partial z$, i.e. on uniform mesh:
  - backward difference: $(u_z)_i = \frac{u_i - u_{i-1}}{\Delta z} + \frac{\Delta z}{2}(u_{zz})_i$
  - central difference: $(u_z)_i = \frac{u_{i+1} - u_{i-1}}{2\Delta z_i} - \frac{\Delta z^2}{6}(u_{zzz})_i$
  - other second order: $(u_z)_i = \frac{u_{i+1} + 3u_i - 5u_{i-1} + u_{i-2}}{4\Delta z_i} + \frac{\Delta z^2}{12}(u_{zzz})_i - \frac{\Delta z^3}{8}(u_{zzzz})_i$
  - third order: $(u_z)_i = \frac{2u_{i+1} + 3u_i - 6u_{i-1} + u_{i-2}}{6\Delta z_i} - \frac{\Delta z^3}{12}(u_{zzzz})_i$
- Provides the computable lumped model: $dq/dt = Aq + s$
- The choice of the discretization scheme directly affects the definition of A and its eigenvalues distribution: need to check stability and precision!

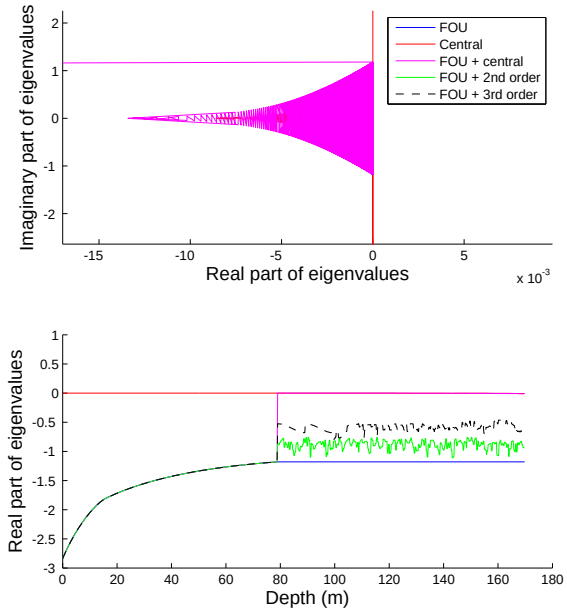
Introduction and basic modeling
E. Witrant
Inverse problems
Mappings
Solutions
Inversion scheme
Statistical methods
Approximation methods
Kalman filtering
Optimization
Conservation laws
Convection-diffusion
Euler and Navier-Stokes
Computation issues
From distributed to lumped dynamics
Green's function
Conclusions

E.g. stability: eigenvalues of A for CH_4 at NEEM with $dt \approx 1$ week



Introduction and basic modeling
E. Witrant
Inverse problems
Mappings
Solutions
Inversion scheme
Statistical methods
Approximation methods
Kalman filtering
Optimization
Conservation laws
Convection-diffusion
Euler and Navier-Stokes
Computation issues
From distributed to lumped dynamics
Green's function
Conclusions

E.g. eig(A) for CH_4 at NEEM with $dt \approx 1$ week, zoom



Introduction and basic modeling
E. Witrant
Inverse problems
Mappings
Solutions
Inversion scheme
Statistical methods
Approximation methods
Kalman filtering
Optimization
Conservation laws
Convection-diffusion
Euler and Navier-Stokes
Computation issues
From distributed to lumped dynamics
Green's function
Conclusions

Numerical system modeling using Green's function

- Used to solve inhomogeneous DE with specific boundary conditions
- For linear time-invariant (LTI) systems:

$$\dot{x}(t) = Ax(t) + Bu(t), \quad x(t_0) = x_0$$
$$y(t) = Cx(t)$$

  Green's function is equivalent to the impulse response (e.g. experimental)
- Provide a numerical I/O map for complex models, supposing a dominant LTI behavior
- If invertible mapping, then the inputs can be inferred from the measurements

I/O mapping for LTI with piecewise continuous input

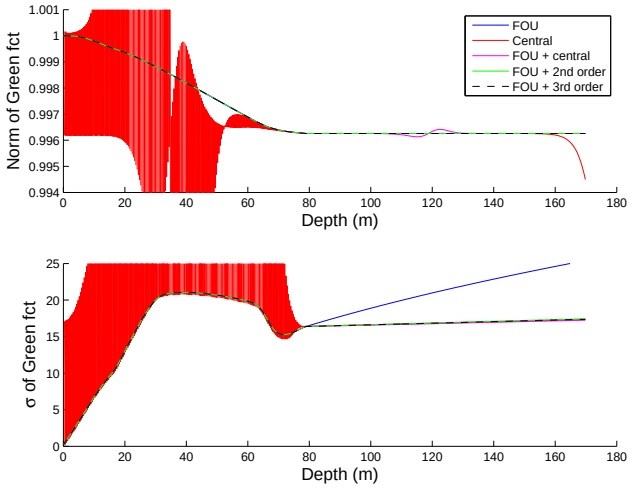
- Consider the previous LTI system
- The solution of the state-space equations writes as: $y(t) = Ce^{A(t-t_0)}x(t_0) + C\int_{t_0}^{t} e^{A(t-\tau)}Bu(\tau)d\tau$
- Considering a piecewise continuous inputs $u(t)$ for $t \in [t_0, t_f]$, the discretized version is:

$$\begin{aligned} y(t_k) &= Ce^{A(t_k-t_0)}x(t_0) + t_s C\sum_{i=0}^{k} e^{A(t_k-t_i)}Bu(t_i) \\ &= G_0(t_k)x(t_0) + G(t_k)U(t_k) \end{aligned}$$

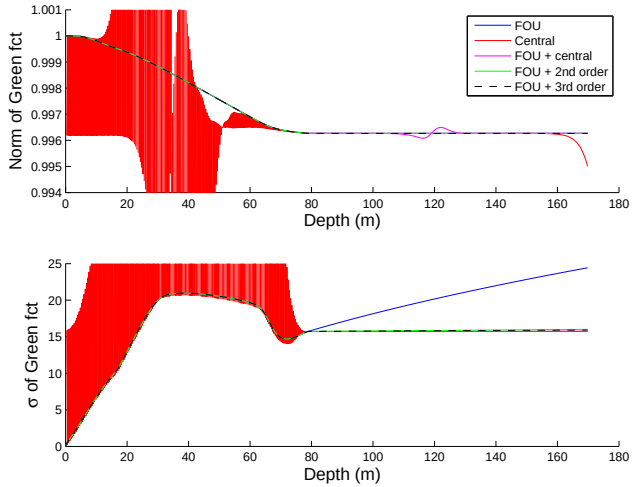
$$\text{where } \begin{cases} G_0(t_k) \doteq Ce^{A(t_k-t_0)} \in \mathbb{R}^{m\times n} \\ G(t_k) \doteq t_s[Ce^{A(t_k-t_0)}B,\ Ce^{A(t_k-t_1)}B, \ldots, CB] \\ U(t_k) \doteq [u(t_0), \ldots, u(t_k)]^T \in \mathbb{R}^k \end{cases}$$

- Note that $G(t_k) \in \mathbb{R}^{m\times k}$ corresponds to the Green's function or impulse response of the LTI system.

Precision evaluation: Green's function for CH_4 at NEEM with $dt = 1$ month



Green's function for CH_4 at NEEM with $dt \approx 1$ week



Conclusions

- Inverse problems:
  - formulate the performance evaluation while handling ill-posedness by adding constraints
  - determine a variation law for the parameters
  - use stochastic or deterministic approaches
- Models:
  - inferred from conservation laws
  - system of PDEs, possibly coupled, nonlinear, etc.
- Computation issues:
  - reduce to the final dimensional case
  - alternative simplified models

References

- L. Ljung, *System Identification: Theory for the User*, 2^{nd} Edition, Information and System Sciences, (Upper Saddle River, NJ: PTR Prentice Hall), 1999.
- L. Biegler *et al.* (Eds), *Large-Scale Inverse Problems and Quantification of Uncertainty*, Wiley series in computational statistics, 2011.
- A. Tarantola, *Inverse Problem Theory*, SIAM, 2005.

http://www.ipgp.fr/~tarantola/Files/Professional/Teaching/CalTech/

- Anderson, J.: Fundamentals of Aerodynamics, McGraw-Hill Companies, 1991.
- C. Hirsch, Numerical Computation of Internal & External Flows: the Fundamentals of Computational Fluid Dynamics, 2nd ed. Butterworth-Heinemann (Elsevier), 2007.

Inverse Problems and Environment:

some new tools for atmospheric studies

Lesson 3 - Optimization methods: problem formulation

Emmanuel WITRANT
emmanuel.witrant@ujf-grenoble.fr

PhD school "MATH et COMPLEX", Department of Mathematics UNamur, Belgium, March 10th, 2014.

Optimization: problem formulation
E. Witrant

Optimization: problem formulation
E. Witrant

Motivation

- Given a forward model, solve the inverse problem for a particular set of parameters
- Inverse problem ⇔ Optimization problem
- Main issues:
  - how to formulate the optimization problem with respect to the data set?
  - how to add constraints on the parameters to select among the infinite possible solutions (curse of ill-posedness)?
- Focus on ordinary differential equations (e.g. discretization)
- The same principles apply to the infinite dimensional (PDE) case, provided some extra technical issues

Optimization: problem formulation
E. Witrant

Background on state variables representation of systems

- Considering a dynamical system with input u, internal state x and output y, we establish a general description of the input/output map, called the state-space representation.
- Classification of systems depenting on their nonlinearities and time-variations:

State dynamics	**Output**	**Main property**
$\dot{x} = f(x,u,t)$,	$y = g(x,u,t)$,	NL, TV
$\dot{x} = f(x,u)$,	$y = g(x,u)$,	NL, TI
$\dot{x} = A(t)x + B(t)u$,	$y = C(t)x + D(t)u$,	LTV
$\dot{x} = Ax + Bu$,	$y = Cx + Du$,	LTI

Nonlinear (NL), time-varying (TV), time-invariant (TI), linear time-varying (LTV), linear time-invariant (TI)

Solution of the state equations for linear systems

- LTV: $x(t) = \phi(t, t_0)x(t_0) + \int_{t_0}^{t} \phi(t, \tau)B(\tau)u(\tau)d\tau$
  where $\phi(\cdot, \cdot)$ is the state transition matrix.
- If LTI and $t_0 = 0$, equivalent forms:

$$\begin{aligned} x(t) &= \mathcal{L}^{-1}\{[sI - A]^{-1}x_0 + [sI - A]^{-1}BU(s)\} \\ &= \mathcal{L}^{-1}\{\Phi(s)x_0 + H(s)U(s)\} \\ &= e^{At}x_0 + e^{At}\int_0^t e^{-A\tau}Bu(\tau)d\tau \end{aligned}$$

with $e^{At} \doteq I + At + \frac{1}{2!}A^2t^2 + \ldots + + \frac{1}{k!}A^kt^k + \ldots$
Hence $e^{At} = \mathcal{L}^{-1}\{\Phi(s)\} = \mathcal{L}^{-1}\{[sI - A]^{-1}\} \doteq \phi(t)$ and
$e^{At}\int_0^t e^{-A\tau}Bu(\tau)d\tau = \mathcal{L}^{-1}\{H(s)U(s)\} =$
$\mathcal{L}^{-1}\{[sI - A]^{-1}BU(s)\} \doteq \phi(t)\int_{t_0}^{t} \phi(-\tau)Bu(\tau)d\tau$

State transition matrix

Properties:

LTI	*LTV*
$\phi(0) = I$	$\phi(t, t) = I$
$\phi(t_2 - t_1)\phi(t_1 - t_0) = \phi(t_2 - t_0)$	$\phi(t_2, t_1)\phi(t_1, t_0) = \phi(t_2, t_0)$
$\phi^{-1}(t_2 - t_1) = \phi(t_2 - t_1)$	$\phi^{-1}(t_2, t_1) = \phi(t_2, t_1)$
$\frac{d}{dt}\phi(t) = A\phi(t)$	$\frac{d}{dt}\phi(t, t_0) = A(t)\phi(t, t_0)$

Determination:

- For LTI:
  1. invert $[sI - A]$ and find $\mathcal{L}^{-1}$ of each element
  2. evaluate the matrix expansion
- For LTV: numerical integration of $\frac{d}{dt}\phi(t, t_0)$ with $\phi(t_0, t_0) = I$

Problem formulation

In order to formulate the optimization/inversion problem, we need to formalize:

1. a mathematical model of the system = forward model
2. the physical/statistical constraints
3. a performance criterion: which minimized quantity can validate the optimization efficiency?

Mathematical model

$$\dot{x} = f(x, p, u, t)$$

Example: simplified gas diffusion with boundary input:

$$\frac{\partial \rho}{\partial t} = \frac{\partial}{\partial z}\left[D(z)\frac{\partial \rho}{\partial z}\right], \text{ with } \begin{cases} \rho(0, t) = \rho_{in}(t) \\ D(L)\frac{\partial \rho}{\partial z}(L, t) = 0 \\ \rho(z, 0) = \rho_0(z) \end{cases}$$

leads to the abstract LPV system [board]:

$$\dot{x}(t) = \mathcal{A}(p)x(t) + \mathcal{B}(p)u(t)$$

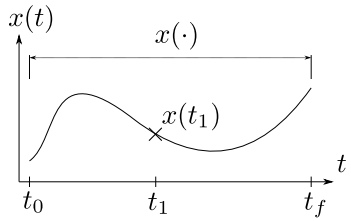
where x is the density, p the diffusion, u the boundary concentration.
The gas density variation is determined by the boundary density u, which diffuses at a rate set by the diffusion profile D (remember the scalar case $\dot{x} = -ax \Leftrightarrow x(t) = x(0)e^{-at}$).

Definitions

For $t \in [t_0,\ t_f]$:

- $u(t)$ is called the input history, or input, an exogeneous time-varying parameter, possibly controlled
- $x(t)$ is called the state trajectory (history)
- p is a model parameter (e.g. diffusion)
- sloppy distinction between a parameter and an input: let's agree that p acts in the state-space matrices and u is an exogeneous drive

Note: function $x(\cdot) \neq$ value of the function $x(t_1)$



Physical constraints

Apply both on state and on inputs. E.g. diffusion example:

- state constraints: the density profile
  - is initially zero (initial condition) $x(t_0) = [0\ 0]^T$
  - can't be negative $x(t) \geq 0\ \forall t$
  - can't exceed the maximum peak of the boundary $x(t) < \max_t u(t) = M_1\ \forall t$
- on the input: gaz density at the boundary
  - $0 \leq u(t) \leq M_1$
  - its rate of variation (increase or decrease) is limitted $|\ddot{u}(t)| < M_2$.
- on the parameter: diffusion
  - is positive and bounded $0 < p < M_3$
  - can only decrease along the tube $D_z \leq 0 \Rightarrow M_4 D \leq 0$

Definitions

- Admissible input = input history that satisfies the constraints on $[t_0\ t_f]$: $u \in \mathcal{U}$
- Admissible trajectory = state trajectory that satisfies the state/variation constraints on $[t_0\ t_f]$: $x \in \mathcal{X}$
- Terminal value: target state (point if x_f, t_f fixed)

Note: admissibility limits the range of values for both the state and input

Performance measure

- Optimal input = min (or max) a performance measure (sometimes subjective)
- Ex: make the concentration reach a desired profile at final time: $J = \|x(t_f) - x_{ref}\|$
- General form: $J = \underbrace{h(x(t_f), t_f)}_{\text{terminal cost}} + \underbrace{\int_{t_0}^{t_f} g(x(t), u(t), t)dt}_{\text{cost to go}}$

  where t_f can be specified or free
- $x(t_0) = x_0$ and $u(t)$, $t \in [t_0\ t_f]$ set the state trajectory
- Performance measure = unique real number for each trajectory of the system

The optimization problem

Find the admissible $u^*(t) \in \mathcal{U}$ such that $\dot{x} = f(x,u,t)$ follows $x^*(t) \in \mathcal{X}$ that minimizes the performance measure J. $u^*(t) \doteq$ optimal input, $x^*(t) \doteq$ optimal state trajectory.

Note:

- the optimal input may not exist (admissible input inducing an admissible trajectory)
- if it does, it may not be unique (choose)
- seek absolute/global min u^* such that

$$\begin{aligned} J^* &= h(x^*(t_f),t_f) + \int_{t_0}^{t_f} g(x^*(t),u^*(t),t)dt \\ &\leq h(x(t_f),t_f) + \int_{t_0}^{t_f} g(x(t),u(t),t)dt \end{aligned}$$

$\forall \{u \in \mathcal{U} \text{ s.t. } x \in \mathcal{X}\}$

- if max objective, then min $-J$

Optimization: problem formulation
E. Witrant
State variables
Solution of the state equations
State transition matrix
Problem formulation
Mathematical model
Physical constraints
Performance measure
Optimization problem
NC for optimality
Performance measure
Measures of length
Generalized inverses
Cross validation
Regularization
Tikhonov regularization
Barrier functions
Maximum entropy regularization
Lagrange multipliers
Conclusions

Forms of the optimal input (OI)

Definitions:

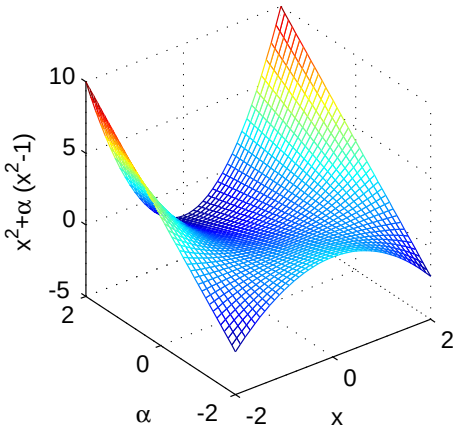
- if $u^*(t) = f(x(t),t)$ (OI $\forall x \in \mathcal{X}$) can be found at t, then f is the optimal law or optimal policy
  i.e. if $x^*(t) = Fx(t)$, $F \in \mathbb{R}^{m \times n}$, then LTI (linear time invariant) feedback of states
- if OI is determined as a function of time for a given initial condition: $u^*(t) = e(x_0,t)$, then open-loop form

$\Rightarrow$ The optimal open-loop input is optimal only for a specific x_0 while the optimal law is optimal $\forall$ state values

Optimization: problem formulation
E. Witrant
State variables
Solution of the state equations
State transition matrix
Problem formulation
Mathematical model
Physical constraints
Performance measure
Optimization problem
NC for optimality
Performance measure
Measures of length
Generalized inverses
Cross validation
Regularization
Tikhonov regularization
Barrier functions
Maximum entropy regularization
Lagrange multipliers
Conclusions

Necessary conditions for optimality (a glimpse)

Example: $\min_x J(x) = x^2 + \alpha(x^2-1)$ for a given α



- Sufficient condition: $J(x^*) \leq J(x), \forall x$
- Necessary conditions:
  - first order: $\partial J/\partial x = 0$
  - second order: $\partial^2 J/\partial x^2 > 0$
- Depending on α the problem is convex or ill-posed (multiple or $-\infty$ solutions)

Optimization: problem formulation
E. Witrant
State variables
Solution of the state equations
State transition matrix
Problem formulation
Mathematical model
Physical constraints
Performance measure
Optimization problem
NC for optimality
Performance measure
Measures of length
Generalized inverses
Cross validation
Regularization
Tikhonov regularization
Barrier functions
Maximum entropy regularization
Lagrange multipliers
Conclusions

Evaluating performance

- Performance measure:

$$J = h(x(t_f),t_f) + \int_{t_0}^{t_f} g(x(t),u(t),t)dt$$

  where x is the state of the forward model used to predict the measurement, i.e. $\hat{y} = f(x)$, which we want to compare with the actual measured data y
- Defining the prediction error $e = \hat{y} - y$,
  - which function of e (and possibly $\hat{y}$ and y) should we wish to minimize?
  - what does it imply on the solution?

$\Rightarrow$ How to set the dependency of $h(\cdot)$ and $g(\cdot)$ on the error?

- Typically associated with model assessment and selection.

Measures of length

- Norm = measure of length or size of a vector, e.g. sum of powers n denoted by $\mathcal{L}_n$:

$$\mathcal{L}_1 \text{ norm: } \|e\|_1 = \left[\sum_i |e_i|^1\right], \qquad \mathcal{L}_2 \text{ norm: } \|e\|_2 = \left[\sum_i |e_i|^2\right]^{1/2}$$

$$\ldots \mathcal{L}_n \text{ norm: } \|e\|_n = \left[\sum_i |e_i|^n\right]^{1/n}, \qquad \mathcal{L}_\infty \text{ norm: } \|e\|_\infty = \max_i |e_i|$$

- The higher is n, the more weight we put on outliers from the average trend

→ Guideline: for very accurate data, a prediction far from the observed value is important, contrarily to scattered data.

- Weighted length: weight e with the matrix W prior to norm.
- Note: $\mathcal{L}_2$ (least squares) = data obeys gaussian statistics. E.g. $J = e^T[\text{cov}(y)]^{-1}e$: maximum likelyhood method.

Generalized inverses

- Consider the linear problem: $y = Mp$ then we can find an explicit solution expressed with the generalized inverse M^{-g} such that $\hat{p} = M^{-g}y$
- Then $\hat{y} = M\hat{p} = MM^{-g}y = Ny$ and N = data resolution matrix (i.e. ideal if $N = I$): indicates the weight of each observation on the predicted value and summarized by $n = \text{diag}(N)$ (*importance* of data). Depends only on the forward model, not the data values.
- Similarly: $\hat{p} = M^{-g}Mp = Rp$ and R = model resolution matrix, if not I, $\hat{p}$ = weighted average of p
- Link between the covariances of the model parameter and of the data through the unit covariance matrix, i.e. correlated data: $[\text{cov}_u p] = M^{-g}[\text{cov}_u d]M^{-gT}$
- Measure the goodness of the resolution from the spread of off-diagonal elements (Dirichlet spread functions):

$$\text{spread}(N) = \|N - I\|_2^2, \quad \text{spread}(R) = \|R - I\|_2^2$$

Cross validation (see survey [Arlot and Celisse, 2010])

- Statistical method used to evaluate how the predicted result extends to an independent data set.
- Provides a good trade-off between bias and variance.
- From the mapping between data and predicted output $\hat{y} = Ny$, calculate the generalized cross-validation (GCV) form for n_{data} data points as:

$$\text{GCV} = \frac{1}{n_{data}}\left(\frac{\|(I-N)y\|_2}{\text{trace}(I-N)/n_{data}}\right)^2$$

- For sparse data, robust version [Lukas 2006, 2008]: $\text{RGCV} = \gamma\text{GCV} + (1-\gamma)\mu\text{GCV}$ where $\gamma \in [0;\ 1]$ is the robustness parameter (small for more robust results) and:

$$\mu \doteq \text{trace}(N^2)/n_{data} \quad \text{for uncorrelated data}$$

$$\mu = [\text{trace}(N) - \text{trace}(N^2)]/n_{data} \quad \text{for correlated data}$$

Regularization

- Needed to handle the ill-conditioning of inverse problems
- Aims to stabilize the set of possible solution and/or handle the non-uniqueness by setting the dependency of the performance measure on the input or parameter:

$$J = h(x(t_f), t_f) + \int_{t_0}^{t_f} g(x(t), u(t), t)dt$$

- Motivated by mathematical, statistical and/or physical constraints

Tikhonov regularization

- Consider the general least-squares problem $J = \|y - \hat{y}(u)\|_2^2$: infinitely many solutions
- "Adequate fit" with data agreed when $\|y - \hat{y}(u)\|_2$ is small enough, and we add some penalty on $\|u\|$:

$$\begin{array}{ll} \min_u & \|u\|_2 \\ \text{such that} & \|y - \hat{y}(u)\|_2 \leq \delta \end{array} \Leftrightarrow \begin{array}{ll} \min_u & \|y - \hat{y}(u)\|_2 \\ \text{such that} & \|u\|_2 \leq \epsilon \end{array}$$

- Combined in a single criterion as (denoting $\|v\|_M^2 = v^T M v$):

$$\min_u \|y - \hat{y}(u)\|_2^2 + \|\Gamma u\|_2^2, \text{ or } \min_u \|y - \hat{y}(u)\|_Q^2 + \|u\|_R^2$$

where Γ it the Tikhonov matrix, $R = \Gamma^T \Gamma$ (e.g. whitening filter) and Q is the inverse covariance matrix of y (e.g. Bayesian interpretation)

Tikhonov regularization (2)

- Can also be used to minimize the norm of u' (total variation - TV reg.) or u'' (rugosity) → first or second order Tikhonov regularization by using Γ for numerical discretization.
- Typically parameterized, i.e. (uncorrelated data)

$$\min_u J = (y - \hat{y}(u))^T [\text{diag}(1/\sigma^2)](y - \hat{y}(u)) + \kappa^2 u^T \Gamma^T \Gamma u$$

where $\Gamma \approx d^2/dt^2$ (discretized in a matrix form) and κ is the rugosity parameter.

- Combine with an optimal strategy to solve for κ (i.e. GCV, see firn example).

Barrier functions

- Used to transform an inequality contraint, e.g. $u \leq 0$, into an equality one included in the cost.

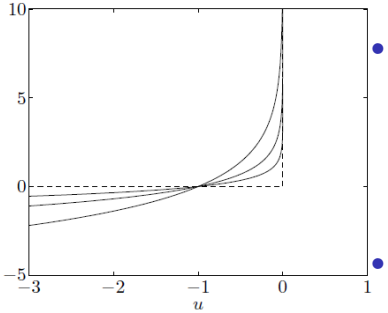


Figure 11.1 The dashed lines show the function $I_-(u)$, and the solid curves show $\hat{I}_-(u) = -(1/t)\log(-u)$, for $t = 0.5,\ 1,\ 2$. The curve for $t = 2$ gives the best approximation.

[Boyd & Vandenberghe, *Convex Optimization* 2004.]

- Consider the indicator function $\mathcal{I}_- : \mathbb{R} \to \mathbb{R}$ for nonpositive reals:

$$\mathcal{I}_-(u) = \begin{cases} 0 & u \leq 0 \\ \infty & u > 0 \end{cases}$$

- Substituted by a logarithmic barrier function (C^1 and convex): $\hat{\mathcal{I}}_-(u) = -\frac{1}{M}\log(-u)$ in J to ensure the inequality contraint.
- The precision of the approximation increases with M but a solution is harder to find → increase M iteratively (e.g. external loop on the "u-finding" loop).

Maximum entropy regularization

- Use a regularization function of the form ($w_i > 0$ weights):

$$\sum_{i=1}^{n} u_i \ln(w_i u_i)$$

- "Max entropy" from Bayesian approach to select prior probability distribution $\mathcal{P}$, i.e. (discrete case)

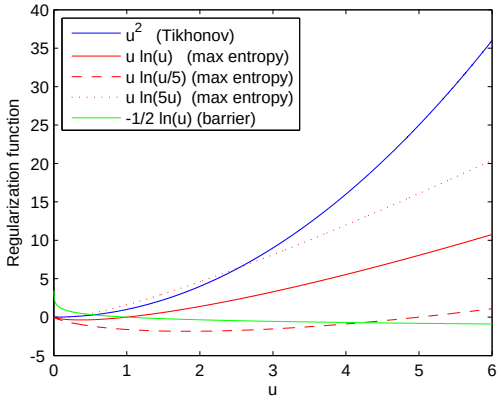
$$\begin{array}{l} \max \{J = -\sum_{i=1}^n \mathcal{P}_i \ln(\mathcal{P}_i)\} \quad \text{(entropy in statistical physics)} \\ \text{subject to } \sum_{i=1}^n \mathcal{P}_i = 1 \end{array}$$

- Can be combined with LS as:

$$\min_{u \geq 0} J(u) = \|y - \hat{y}(u)\|_2^2 + \alpha^2 \sum_{i=1}^{n} u_i \ln(w_i u_i)$$

strictly convex for linear systems (unique solution) for $\alpha \geq 0$ but may become badly conditionned when $\alpha \to 0$.

Comparison



- More penalty on large values with Tikhonov
- Increase of barrier close to 0
- Max entropy: min at $u = 1/ew$ and penalizes parameters with smaller or especially larger values

Lagrange multipliers

- Used to move equality constraints into the cost function, i.e.

$$\begin{array}{ll}\min_u & f(x,u) \\ \text{subject to} & g(x,u) = c\end{array} \quad \Leftrightarrow \quad \min_u \quad f(x,u) + \lambda(g(x,u) - c)$$

- For dynamical systems

$$\begin{array}{ll}\min_u & J = h(x(t_f), t_f) + \int_{t_0}^{t_f} g(x(t), u(t), t)dt \\ \text{subject to} & \dot{x} = f(x,u)\end{array}$$
$$\Leftrightarrow \quad \min_u \quad J + \int_{t_0}^{t_f} \lambda^T(\dot{x} - f(x,u))dt = J_a$$

where λ is the adjoint state or costate and J_a the augmented cost function (augmented Lagrangian method).

- Generalized by the Karush-Kuhn-Tucker conditions to include inequality constraints

Conclusions

- Starting from an "abstract" inversion desire, we get a mathematical formulation of the problem
- Depending on the data (deterministic or stochastic, importance of outliers) and on the model (general trust?), several possibilities for performance evaluation
- Regularization issues on the parameters, to add aditional (equality/inequality/norm) constraints
- Everything is now packed in a single function J that has to be minimized

References

- Arlot, S. and Celisse, A. (2010). *A survey of cross-validation procedures for model selection*. Statistics Surveys, 4, 40-79.
- Lukas, M. (2006). *Robust generalized cross-validation for choosing the regularization parameter*. Inv. Problems, 22, 1883.
- Lukas, M. (2008). *Strong robust generalized cross-validation for choosing the regularization parameter*. Inv. Prob., 24, 034006.
- Menke, W. (1989). *Geophysical data analysis: Discrete inverse theory*. Academic Press, New York.
- Kirk, D. (2004, 2nd Edition), *Optimal Control Theory: An Introduction*, Dover Books on Electrical Engineering.
- Aster, R.C., Borchers, B. and Thurber, C.H. (2013, 2nd Edition), *Parameter estimation and inverse problems*, Elsevier Inc. `http://www.sciencedirect.com/science/book/9780123850485`
- Boyd, S. and Vandenberghe, L. (2004), *Convex Optimization*, Cambridge University Press.

Inverse Problems and Environment:

some new tools for atmospheric studies

Lesson 4 - Optimization methods: analytical and numerical solutions

Emmanuel WITRANT
emmanuel.witrant@ujf-grenoble.fr

PhD school "MATH et COMPLEX", Department of Mathematics
UNamur, Belgium, March 10th, 2014.

Optimization: solutions
E. Witrant

Optimization: solutions
E. Witrant

Calculus of variations

Some definitions

- Function x assigns to each element $t \in \mathcal{D}$ (domain) a unique element in $\mathcal{R}$ (range)
- Functional J assigns to each function x in a class Ω (domain) a unique real number (range). Linear if and only if it satisfies the principle of homogeneity $J(\alpha x) = \alpha J(x)$
- Increment:
  - $\Delta x \doteq x(t + \Delta t) - x(t)$, noted as $x(t, \Delta t)$
  - $\Delta J \doteq J(x + \delta x) - J(x)$, noted as $J(x, \delta x)$ where δx is a variation. Example:

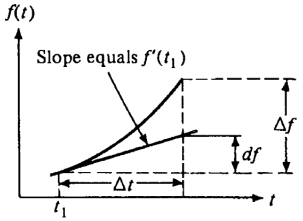
$$J(x) = \int_{t_0}^{t_f} x(t)^2 dt \rightarrow \Delta J = \int_{t_0}^{t_f} 2x(t)\delta x(t) + \delta x(t)^2 dt$$

Optimization: solutions
E. Witrant

Variation

Used to find extremes of functionals, like differentiation for functions, i.e.

- Increment of a function f of n variables q:



$$\Delta f(q, \Delta q) = \underbrace{df(q, \Delta q)}_{\text{linear in } \Delta q} + g(q, \Delta q)\|\Delta q\|$$

If $\lim_{\|\Delta q\| \to 0}\{g(q, \Delta q)\} = 0$ then f is differentiable at q and df is the differential form of f at q.

E.g. single variable $\rightarrow$ derivative, n variables $\rightarrow$ $df = \frac{\partial f}{\partial q_1}\Delta q_1 + \ldots + \frac{\partial f}{\partial q_n}\Delta q_n$

- For a functional $\Delta J(x, \delta x) = \delta J(x, \delta x) + g(x, \delta x)\|\Delta x\|$. δJ is the variation of J evaluated for x. Previous example:

$$\Delta J = \int_{t_0}^{t_f} \underbrace{2x(t)\delta x(t)}_{\text{linear in } \delta x} + \underbrace{\delta x(t)^2}_{\rightarrow 0 \text{ as } \|\delta x\| \rightarrow 0} dt$$

Fundamental theorem of the calculus of variations (FTCV)

If x^* is an extremal, the variation of J must vanish at x^*:

$$\delta J(x^*, \delta x) = 0 \quad \forall \delta x$$

Proof: by contradiction.

Functionals of a single variable

- Simplest variational problem $J(x) = \int_0^T g(x, \dot{x}, t)dt$ with $g \in C^2$, T fixed, x_0 and x_f specified,
- FTCV provides the necessary conditions for optimality [board]:

$$\frac{\partial g}{\partial x}(x^*, \dot{x}^*, t) - \frac{d}{dt}\left[\frac{\partial g}{\partial \dot{x}}(x^*, \dot{x}^*, t)\right] = 0$$

called the Euler equation.

Functionals with more variables

LPV example [handout]

Optimality for non-autonomous dynamics

- **Problem**: find u^* such that $\dot{x} = f(x(t), u(t), t)$ follows x^* that min:

$$J(u) = h(x(t_f), t_f) + \int_{t_0}^{t_f} g(x(t), u(t), t)dt$$

$\Rightarrow$ Define the Hamiltonian $\mathcal{H}(x, u, \lambda, t) \doteq g(x, u, t) + \lambda^T f(x, u, t)$. The set of NC is:

$$\dot{x}^* = \frac{\partial \mathcal{H}}{\partial \lambda}^T, \ \dot{\lambda}^* = -\frac{\partial \mathcal{H}}{\partial x}^T, \ \frac{\partial \mathcal{H}}{\partial u} = 0, \text{ at } x^*, \lambda^*, u^*, \forall t \in [t_0, t_f[$$

$$\left[\frac{\partial h}{\partial x}^T - \lambda\right]^T \delta x_f + \left[\mathcal{H} + \frac{\partial h}{\partial t}\right] \delta t_f = 0, \text{ at } t_f$$

- I.e. LQR $\dot{x} = Ax + Bu$, $x(0) = 0$, $J = \int_0^{t_f} x^T Q x + u^T R u \, dt$, t_f known $\Rightarrow$ Differential Riccati Equation [board].

Optimizing linear systems

- Solution for linear maps: $\hat{y} = M\hat{p}$ minimizing e.g. the least squares error ($Q, R > 0$)

$$\hat{p}^* = \arg\min_{\hat{p}} \left\{ J = \frac{1}{2}(y - \hat{y})^T Q (y - \hat{y}) + \frac{1}{2}\hat{p}^T R \hat{p} \right\}$$

gives the necessary conditions ($\partial J/\partial \hat{p} = 0$, $\partial^2 J/\partial \hat{p}^2 > 0$) [board]:

$$\hat{p}^* = (M^T Q M + R)^{-1} M^T Q y \text{ and } M^T Q M + R > 0$$

- Solution for linear dynamics: LQR $\rightarrow$ optimum = feedback effect
- The LPV case [handout]
- Linearize, solve and update the linearization point or use nonlinear programming?

Nonlinear Programming

E.g. Parameters estimation with Gauss-Newton gradient descent ⇒ A possible solution to determine the optimal parameters of each layer.

Problem description

Consider n_o system outputs $y \in \mathbb{R}^{n_m \times n_o}$, with n_m measurements for each output, and a model output $\hat{y} \in \mathbb{R}^{n_m \times n_o}$ (predicted by the forward model).
Objective: determine the optimal set of model parameters $\hat{p}$ which minimizes the quadratic cost function

$$J(\hat{p}) \doteq \frac{1}{n_m} \sum_{i=1}^{n_m} \|y(i) - \hat{y}(\hat{p}, i)\|_2^2$$

Output error variance is minimized for $\hat{p}^* = \arg\min_{\hat{p}} J(\hat{p})$.

Optimization: solutions
E. Witrant
Calculus of variations
Fundamental concepts
Single variable
Non-autonomous dynamics
Optimizing linear systems
Nonlinear Programming
Stochastic descent
Assumptions
For black-box models
Solving Nonlinear Equations
General Optimization
Solving Nonlinear Equations
Nonlinear Least Squares
Parameter Fitting
Linear and Quadratic Programming
Differentiation Utilities
Conclusions

Stochastic descent algorithm

Based on the sensitivity of $\hat{y}(\hat{p}, i)$ with respect to $\hat{p}$

$$S(\hat{p}, i) \doteq \frac{\partial \hat{y}}{\partial \hat{p}} = \left[\frac{\partial \hat{y}}{\partial \hat{p}_1}, \ldots, \frac{\partial \hat{y}}{\partial \hat{p}_{n_v}} \right],$$

the gradient of the cost function writes as

$$\nabla_{\hat{p}} J(\hat{p}) = -\frac{2}{n_m} \sum_{i=1}^{n_m} (y(i) - \hat{y}(\hat{p}, i))^T S(\hat{p}, i)$$

Optimization: solutions
E. Witrant
Calculus of variations
Fundamental concepts
Single variable
Non-autonomous dynamics
Optimizing linear systems
Nonlinear Programming
Stochastic descent
Assumptions
For black-box models
Solving Nonlinear Equations
General Optimization
Solving Nonlinear Equations
Nonlinear Least Squares
Parameter Fitting
Linear and Quadratic Programming
Differentiation Utilities
Conclusions

Stochastic descent algorithm (2)

$\hat{p}^*$ obtained by moving along the steepest slope $-\nabla_{\hat{p}} J(\hat{p})$ with a step η, which has to ensure that

$$\hat{p}^{l+1} = \hat{p}^l - \eta^l \nabla_{\hat{p}} J(\hat{p}^l)$$

converges to $\hat{p}^*$, where $l \doteq$ algorithm iteration index. η^l chosen according to Gauss-Newton's method as

$$\eta^l \doteq (\Psi_{\hat{p}} J(\hat{p}^l) + \upsilon I)^{-1},$$

where $\upsilon > 0$ is a constant introduced to ensure strict positiveness and $\Psi_{\hat{p}} J(\hat{p}^l)$ is the pseudo-Hessian, obtained using Gauss-Newton approximation

$$\Psi_{\hat{p}} J(\hat{p}^l) = \frac{2}{n_m} \sum_{i=1}^{n_m} S(\hat{p}^l, i)^T S(\hat{p}^l, i)$$

Optimization: solutions
E. Witrant
Calculus of variations
Fundamental concepts
Single variable
Non-autonomous dynamics
Optimizing linear systems
Nonlinear Programming
Stochastic descent
Assumptions
For black-box models
Solving Nonlinear Equations
General Optimization
Solving Nonlinear Equations
Nonlinear Least Squares
Parameter Fitting
Linear and Quadratic Programming
Differentiation Utilities
Conclusions

Stochastic descent algorithm (3)

Consider dynamical systems modeled as ($t \in [0, T]$)

$$\begin{cases} \dfrac{dx_m}{dt} &= f_m(x_m(t), \hat{p}, u(t)), \quad x_m(t_0) = x_{m0} \\ \hat{y}(t) &= g_m(x_m(t), \hat{p}, u(t)) \end{cases}$$

with known inputs $u(t)$, x_m the predicted state and $f_m(\cdot) \in C^1$, then

$$S(\hat{p}, t) = \frac{\partial \hat{y}}{\partial \hat{p}} = \frac{\partial g_m}{\partial x_m} \frac{\partial x_m}{\partial \hat{p}} + \frac{\partial g_m}{\partial \hat{p}}$$

The state sensitivity $\dfrac{\partial x_m}{\partial \hat{p}}$ is obtained by solving the ODE

$$\frac{d}{dt} \left[\frac{\partial x_m}{\partial \hat{p}} \right] = \frac{\partial f_m}{\partial x_m} \frac{\partial x_m}{\partial \hat{p}} + \frac{\partial f_m}{\partial \hat{p}}$$

Assumptions

- n_i independent system inputs $u = \{u_1, \dots, u_{n_i}\} \in \mathbb{R}^{n_m \times n_i}$, available during the optimal parameter search process.
- The set $\{y, u\}$ corresponds to historic data and J is the data variance.
- The set of n_m measurements is large enough and well chosen (sufficiently rich input) to be considered as generators of persistent excitation to ensure that the resulting model represents the physical phenomenon accurately within the bounds of u.

For black-box models

Consider the nonlinear black-box structure

$$\hat{y} = g(\phi, \hat{p}) = \sum_{k=1}^{n} \alpha_k \kappa(\beta_k(\phi - \gamma_k))$$

with $\hat{p} = \{\alpha_k, \beta_k, \gamma_k\}$. To find the gradient $\nabla_{\hat{p}} J(\hat{p})$ we just need to compute

$$\begin{aligned}
\frac{\partial}{\partial \alpha}[\alpha\kappa(\beta(\phi-\gamma))] &= \kappa(\beta(\phi-\gamma)) \\
\frac{\partial}{\partial \beta}[\alpha\kappa(\beta(\phi-\gamma))] &= \alpha\frac{\partial}{\partial \beta}[\kappa(\beta(\phi-\gamma))]\phi \\
\frac{\partial}{\partial \gamma}[\alpha\kappa(\beta(\phi-\gamma))] &= -\alpha\frac{\partial}{\partial \gamma}[\kappa(\beta(\phi-\gamma))]
\end{aligned}$$

Example: sigmoid functions family

$$\kappa_j = \frac{1}{1 + e^{-\beta_j(x-\gamma_j)}}$$

The sensibility function is set with

$$\frac{\partial \hat{y}}{\partial \alpha_j} = \frac{1}{1 + e^{-\beta_j(x-\gamma_j)}}, \quad \frac{\partial \hat{y}}{\partial \beta_j} = \frac{\alpha_j e^{-\beta_j(x-\gamma_j)}(x-\gamma_j)}{(1 + e^{-\beta_j(x-\gamma_j)})^2},$$
$$\frac{\partial \hat{y}}{\partial \gamma_j} = -\frac{\alpha_j e^{-\beta_j(x-\gamma_j)}\beta_j}{(1 + e^{-\beta_j(x-\gamma_j)})^2}.$$

Notes:

- any continuous function can be arbitrarily well approximated using a superposition of sigmoid functions [Cybenko, 1989]
- nonlinear function ⇒ nonlinear optimization problem

Example 2: LPV dynamics [handout]

Optimizing and Solving Nonlinear Equations with SCILAB

Minimization problem

$$\min_x f(x),$$

where $f \in \mathbb{R}$ maps vector variable x. Note: $\max f(x) = \min(-f)$.

Constraints

- bound, or box: x in specific intervals, i.e. $3D$ $2 \leq x(1) \leq 5$, $-1 \leq x(3) \leq 1$;
- linear equality $b^T x - c = 0$, b, x column vectors or linear inequality $b^T x - c \leq 0$;

Constraints (2)

- more general $g(x) = 0$ not solved in Scilab but

$$\min_x f(x),$$
$$g(x) = 0,$$

with $f, g \in C^1$ has solution x^* satisfying the NC (cf. Lagrangian)

$$\begin{aligned} f_x(x) + \lambda^T g_x(x) &= 0, \\ g(x) &= 0, \end{aligned}$$

where g_x is the Jacobian of g, may be solved using `fsolve`.

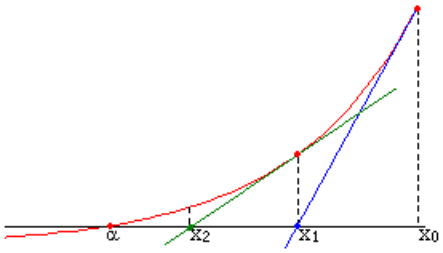
Main issues: f differentiable and gradient computation → iterative methods, search direction, how far to move, for how long, computational cost, local vs. global minima ("dart-throwing algorithm")...

Nonlinear equation solving

$$f(x) = 0$$

for m equations with n unknowns (simplest case) → key role of the Jacobian

$$J(x) = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \cdots & \frac{\partial f_1}{\partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_m}{\partial x_1} & \cdots & \frac{\partial f_m}{\partial x_n} \end{bmatrix}.$$



Invertible if $m = n$: simplest iterative method (Newton's)

$$x_{j+1} = x_j - J(x_j)^{-1} f(x_j).$$

Actual solvers: ↗ region of convergence and estimate J.

General Optimization

optim

- Accepts bound constraint, needs the gradient: if not, numerical differentiation (finite difference) of the cost `NDcost`.
- `[fopt,xopt]=optim(costf,x0)`:
  - `x0`: initial guess;
  - `[f,g,ind]=costf(x,ind)`: provides the minimized function `f`, its gradient `g` and a variable used by the optimization routine: `ind` indicates wether f can be evaluated at `x` or an interruption;
  - `fopt`: optimum value;
  - `xopt`: where this optimum value occurs.
- `[fopt,xopt]=optim(list(NDcost,myf),x0)`: if the gradient is not provided, initial guess must be a column vector.

Example

- Minimize the cost function

$$f(x, y, z) = (x - z)^2 + 3(x + y + z - 1)^2 + (x - z + 1)^2$$

Its gradient is

$$\begin{aligned} \nabla f &= \left[\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}, \frac{\partial f}{\partial z}\right] \\ &= [2(x - z) + 6(x + y + z - 1) + 2(x - z + 1), \\ &\quad 6(x + y + z - 1), \\ &\quad -2(x - z) + 6(x + y + z - 1) - 2(x - z + 1)], \end{aligned}$$

and we take an initial guess of $x_0 = [0, 0, 0]$.

Example (2): algorithm

```
function z= myf(x)
    z=(x(1)-x(3))^2+3*(x(1)+x(2)+x(3)-1)^2+(x(1)-x(3)+1)^2
endfunction

function z=myg(x)
    xs=x(1)+x(2)+x(3)-1;
    z=[2*(x(1)-x(3))+6*xs+2*(x(1)-x(3)+1), 6*xs,...
       -2*(x(1)-x(3))+6*xs-2*(x(1)-x(3)+1)]
endfunction

function [f,g,ind]=costf(x,ind)
    f=myf(x);g=myg(x);
endfunction

x0=[0 0 0];                              % initial condition
[fopt,xopt]=optim(costf,x0);             % x0 a row vector
[fopt,xopt]=optim(costf,x0');            % x0 a column vector
[fopt,xopt,gopt]=optim(list(NDcost,myf),x0'); % x0 column
```


Example (3): Running the algorithm

```
-> x0= [0 0 0];
-> [fopt,xopt]=optim(costf,x0)
    xopt = !  0.0833333   0.3333333   0.5833333 !
    fopt = 0.5
-> [fopt,xopt]=optim(costf,x0')
    xopt =
        ! 0.0833333 !
        ! 0.3333333 !
        ! 0.5833333 !
    fopt = 0.5
-> [fopt,xopt,gopt]=optim(list(NDcost,myf),x0')
    gopt =
        ! 0.          !
        ! 0.          !
        ! 1.833D-11 !
    xopt =
        ! 0.0833333 !
        ! 0.3333333 !
        ! 0.5833333 !
    fopt = 0.5
```


Full call

To set limits on number of iterations (local optimum $\rightarrow \nabla J < \epsilon$) or constrain the optimization problem:

```
[f,[xopt,[gradopt,[work]]]]=optim(costf,[contr],x0,['algo'],...
[df0,[mem]],[work],[stop],['in'],[imp=iflag])
```

- `contr`: include constraints $b^T x - c = 0$ or $x_{inf} \leq x \leq x_{sup}$
- `algo`: specify the optimization algorithm, e.g. quasi-Newton, conjugate gradient or nondifferentiable
- `mem`: nb of variables for Hessian approx.
- `stop`: controls the algorithm convergence with max. number of calls/iterations, threshold on gradient norm/defrease of `f`/variation of x
- `work`: working array for hot restart (quasi-Newton)

Solving Nonlinear Equations

- For nonlinear system

$$f(x) = 0$$

uses the Powell hybrid method (bi-directional search along each search vector) and is based on the package MINPACK

- `[x [,v [,info]]]=fsolve(x0,fct [,fjac] [,tol])` where `info` indicates why termination occurred.
- Example:

$$\min f(x,y,z) = (x-z)^2 + 3(x+y+z-1)^2 + (x-z+1)^2,$$

where $g(x,y,z) = 1 - x - 3y - z^2 = 0$. The solution satisfies:

$$\begin{aligned} \nabla f + \lambda \nabla g &= 0 \\ g &= 0 \end{aligned}$$

Solved with the script:

```
function z=fct(x)
  xs = x(1)+x(2)+x(3)-1;
  wl = [2*(x(1)-x(3))+6*xs+2*(x(1)-x(3)+1),6*xs, ...
      -2*(x(1)-x(3))+6*xs-2*(x(1)-x(3)+1)];
  w2 = [-l -3 -2*x(3)];
  z  = [w1'+x(4)*w2';l-x(1)-3*x(2)-x(3)^2];
endfunction

x0 = [0 0 0 0];
[x,v] = fsolve(x0,fct);
```

We get:

```
-> v              % value of function at x
  ans = 1.OD-16 * ! 0.00167 0.00502 0.00233 -1.6653345 !
-> x
  ans = ! 0.19722 0.10555 0.69722 -1.675D-19 !
```

and have found a solution since $v \approx 0$.

Nonlinear Least Squares

Problem description

$$\min_x f(x) \Rightarrow \min_x \|f(x)\|^2 = f(x)^T f(x) = \sum_{i=1}^{m} f_i(x)^2$$

Note: this allows for $m > n$ (more equations than unknowns).

leastsq

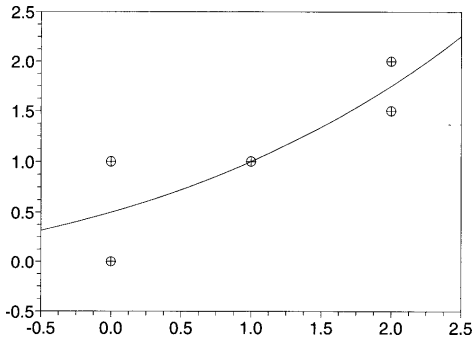
- Same as `optim` except that `costf` replaced by `f`
- Short call:
  `[f,xopt]=leastsq([imp,] fun [,Dfun],x0)`
- Long call:

```
[f,[xopt,[gradopt]]]=leastsq(fun [,Dfun],[contr],x0,...
                             ['algo'],[df0,[mem]],[stop],['in'])
```


- Example: we want to fit the data points

$$\{(0,0),(0,1),(1,1),(2,1.5),(2,2)\}$$

with parameters $p = (a, b, c)$ such that $y = ae^{bt} + c$
$\rightarrow$ 5 equations (points) for 3 param.: $y_i - p_1 e^{p_2 t_i} - p_3 = 0$

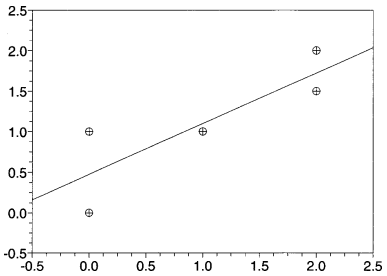


lsqrsolve

- Minimizes the sum of squares using Levenberg-Marquardt algorithm ($\approx$ gradient + Gauss-Newton, numerical).
- Script:

```
[x [,v [,info]]]=lsqrsolve(x0,fct,m [,stop [,diag]])
[x [,v [,info]]]=lsqrsolve(x0,fct,m, fjac [,stop [,diag]])
```

  where `diag` contains multiplicative scale factors for the variables
- Example: same as before, results in a different solution vector with close $\|f(x)\|^2$



$\Rightarrow$ Problems can have several solutions, even with same IC, depending on numerical method!

Parameter Fitting

- Dedicated Scilab function `datafit` based on `optim`: for given function $G(p, y)$, find the best p for $G(p, y_i) = 0$ in the set of measurement vectors y_i.
- p^* from min $\sum_{i=1}^{n} G(p, y_i)^T W G(p, y_i)$: weighted LS.
- `[p, err] =datafit(G,Y,p0)`, e.g.:

```
Y= [0 0 1 2 2; 0 1 1 1.5 2]
function e=G(p,y)
   e=y(2)-p(1)*exp(p(2)*y(1))-p(3)
endfunction
p0=[0 0 0]';
[p,err]=datafit(G,Y,p0);
```

- long call:

```
[p,err]=datafit([imp,] G [,DG],Y [,W],[contr],p0,[algo],
[df0,[mem]],[work],[stop],['in'])
```

Optimization: solutions
E. Witrant
Calculus of variations
Fundamental concepts
Single variable
Non-autonomous dynamics
Optimizing linear systems
Nonlinear Programming
Stochastic descent
Assumptions
For black-box models
Solving Nonlinear Equations
General Optimization
Solving Nonlinear Equations
Nonlinear Least Squares
Parameter Fitting
Linear and Quadratic Programming
Differentiation Utilities
Conclusions

Linear and Quadratic Programming

Linear programs

- Minimize $p^T x$ subject to linear constraints:

$$\begin{aligned} &min_x p^T x, \\ &C_1 x \leq b_1, \\ &c_l \leq x \leq c_l, \quad (IneqC) \\ &C_2 x = b_2. \end{aligned}$$

Optimization: solutions
E. Witrant
Calculus of variations
Fundamental concepts
Single variable
Non-autonomous dynamics
Optimizing linear systems
Nonlinear Programming
Stochastic descent
Assumptions
For black-box models
Solving Nonlinear Equations
General Optimization
Solving Nonlinear Equations
Nonlinear Least Squares
Parameter Fitting
Linear and Quadratic Programming
Differentiation Utilities
Conclusions

- Calling sequence depends on how many types of constraints:

```
[x,lagr,f]=linpro(p,C,b [,x0])
[x,lagr,f]=linpro(p,C,b,cl,cu [,x0])
[x,lagr,f]=linpro(p,C,b,cl,cu,me [,x0])
[x,lagr,f]=linpro(p,C,b,cl,cu,me,x0 [,imp])
```

where

- `C`: LH constraints matrix in $Cx \leq b$, if (IneqC) $\rightarrow$ `C = []`, equality const. listed first;
- `b`: RH constraint vector, if (IneqC) $\rightarrow$ `b = []`;
- `cl/cu`: lower/upper bounds in $c_l \leq x \leq c_u$;
- `me`: number of equality constraints;
- `x0`: initial guess or property (vertex) of the calculated initial feasible point.

Optimization: solutions
E. Witrant
Calculus of variations
Fundamental concepts
Single variable
Non-autonomous dynamics
Optimizing linear systems
Nonlinear Programming
Stochastic descent
Assumptions
For black-box models
Solving Nonlinear Equations
General Optimization
Solving Nonlinear Equations
Nonlinear Least Squares
Parameter Fitting
Linear and Quadratic Programming
Differentiation Utilities
Conclusions

Quadratic Programs

- Cost function replaced by the quadratic expression

$$\frac{1}{2} x^T Q x + p^T x$$

- `quapro`:

```
[x,lagr,f]=quapro(Q,p,C,b [,x0])
[x,lagr,f]=quapro(Q,P,C,b,ci,cs [,x0])
[x,lagr,f]=quapro(Q,p,C,b,ci,cs,me [,x0])
[x,lagr,f]=quapro(Q,p,C,b,ci,cs,me,x0 [,imp])
```

Semidefinite Programs (i.e. LMIs)

`[x,Z,ul,info]=semidef(x0,Z0,F,blck-szs,c,options)`: cf. Scilab help. Alternative SeDuMi, YALMIP...

Differentiation Utilities

→ To get numerical estimates of derivatives and Jacobians directly: `numdiff` and `derivative`.

numdiff

- Numerical estimate using finite difference method.
- g=`numdiff(fun,x [,dx])`, where
  - `fun`: differentiated function,
  - `x`: vector argument,
  - `dx`: discretization step,
  - `g`: estimated gradient (Jacobian).
- Example: Compute the Jacobian of

$$f(x_1,x_2,x_3)=\begin{pmatrix} x_1+2x_2^2x_3 \\ \sin(x_1x_2x_3)\end{pmatrix}, \quad \text{at } x=\begin{pmatrix}1\\2\\3\end{pmatrix}$$

using `numdiff` and compare with the true Jacobian.

Optimization: solutions
E. Witrant
Calculus of variations
Fundamental concepts
Single variable
Non-autonomous dynamics
Optimizing linear systems
Nonlinear Programming
Stochastic descent
Assumptions
For black-box models
Solving Nonlinear Equations
General Optimization
Solving Nonlinear Equations
Nonlinear Least Squares
Parameter Fitting
Linear and Quadratic Programming
Differentiation Utilities
Conclusions

- Scilab script:

```
function z = myf(x)
   z = [x(1)+2*x(2)-x(2)^2*x(3);
        sin(x(1)*x(2)*x(3))]
endfunction
x = [1;2;3];
J = numdiff(myf,x)
TrueJ = [1,2-2*x(2)*x(3), -x(2)^2];
a = cos(x(1)*x(2)*x(3));
TrueJ = [TrueJ;a*[x(3)*x(2),x(1)*x(3),x(1)*x(2)]];
Difference = J-TrueJ;
```

- Solution:

```
-> J =  ! 1.0000000  - 10.         - 4.       !
        ! 5.7610218    2.8805109    1.9203406 !
-> TrueJ =  ! 1.          - 10.         - 4.       !
            ! 5.7610217    2.8805109    1.9203406 !
-> Difference =  ! 1.286D-08  - 4.045D-08    3.518D-08 !
                 ! 1.157D-07    5.690D-08  - 1.100D-08 !
```

Optimization: solutions
E. Witrant
Calculus of variations
Fundamental concepts
Single variable
Non-autonomous dynamics
Optimizing linear systems
Nonlinear Programming
Stochastic descent
Assumptions
For black-box models
Solving Nonlinear Equations
General Optimization
Solving Nonlinear Equations
Nonlinear Least Squares
Parameter Fitting
Linear and Quadratic Programming
Differentiation Utilities
Conclusions

Higher order derivatives

- Numerical differentiation → ill conditioning and error, especially for higher order derivatives.
- Alternative: symbolic differentiation (*Maple* or ADOL-C: arbitrary order forward/reverse) but may be very slow for complex problems → prefer automatic differentiation.
- If first and second order, `derivative` can be useful.
- Suppose m values of $x\in\mathbb{R}^n$, then at a (Taylor):

$$f(x)=\begin{pmatrix} f_1(x)\\ \vdots \\ f_m(x)\end{pmatrix}=f(a)+J(a)(x-a)+\begin{pmatrix}(x-a)^TH_1(a)(x-a)\\ \vdots \\ (x-a)^TH_m(a)(x-a)\end{pmatrix}+\dots$$

First derivative $J\in\mathbb{R}^{m\times n}$ and m second derivatives $H_i\in\mathbb{R}^{n\times n}$.

Optimization: solutions
E. Witrant
Calculus of variations
Fundamental concepts
Single variable
Non-autonomous dynamics
Optimizing linear systems
Nonlinear Programming
Stochastic descent
Assumptions
For black-box models
Solving Nonlinear Equations
General Optimization
Solving Nonlinear Equations
Nonlinear Least Squares
Parameter Fitting
Linear and Quadratic Programming
Differentiation Utilities
Conclusions

Higher order derivatives (2)

`[J [,H]] = derivative(f, x[, h, order, H-form, Q])`

- `h`: step size (better free);
- `order`: of the finite difference used to approximate the derivatives;
- `H-form`: form in which the Hessian will be returned;
- `Q`: real orthogonal matrix.

Example (same as before):

```
function z=myf(x)
  z=[x(1)+2*x(2)-x(2)^2*x(3); sin(x(1)*x(2)*x(3))]
endfunction

x=[1; 2; 3];
[J,H]=derivative(myf,x,H_form='hypermat')
```

Higher order derivatives (3)

```
H =
(:,:,1)
! 0.    0.    0. !
! 0.  - 6.  - 4. !
! 0.  - 4.    0. !
(:,:,2)
! 10.058956   7.9099883   5.2733256 !
! 7.9099883   2.5147394   2.6366631 !
! 5.2733256   2.6366631   1.117662  !

J =
! 1.          10.          4.         !
! 5.7610217   2.8805109    1.9203406 !

-> TrueJ =  ! 1.          - 10.          - 4.       !
            ! 5.7610217     2.8805109    1.9203406 !
```

Optimization: solutions
E. Witrant

Calculus of variations
Fundamental concepts
Single variable
Non-autonomous dynamics
Optimizing linear systems
Nonlinear Programming
Stochastic descent
Assumptions
For black-box models
Solving Nonlinear Equations
General Optimization
Solving Nonlinear Equations
Nonlinear Least Squares
Parameter Fitting
Linear and Quadratic Programming
Differentiation Utilities
Conclusions

Conclusions

- Identify the appropriate tools for your class of problems
- Include as much preliminary analytical work as possible (e.g. Jacobian, bounds, ...)
- Do not hesitate to try different resolution schemes and compare
- Can also use imbricated schemes to "approach" the solution successively

Optimization: solutions
E. Witrant

Calculus of variations
Fundamental concepts
Single variable
Non-autonomous dynamics
Optimizing linear systems
Nonlinear Programming
Stochastic descent
Assumptions
For black-box models
Solving Nonlinear Equations
General Optimization
Solving Nonlinear Equations
Nonlinear Least Squares
Parameter Fitting
Linear and Quadratic Programming
Differentiation Utilities
Conclusions

References

1. Kirk, D. (2004, 2nd Edition), *Optimal Control Theory: An Introduction*, Dover Books on Electrical Engineering.
2. S. Campbell, J-P. Chancelier and R. Nikoukhah, *Modeling and Simulation in Scilab/Scicos*, Springer, 2005.
3. Scilab website: *http://www.scilab.org*.

Inverse Problems and Environment : some new tools for atmospheric studies

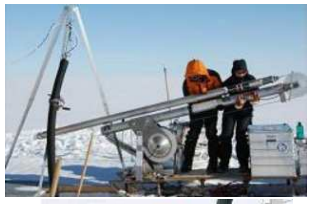
Lesson 6 - From polar cores to trace gas history

Emmanuel WITRANT[1], Patricia MARTINERIE[2] et al.[3]

[1]UJF Physics dept, GIPSA-lab / Systems & Control
[2]CNRS, Laboratoire de Glaciologie et Géophysique de l'Environnement (LGGE)
[3]NEEM gas modelling group (>14 countries, 6 firn models), CIC (Denmark), IMAU (Netherlands), INSTAAR (USA), Stony Brook (USA), UEA (UK)

PhD school "MATH et COMPLEX", UNamur, March 11th, 2014

Motivation



Trace gas measurements in interstitial air from polar firn

- allow to reconstruct their atmospheric concentration time trends over the last 50 to 100 years
- provides a unique way to reconstruct the recent anthropogenic impact on atmospheric composition

Converting depth-concentration profiles in firn into atmospheric concentration histories requires models of trace gas transport in firn

Challenges

Firn air analysis is a complex problem involving :

- Physical modelling & Fluid mechanics
- Transport description with distributed (PDE) equations
- Optimization in a large-scale (i.e. 400×9 ODE) framework
- Close connection with instrumental issues
- Sparse measurements

$\Rightarrow$ Need for a pluridisciplinary approach

Firn trace gas modeling

From measurements in interstitial air

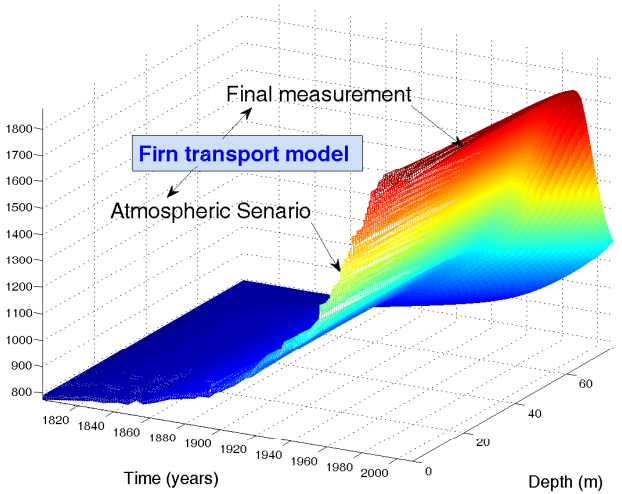
- Gases with known history ("atmospheric scenario") :
  - direct model can be used to compute the space distribution when ice core is drilled ("final time")
  - matching measurement and final modeled distribution by adjusting $D(z)$ = inverse diffusivity model
  - use of several gases to constrain D = multiple gases diff. model

  $\Rightarrow$ characterize the physical transport properties of each site
- Gas with unknown history :
  - use firn properties and final measurements to reconstruct history = inverse scenario model
  - refined by considering optimization over multiple sites

$\Rightarrow$ Atmospheric concentration reconstruction

Direct model

I.e. CH_4 transport at NEEM (Greenland)



$\Rightarrow$ Constrain the dynamics by conservation laws

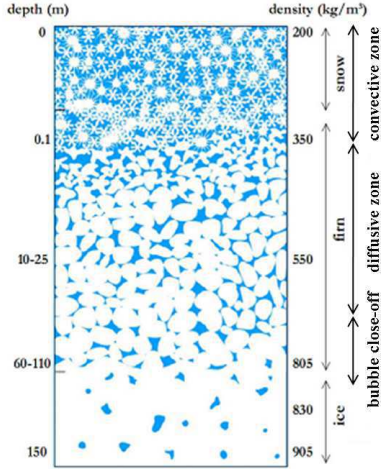
Poromechanics : three interconnected networks in firns

Ice lattice, gas connected to the surface (open pores) and gas trapped in bubbles (closed pores) :

$$\frac{\partial[\rho_{ice}(1-\epsilon)]}{\partial t} + \nabla[\rho_{ice}(1-\epsilon)\vec{v}] = 0$$

$$\frac{\partial[\rho^o_{gas} f]}{\partial t} + \nabla[\rho^o_{gas} f(\vec{v}+\vec{w}_{gas})] = -\vec{r}^{o\to c}$$

$$\frac{\partial[\rho^c_{gas}(\epsilon-f)]}{\partial t} + \nabla[\rho^c_{gas}(\epsilon-f)\vec{v}] = \vec{r}^{o\to c}$$



Scheme adapted from [Sowers et al.'92, Lourantou'08].

Trace gas conservation in open pores [Rommelaere & al.'97, Witrant et. al ACP'12]

- Flux driven by advection with air and firn sinking
- Flux driven by mol. diff. due to concentration gradients
- Flux driven by external forces : gravity included with Darcy-like flux
- Sink = particles trapped in bubbles & radioactive decay
- Boundary input : surface concentration
- Results in transport PDE :

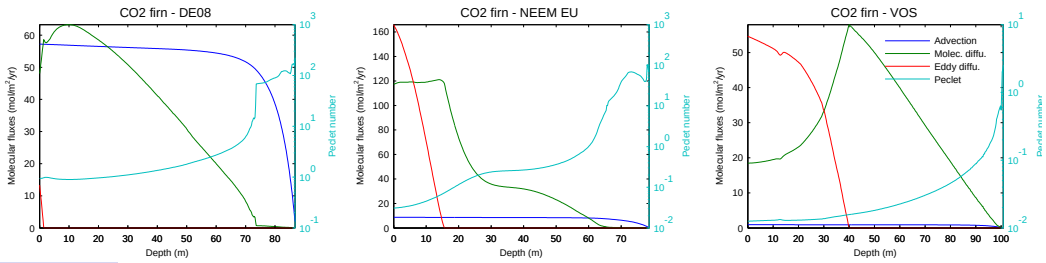
$$\frac{\partial}{\partial t}[\rho^o_\alpha f] + \frac{\partial}{\partial z}[\rho^o_\alpha f(v+w_{air})] + \rho^o_\alpha(\tau+\lambda) - \frac{\partial}{\partial z}\left[\mathbf{D}_\alpha\left(\frac{\partial \rho^o_\alpha}{\partial z} - \rho^o_\alpha\frac{\partial \rho_{air}/\partial z}{\rho_{air}} + \mathcal{A}_{ss}\right)\right] = 0$$

$$\rho^o_\alpha(0,t) = \rho^{atm}_\alpha(t), \qquad \frac{RT}{M_f}\frac{\partial \rho^o_\alpha}{\partial z}(z_f) - \rho^o_\alpha(z_f) = 0$$

with $\mathcal{A}_{ss}$ such that $\partial[\rho^o_{\alpha,ss} f]/\partial t = 0$ at steady state, i.e.

$$\mathcal{A}_{ss} = -\frac{\rho^o_{\alpha,ss} f}{D_\alpha}(w_\alpha - w_{air}) - \left(\frac{\partial \rho^o_{\alpha,ss}}{\partial z} - \rho^o_{\alpha,ss}\frac{\partial \rho_{air}/\partial z}{\rho_{air}}\right)$$

CO_2 transport at 3 polar sites (↘ accumulation)



- Molecular fluxes ($\rho^o_{gas} f(\vec{v} + \vec{w}_{gas})$) due to :
  - Air and firn sinking $v + w_{air}$
  - Molecular diffusion $\Delta\vec{w}_\alpha$
  - Surface "eddy" flows $\Delta\vec{w}_F$
- Péclet number (advection/diffusion)

Inverse diffusivity model

Problem formulation

- Least squares minimization (single gas) :

$$D^*_\alpha = \arg\min_{D_\alpha} \frac{1}{z_f}\int_0^{z_f} \frac{1}{\sigma^2_\alpha}\left(m_\alpha - \frac{\rho^o_\alpha(D_\alpha)}{\rho^o_{air}}\right)^2 \delta_\alpha dz$$

  with the constraints on $\partial\rho^o_\alpha/\partial t$, $D(z) > 0$ and $dD/dz < 0$
- For N gas :

$$D^*_{CO_2} = \arg\min_{D_{CO_2}} \sum_{i=1}^{N} \frac{1}{z_f}\int_0^{z_f} \frac{1}{\sigma^2_{\alpha_i}}\left(m_{\alpha_i} - \frac{\rho^o_{\alpha_i}(D_{CO_2})}{\rho^o_{air}}\right)^2 \delta_{\alpha_i} dz$$

- Nonlinear optimization problem (at least with implicit schemes)

Optimal diffusivity identification [IEEE Med'10]

Final-cost optimization problem with dynamics and inequality constraints

$$\min_D \mathcal{J}(D) = \mathcal{J}_{meas} + \mathcal{J}_{reg}, \text{ under the constraints } \begin{cases} C(\rho, D) = 0 \\ \mathcal{I}(D) < 0 \end{cases}$$

Considering N gas and including the constraints in the cost (*Lagrange* param.) :

$$\min_D \mathcal{J}(D) \doteq \sum_{i=1}^{N} [\mathcal{J}_{meas}(\rho_i, \rho_{meas}) + \mathcal{J}_{trans}(C(\rho_i, D))] + \mathcal{J}_{ineq}(D) + \mathcal{J}_{reg}(D)$$

with :

$$\begin{cases} \mathcal{J}_{meas} = \frac{1}{2}\int_0^{z_f} r_i(\rho_{meas} - \rho_i|_{t=t_f})^2 \delta_z \, dz & \text{Measurement cost} \\ \mathcal{J}_{trans} = \int_0^{t_f}\int_0^{z_f} \lambda_i C(\rho_i, D)\, dzdt & \text{Transport constraint} \\ \mathcal{J}_{reg} = \frac{1}{2}\int_0^{z_f} s(z) D^2 \, dz & \text{Regularization function} \end{cases}$$

Theorem 1 - Transport and linearized dynamics

Consider the general transport equation

$$\begin{cases} \partial_t y + f_1(z,t)y + f_2(z,t)\partial_z y = \partial_z\left[g(y, \partial_z y, u)\right] \\ y(0,t) = y_0(t), \quad k_1\partial_z y(L,t) + k_2 y(L,t) = 0 \\ y(z,0) = y_I(z) \end{cases}$$

Its linearized dynamics along the reference trajectory $(\bar{y}, \bar{u}, \bar{y}_I)$ with perturbations $(\tilde{y}, \tilde{u}, \tilde{y}_I)$ is given by

$$\begin{cases} \partial_t\tilde{y} + f_1(z,t)\tilde{y} + f_2(z,t)\partial_z\tilde{y} \\ \quad = \partial_z\left[\partial_y\bar{g}\,\tilde{y} + \partial_{\partial_z y}\bar{g}\,\partial_z\tilde{y} + \partial_u\bar{g}\,\tilde{u}\right] \\ \tilde{y}(0,t) = 0, \quad k_1\partial_z\tilde{y}(L,t) + k_2\tilde{y}(L,t) = 0 \\ \tilde{y}(z,0) = \tilde{y}_I(z) \end{cases}$$

where $\bar{g} \doteq g(\bar{y}, \partial_z\bar{y}, \bar{u})$.

Theorem 2 - Adjoint state

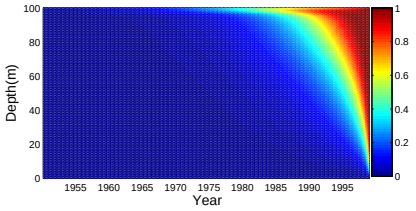
Consider the linearized transport equation without input :

$$\begin{cases} \partial_t \tilde{y} = \partial_z \left[f_1(z,t)\partial_z \tilde{y} + f_2(z,t)\tilde{y} \right] + f_3(z,t)\tilde{y} \\ \tilde{y}(0,t) = 0, \quad k_1 \partial_z \tilde{y}(L,t) + k_2 \tilde{y}(L,t) = 0, \quad \tilde{y}(z,0) = 0 \end{cases}$$

The corresponding adjoint state is given as :

$$\begin{cases} \partial_t \lambda = -f_3 \lambda + (f_2 - \partial_z f_1)\, \partial_z \lambda - f_1 \partial_{zz} \lambda \\ \lambda(0,t) = 0,\ f_1\ \partial_z \lambda + [f_1 k_2/k_1 - f_2]\lambda\,|_{z=L} = 0, \quad \lambda(z,T) = 0 \end{cases}$$

Adjoint state with $\lambda(z,t_f) = 1$:



Firn air modeling
E. Witrant
Direct model
Interconnected networks
Conservation in open pores
CO_2 transport at 3 polar sites
Inverse diffusivity
Problem formulation
Optimal diffusivity identification
Multi-gas results
Diffusivities
$\delta^{15}N$ and Pe
CH_3CCl_3
Inverse scenario
Isotopic ratio models
Automatic rugosity tuning
Results
Heptafluoropropane in atmosphere
CO budget
Closed pores

Inequality constraint

- Applies on $D > 0$ and $\partial_z D < 0$
- Change of variables

$$\begin{aligned} \partial_z y_{IC} &= u, \quad y_{IC}(z_f) = 0 \\ D &= y_{IC} \end{aligned}$$

  and where u is the new optimization variable
- Introducing *Lagrange* parameters $\lambda_{IC}(z,t)$ and a barrier function $\mathcal{R}(u)$ s.t. $u < 0$:

$$\begin{aligned} \mathcal{J}_{ineq} &= \int_0^L \lambda_{IC}(\partial_z y_{IC} - u) + \mathcal{R}(u)\, dz \\ &= \int_0^L -y_{IC}\partial_z \lambda_{IC} - u\lambda_{IC} + \mathcal{R}(u)\, dz + \lambda_{IC} y_{IC}|_0^{z_f} \end{aligned}$$

Firn air modeling
E. Witrant
Direct model
Interconnected networks
Conservation in open pores
CO_2 transport at 3 polar sites
Inverse diffusivity
Problem formulation
Optimal diffusivity identification
Multi-gas results
Diffusivities
$\delta^{15}N$ and Pe
CH_3CCl_3
Inverse scenario
Isotopic ratio models
Automatic rugosity tuning
Results
Heptafluoropropane in atmosphere
CO budget
Closed pores

Theorem 3 - Adjoint-based gradient

The gradients of $\mathcal{J}$ with respect to the decision variables u and y_{li} along the reference trajectory $(\bar{u}, y(\bar{u}))$ are given by :

$$\nabla_u \mathcal{J} = \mathcal{R}'(\bar{u}) - \int_0^T \lambda_{IC}\, dt, \quad \nabla_{y_i} \mathcal{J} = -\lambda_i(z,0)$$

where λ_i are the solutions of ($\mathcal{P} \doteq$ meas. cost) :

$$\begin{cases} \partial_t \lambda_i = -f_3 \lambda_i + (f_2 - \partial_z f_1)\, \partial_z \lambda_i - f_1 \partial_{zz} \lambda_i \\ \lambda_i(0,t) = 0, \quad k_1 f_1\ \partial_z \lambda_i + [k_2 f_1 - k_1 f_2]\lambda_i\,|_{z=L} = 0 \\ \lambda_i(z,T) = -\mathcal{P}'(\bar{y}_i(T)) \end{cases}$$

and λ_{IC} is obtained from : $\begin{cases} \partial_z \lambda_{IC} = \sum_{i=1}^N f_4 \partial_z \lambda_i \\ \lambda_{IC}(0,t) = 0, \end{cases}$

i.e. applies with

$$\mathcal{P} = \frac{1}{2} r_i (q_{meas} - q_i|_{t=t_f})^2 \delta_z, \quad \mathcal{R} = -\frac{1}{M}\log(-\partial_z D) + \frac{1}{2} s(z) \partial_z D^2$$

Firn air modeling
E. Witrant
Direct model
Interconnected networks
Conservation in open pores
CO_2 transport at 3 polar sites
Inverse diffusivity
Problem formulation
Optimal diffusivity identification
Multi-gas results
Diffusivities
$\delta^{15}N$ and Pe
CH_3CCl_3
Inverse scenario
Isotopic ratio models
Automatic rugosity tuning
Results
Heptafluoropropane in atmosphere
CO budget
Closed pores

Gradient steepest descent algorithm

Require : $\bar{D} = D_{init}$ s.t. $\partial_z \bar{D} < 0$, $\bar{y}_{li} = y_{li,init}$, $M > 0$, $\epsilon_{ineq,\,grad} > 0$, $\Delta M > 0$

while $\left| \mathcal{J}_{ineq} / \mathcal{J}_{meas} \right| > \epsilon_{ineq}$ **do**

  while $\left| \nabla_{\partial_z D} \mathcal{J} + \sum_{i=1}^{N} \nabla_{y_{li}} \mathcal{J} \right| > \epsilon_{grad}$ **do**

    Solve for $\bar{y}_i$ with $\bar{D}$ and $\bar{y}_{li}$

    Compute λ_i and λ_{IC}

    Compute $\partial_z \tilde{D} = -\nabla_{\partial_z D} \mathcal{J}$ and $\tilde{y}_{li} = -\nabla_{y_{li}} \mathcal{J}$, then $\tilde{D} = \int_0^z \partial_\eta \tilde{D} d\eta$

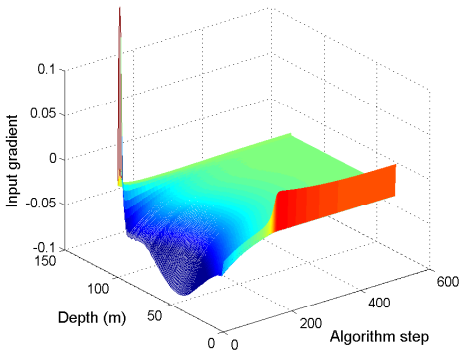
    Update $\bar{D} \doteq \bar{D} + \delta_D \tilde{D}$ s.t. $\partial_z \bar{D} < 0$ and $\bar{y}_{li} \doteq \bar{y}_{li} + \delta_{y_{li}} \tilde{y}_{li}$, with $\delta_{D,\,y_{li}} \in (0,\,1)$

  end while

  $M \doteq M \times \Delta M$

end while

Simulation results

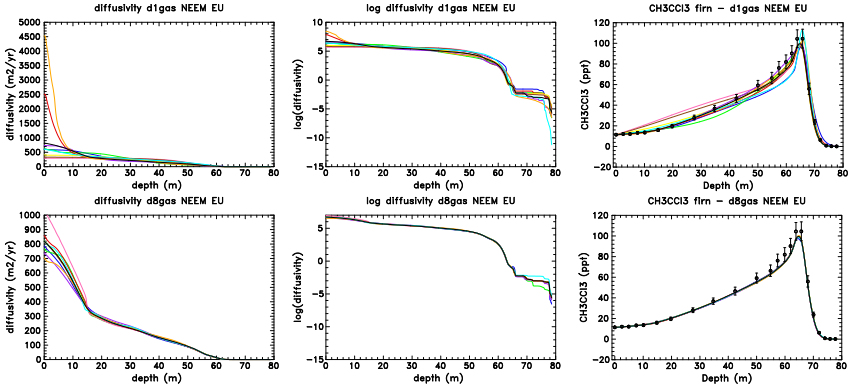


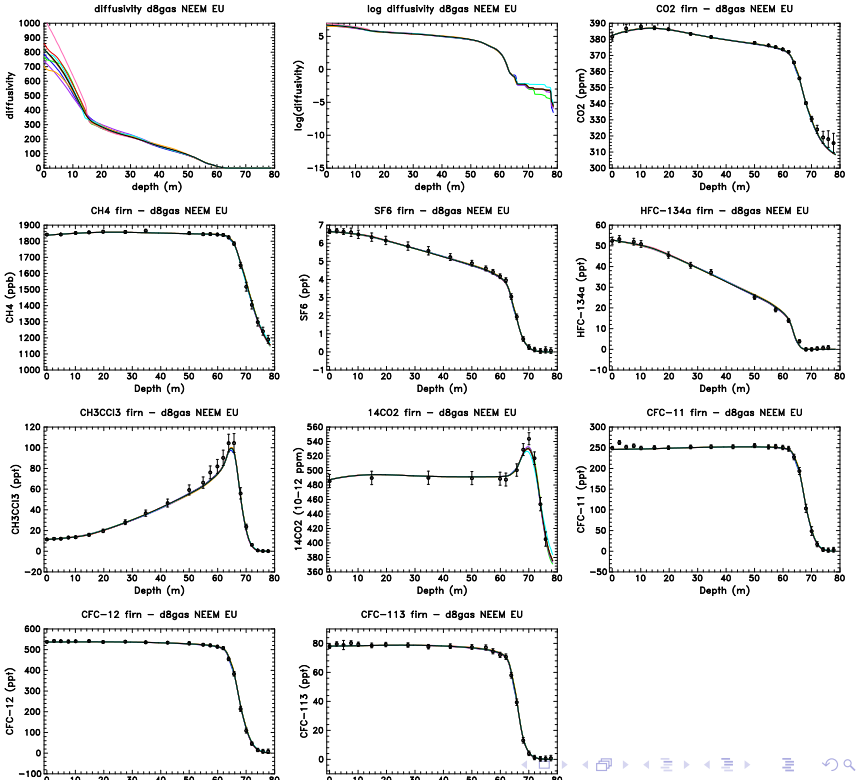
Evolution of the gradient $\nabla_{\partial_z D}\mathcal{J}$

- smooth convergence within ≈ 500 steps
- highly sensitive to design weights and constraint

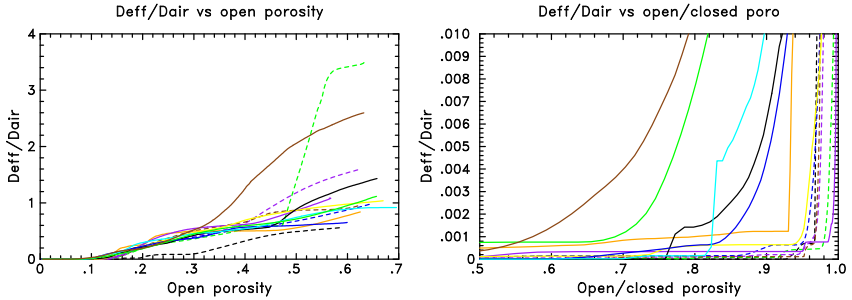
$\Rightarrow$ model revision and reference results from nonlinear LS algorithm

Single vs. multi-gas (8/9 gas) optimization



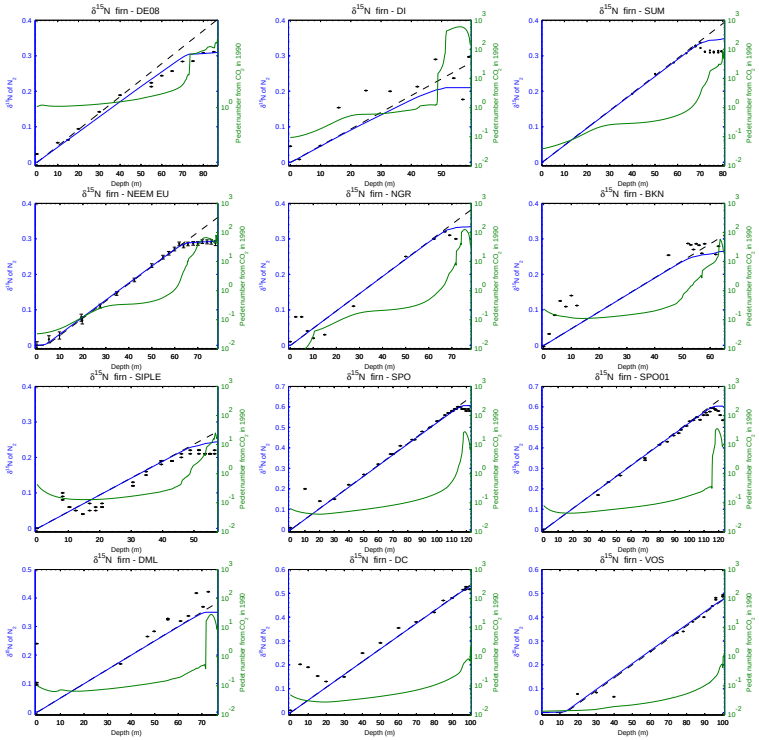


Result = Diffusivities at 11 sites (13 holes) [ACP'12]



Arctic (dashed) : Devon Island (black), Summit (blue), NEEM-EU (purple) and NEEM-US (brown), North GRIP (green).
Antarctic (continuous) : DE08 (orange), Berkner (purple), Siple (yellow), South Pole 1995 (dark blue), South Pole 2001 (light blue), Dronning Maud Land (black), Dome C (green) and Vostok (brown)

- Low diffusivity at Devon Island due to melt layers
- High diffusivity in upper firn related to convection
- Very consistent diff. at intermediate depths (0.1-0.3)
- High diff. in deep firn at Vostok and Dome C (low accu. and cold), consistent with very young ages and no plateau in $\delta^{15}N$
- Reasonable scaling laws $D_{\text{eff}}(f, T, P_{\text{atm}}, accu)$ for paleo studies



Ch_3CCl_3 diffusion at NEEM from the 50s [Buizert et al., ACP'12]

[video]

Inverse scenario model

Background : A "deconvolution" approach for trace gas models [Rommelaere et al., JGR, 1997]

- Green function = impulse response of the firn $\Rightarrow$ age probabilities $\rho_{firn}(z,t_f) = G(z,t) * \rho_{atm}^{true}(t)$: "convolution"
- Deconvolution (estimate ρ_{atm}) :

$$\begin{aligned} \epsilon(z) &= G(z,t)\rho_{atm}(t) - \rho_{firn}(z,t_f) \\ \rho^*_{atm}(t) &= arg\min_{\rho_{atm}} \left[\epsilon^T (diag\{1/\sigma^2_{mes}(z)\})\epsilon + \kappa^2 \rho^T_{atm} R \rho_{atm}\right] \end{aligned}$$

- Under-constrained pb $\Rightarrow$ add extra information with rugosity characteristic matrix $R > 0$ (i.e. d^2/dt^2) + κ.
- Model behavior controlled by κ (rugosity) and $\sigma^2_{mes}(z)$

$\Rightarrow$ Extend to LTV for isotopic ratios (process tracers of geochemical cycles [Hoefs 2009]), measured by

$$\delta^{min}X = 1000 \times \left(\frac{[^{min}X]/[^{maj}X]}{R_{std}} - 1 \right)$$

Isotopic ratio models, an LTV approach [Witrant & Martinerie, IFAC 2013]

- Specific interest for isotopic ratio with measurements in $\delta(t) = \left(\frac{\rho_1(t)/\rho_2(t)}{R_{std}} - 1 \right) \times 1000$ where $\rho_2(t)$ is known
- A direct approach (convert into ρ_1) results in poor results : need to work with a Linear Parameter-Varying system in δ.
- Considering the discretized dynamics ($k = 1, \ldots, N$)

$$\begin{aligned} \rho_{1,k} &= A_{d1}\,\rho_{1,k-1} + B_{d1}\,\rho^{atm}_{1,k} \\ \rho_{2,k} &= A_{d2}\,\rho_{2,k-1} + B_{d2}\,\rho^{atm}_{2,k} \end{aligned}$$

we get the linear parameter-varying (LPV) system :

$$\delta_k = A_{D,k}\,\delta_{k-1} + B_{D2,k}\,\delta^{atm} + 10^3\,(A_{D,k} \times 1 + B_{D2,k} - 1)$$

with $A_{D,k} \doteq \mathrm{diag}(1/\rho_{2,k}) A_{d1}\,\mathrm{diag}(\rho_{2,k-1})$ and $B_{D2,k} \doteq \mathrm{diag}(\rho^{atm}_{2,k}/\rho_{2,k}) B_{d1}$

Problem Formulation

- Consider SIMO LTV systems ($k = 1, \dots, N_t$) :

$$\begin{aligned} x_k &= A_{D,k}x_{k-1} + B_{D,k}u_k + w_k, \quad x_{k=0} = x_0 \in \mathbb{R}^N \\ y_{N_t} &= Cx_{N_t} \quad \in \mathbb{R}^M \end{aligned}$$

- Multi-process case ($i = 1 \dots N_{proc}$) :

$$\begin{aligned} x_{i,k} &= A_{Di,k}x_{i,k-1} + B_{Di,k}u_k + w_{i,k} \\ x_{i,0} &\approx (I - A_{Di,0})^{-1}(B_{Di,0}u_0 + w_{i,0}) \\ y_{i,N_{ti}} &= C_i x_{i,N_{ti}} \quad \in \mathbb{R}^{M_i} \end{aligned}$$

$\Rightarrow$ Find the optimal input history that min (e.g. $\mathcal{L}_2$) the modeling error $\epsilon \doteq y_m - y_{N_t}(u)$

- Underconstrained : use regularization term and stochastic information on the measurements.

Cost function and optimal design

- Optimization problem :

$$u^*(t) = \arg\min_u \left\{ J(u) = \frac{1}{2}\sum_{i=1}^{N_{proc}} \|y_{i,N_{ti}}(u) - y_{m,i}\|^2_{Q_i} + \frac{1}{2}\int_{t_0}^{t_{f,max}} \|u''(t)\|^2_R \, dt \right\}$$

discretized to consider :

$$J(U) = \frac{1}{2}\sum_{i=1}^{N_{sites}} \|G_i U - \bar{y}_i\|^2_{Q_i} + \frac{\kappa}{2}\|FU\|^2_R \, dt$$

where $Q_i = \mathrm{diag}\left(1/\sigma_i^2(j)\right) \geq 0$ and $R > 0$.

$\rightarrow$ Provides the measurement to model mapping :

$$Y = \underbrace{\bar{G}\left[\kappa F^T R F + \sum_{i=1}^{N_{sites}} G_i^T W_i G_i\right]^{-1} G\bar{W}}_{\doteq S_\kappa} \bar{Y}$$

- Problem : what is the "good" κ ?

Automatic rugosity tuning

Physical approaches :

1. Effective degree of freedom $dY_\kappa \approx \mathrm{trace}(S_\kappa)$ and min cross-validation curve :

$$CV(Y) = \frac{1}{N_{data}}\sum_{i=1}^{N_{data}} \left(\frac{\bar{Y}_i - Y_i}{1 - S_\kappa(i,i)}\right)^2$$

2. Data prediction versus model resolution [Menke, 1989 ; Rommelaere et al., 1997] :

$$\kappa^* = \min_\kappa \left\{ rmsd(\bar{Y} - Y) + rmsd(\tilde{Y}_u) \right\}$$

where $\tilde{Y}_u = \bar{\mathcal{G}}\sqrt{\mathrm{diag}(\mathrm{cov}(\bar{U}^*))}$ reflects the model resolution impact on the output.

Automatic rugosity tuning (2)

Stochastic approaches : Bias vs variance [Lukas 2008, 2009] using the generalized CV

$$\mathrm{GCV}(\kappa) = \frac{1}{N_{data}}\left(\frac{\|(I - S(\kappa))\bar{Y}\|}{\mathrm{tr}(I - S(\kappa))/N_{data}}\right)^2$$

robustified (sparse measurements) as

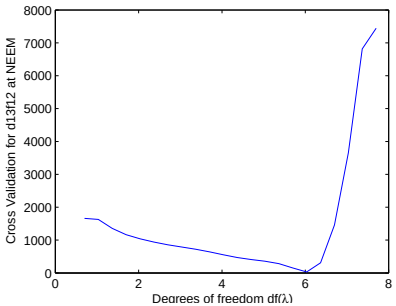
$$\mathrm{RGCV}(\kappa) = \gamma\mathrm{GCV}(\kappa) + (1-\gamma)\mu(\kappa)\mathrm{GCV}(\kappa)$$

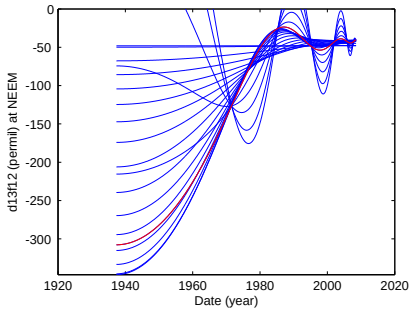
$\Rightarrow \kappa^*$ which min RGCV with

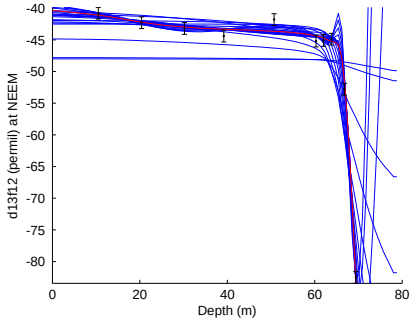
1. $\mu(\kappa) \doteq \mathrm{tr}(S(\kappa)^2)/N_{data}$ (RGCV)
2. $\mu(\kappa) = [\mathrm{tr}(S(\kappa)) - \mathrm{tr}(S(\kappa)^2)]/[N_{data}\kappa]$ (R1GCV)

Inverse scenario for $\delta^{13}C$ of CFC-12 at NEEM EU 2009 (data from [Zuiderweg et al. 2013])

First test with the effective degree of freedom : large variability !



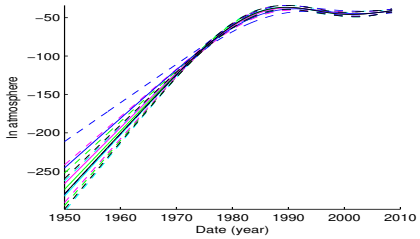


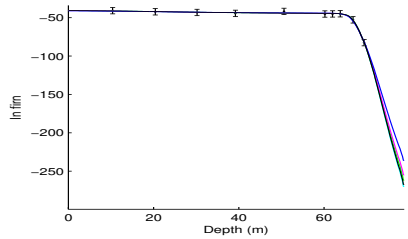


Firn air modeling
E. Witrant
Direct model
Interconnected networks
Conservation in open pores
CO_2 transport at 3 polar sites
Inverse diffusivity
Problem formulation
Optimal diffusivity identification
Multi-gas results
Diffusivities
$\delta^{15}N$ and Pe
Ch_3CCl_3
Inverse scenario
Isotopic ratio models
Automatic rugosity tuning
Results
Heptafluoropropane in atmosphere
CO budget
Closed pores

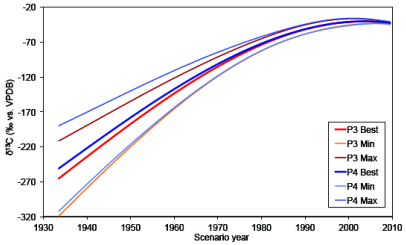
Inverse scenario for $\delta^{13}C$ of CFC-12 at NEEM (2)

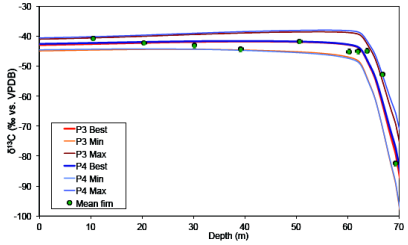
More tractable results obtained with RGCV :





Methods : Weighted RMSD - norm 1, Weighted RMSD - norm 2, Generalized cross-Validation (GCV), Robust GCV, Robust 1 GCV.





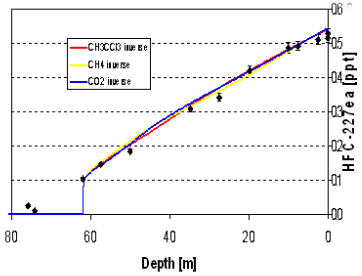
Polynomial approximation + Monte-Carlo ([Zuiderweg et al. 2012]).

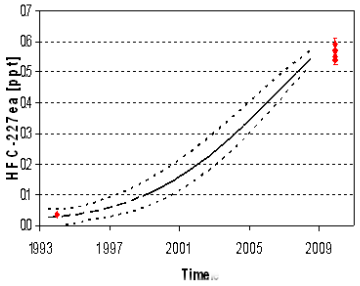
Firn air modeling
E. Witrant
Direct model
Interconnected networks
Conservation in open pores
CO_2 transport at 3 polar sites
Inverse diffusivity
Problem formulation
Optimal diffusivity identification
Multi-gas results
Diffusivities
$\delta^{15}N$ and Pe
Ch_3CCl_3
Inverse scenario
Isotopic ratio models
Automatic rugosity tuning
Results
Heptafluoropropane in atmosphere
CO budget
Closed pores

Some results

Accelerating growth of HFC-227ea in the atmosphere [Laube *et. al*'10]

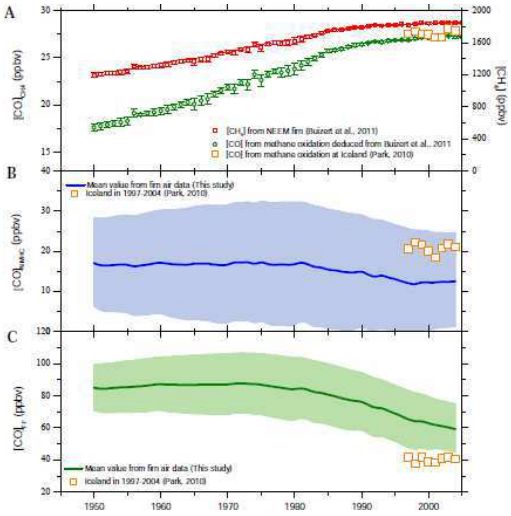
- HFC-227ea = substitute for ozone depleting compounds
- Firn air samples collected in Greenland used to reconstruct a history of atmospheric abundance from 2000 to 2007
- ↗ growth rate confirmed by upper tropospheric air samples in 2009
- Stratospheric lifetime of 370 years from high altitude aircraft and balloons





Firn air modeling
E. Witrant
Direct model
Interconnected networks
Conservation in open pores
CO_2 transport at 3 polar sites
Inverse diffusivity
Problem formulation
Optimal diffusivity identification
Multi-gas results
Diffusivities
$\delta^{15}N$ and Pe
Ch_3CCl_3
Inverse scenario
Isotopic ratio models
Automatic rugosity tuning
Results
Heptafluoropropane in atmosphere
CO budget
Closed pores

The isotopic record of Northern Hemisphere atmospheric carbon monoxide since 1950, implications for the CO budget [Wang *et. al*'12]



⇒ Increase untill the 70s then drop (i.e. associated with fossil fuel : catalytic converters and diesel engines)

Newly detected ozone depleting substances 1 in the atmosphere [Laube et al., Nature Geosciences, 9/03/14]

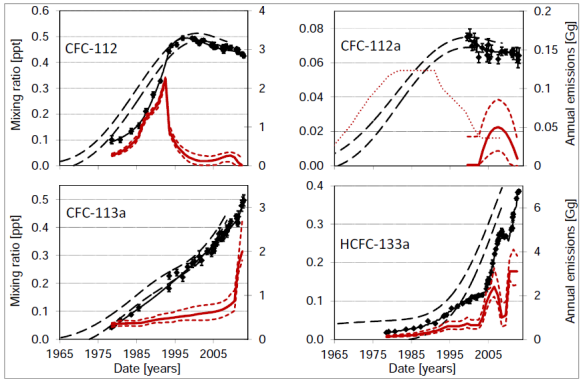


FIGURE: Atmospheric history and global emissions of CFC-112, CFC-112a, CFC-113a, and HCFC-133a from NEEM firn air (dashed), Cape Grim (diamonds), Known emissions (red), model response to emissions (black cont.).

[video from Wall Street Journal]

Other results

- Atmospheric impacts and ice core imprints of a methane pulse from clathrates [Bock *et. al*, EPSL'12]
- Reconstruction of the carbon isotopic composition of methane over the last 50 yr based on firn air measurements at 11 polar sites [Sapart *et. al*, ACPD'12]
- Natural and anthropogenic variations in methane sources over the last 2 millennia [Sapart *et. al*, Nature'12]
- Extreme ^{13}C depletion of CCl2F2 in firn air samples from NEEM, Greenland [Zuiderweg *et. al*, ACP'13]
- Emissions halted of the potent greenhouse gas SF5CF3 [Sturges *et. al*, ACP'12]
- Distributions, long term trends and emissions of four perfluorocarbons in remote parts of the atmosphere and firn air [Laube *et. al*, ACP'12]
- A 60-yr record of atmospheric CO reconstructed from Greenland firn air [Petrenko *et. al*, ACPD'12]
- Eemian interglacial reconstructed from a Greenland folded ice core, [NEEM community members, Nature'13]

Simulation and continuity in closed pores

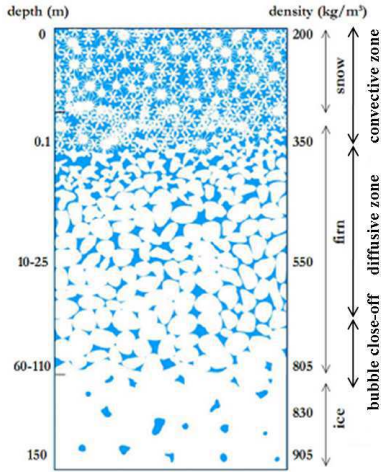
The gas trapped in bubbles (closed pores) continues into the ice.
Air transport is driven by :

$$\frac{\partial[\rho_{ice}(1-\epsilon)]}{\partial t} + \nabla[\rho_{ice}(1-\epsilon)\vec{v}] = 0$$

$$\frac{\partial[\rho^o_{gas} f]}{\partial t} + \nabla[\rho^o_{gas} f(\vec{v} + \vec{w}_{gas})] = -\vec{r}^{o\to c}$$

$$\frac{\partial[\rho^c_{gas}(\epsilon - f)]}{\partial t} + \nabla[\rho^c_{gas}(\epsilon - f)\vec{v}] = \vec{r}^{o\to c}$$

with $\vec{r}^{o\to c} \neq 0$ for $z \in [0,\, z_{co}]$ and $\vec{r}^{o\to c} = 0$ for $z \in]z_{co},\, z_f]$.
(Numerical example seen in Lesson 1.)



Scheme adapted from [Sowers et al.'92, Lourantou'08].

Conclusions

- Forward model obtained from conservation laws + fluxes description.
- Initial linear ill-conditionned model transformed into a nonlinear robust model → mitigated approach to assign an LPV approximation to the nonlinear problem ?
- 2 inverse problems = 2 completely different strategies.
- Importance of normalization and sufficiently robust problem formulation.
- Also used as a tool to adjust intercallibration between the different labs.

Inverse Problems and Environment: some new tools for atmospheric studies
Class exercises

Emmanuel WITRANT, March 8, 2014.
UJF, UFR de Physique, MiSCIT

I. Lesson 1: Modeling

Discretization example.

Consider the gas diffusion in a porous tube with boundary input and trapping:

$$\frac{\partial \rho}{\partial t} = \frac{\partial}{\partial z}\left[D(z)\frac{\partial \rho}{\partial z}\right] - \tau(z)\rho, \text{ with } \begin{cases} \rho(0,t) = \rho_{in}(t) \\ D(L)\dfrac{\partial \rho}{\partial z}(L,t) = 0 \\ \rho(z,0) = \rho_0(z) \end{cases}$$

which we wish to discretize at $z_0 = 0$, $z_1 = \Delta z$, ..., $z_i = i\Delta z$, ..., $z_N = L$.

For example, apply central difference over half steps, i.e. $f_z \approx \frac{f_{i+1/2} - f_{i-1/2}}{\Delta z}$:

$$\begin{aligned} F(z,t) = D(z)\frac{\partial \rho}{\partial z} &\approx \frac{D_{i+1/2} + D_{i-1/2}}{2}\frac{\rho_{i+1/2} - \rho_{i-1/2}}{\Delta z} = F_i \\ \frac{\partial}{\partial z}\left[D(z)\frac{\partial \rho}{\partial z}\right] = \frac{\partial F}{\partial z} &\approx \frac{F_{i+1/2} - F_{i-1/2}}{\Delta z} \\ &= \frac{1}{\Delta z}\left(\frac{D_{i+1} + D_i}{2}\frac{\rho_{i+1} - \rho_i}{\Delta z} - \frac{D_i + D_{i-1}}{2}\frac{\rho_i - \rho_{i-1}}{\Delta z}\right) \\ &= \frac{1}{2\Delta z^2}[D_i + D_{i-1} \quad -(D_{i+1} + 2D_i + D_{i-1}) \quad D_{i+1} + D_i]\begin{bmatrix} \rho_{i-1} \\ \rho_i \\ \rho_{i+1} \end{bmatrix} \end{aligned}$$

This works for $i = 2, \ldots, N-1$. For the extremal values, we need the boundary conditions:

$$\begin{aligned} \rho(0,t) = \rho_{in}(t) &\Rightarrow \rho_0 = \rho_{in} \\ D(L)\frac{\partial \rho}{\partial z}(L,t) = 0 &\Rightarrow D_N\frac{\rho_{N+1} - \rho_N}{\Delta z} = 0 \text{ forward scheme } \Leftrightarrow \rho_{N+1} = \rho_N \end{aligned}$$

We also need some extra constraints on $D(z)$ for the "fictitious values" outside of the domain, e.g. (Neumann): $D_z(0) = D_z(L) = 0$. Thus:

$$\frac{\partial}{\partial z}\left[D\frac{\partial \rho}{\partial z}\right] - \tau\rho \approx \begin{cases} \frac{1}{2\Delta z^2}[-(D_2 + 3D_1) - \tau_1 \quad D_2 + D_1]\begin{bmatrix} \rho_1 \\ \rho_2 \end{bmatrix} + \frac{1}{\Delta z^2}D_1\rho_{in} & \text{for } i = 1 \\ \frac{1}{2\Delta z^2}[D_i + D_{i-1} \quad -(D_{i+1} + 2D_i + D_{i-1}) - \tau_i \quad D_{i+1} + D_i]\begin{bmatrix} \rho_{i-1} \\ \rho_i \\ \rho_{i+1} \end{bmatrix} & \text{for } i = 2, \ldots, N-1 \\ \frac{1}{2\Delta z^2}[D_N + D_{N-1} \quad -(3D_N + D_{N-1}) + 2D_N - \tau_N]\begin{bmatrix} \rho_{N-1} \\ \rho_N \end{bmatrix} & \text{for } i = N \end{cases}$$

and defining the system state as $X = [\rho_1 \ \ldots \ \rho_N]^T \in \mathbb{R}^N$, we obtain the state-space dynamics:

$$\dot{X}(t) = \mathcal{A}(D, \tau, \Delta z)X(t) + \mathcal{B}(D_1, \Delta z)\rho_{in}(t)$$

II. Lesson 3: Formulating the optimization problem

Consider the inverse diffusivity problem with known boundary input and trapping rate. It can be associated with the class of systems (LPV) that write as:

$$\dot{x} = \underbrace{\left(A_0 + \sum_{j=1}^{N_p} A_j a_j\right) x + \left(B_0 + \sum_{j=1}^{N_p} B_j a_j\right) u}_{f(x,a,u)}, \tag{1}$$

$$y = Cx \tag{2}$$

where $x \in \mathbb{R}^N$ is the state, $u \in \mathbb{R}^1$ a known input, $a \in \mathbb{R}^{N_p}$ the unknown parameters, $y \in \mathbb{R}^{N_m}$ the measured output, and A_i, B_i and C the state-space matrices of appropriate dimensions. We suppose that the initial state is known, i.e. $x(t_0) = x_0$.

The optimization problem is formulated as a least squares problem with the cost function with terminal constraints and a regularization term:

$$J = \frac{1}{2}||y(t_f) - y_m||_Q^2 + \text{Reg}(a) \tag{3}$$

where $||\epsilon||_Q^2 \doteq \epsilon^T Q \epsilon$ denotes the weighted quadratic norm and $\text{Reg}(a)$ is an arbitrary regularization term (e.g. quadratic or imposing positivity). The measurements y_m are obtained at the final time t_f.

We take the regularization function:

$$\text{Reg}(a) = \frac{1}{2N_p} a^T R a - \frac{M}{N_x} \mathbf{1}^{1\times N_x} \log(D_0 + M_a a) \tag{4}$$

where $M \geq 0$ is a scalar and $\mathbf{1}^{1\times N_x}$ is a vector of ones of dimension $1 \times N_x$. For example, choosing a as being the diffusivity profile ($a_i = D(x_i)$), R is used to impose a second order Tikhonov regularization (parameterized in terms of a rugosity coefficient) and the log function (with $M_a = I$) is a positivity constraint.

III. Lesson 4: Solving the optimization problem

A. The variations

Including the dynamics constraint (1) in the cost (3) with the Lagrange parameter λ, we obtain the augmented cost function:

$$J_a = \frac{1}{2}||y(t_f) - y_m||_Q^2 + \text{Reg}(a) + \int_{t_0}^{t_f} \lambda^T \left[f(x,a,u) - \dot{x}\right] dt \tag{5}$$

The first order variation of J_a is obtained from the variations on x, x_f, $\dot{x}$, a and λ as:

$$\begin{aligned}\delta J_a = {} & (Cx(t_f) - y_m)^T Q\, \delta x_f + \frac{\partial}{\partial a}\text{Reg}(a)\, \delta a + \int_{t_0}^{t_f} \left[f(x,a,u) - \dot{x}\right]^T \delta\lambda + \lambda^T f_a \delta a dt \\ & + \int_{t_0}^{t_f} (\lambda^T f_x + \dot{\lambda}^T)\delta x\, dt + \lambda(t_f)^T \delta x_f \end{aligned} \tag{6}$$

where we used the identity:

$$[\lambda^T \delta x]_t = \lambda^T \delta\dot{x} + \delta x^T \dot{\lambda} = \lambda^T \delta\dot{x} + \dot{\lambda}^T \delta x \tag{7}$$

and:

$$f_x \doteq A_0 + \sum_{j=1}^{N_p} A_j a_j \tag{8}$$

$$f_a \doteq \left[\begin{array}{ccc} A_1 x + B_1 u & \dots & A_{N_p} x + B_{N_p} u \end{array} \right] \tag{9}$$

$$\frac{\partial}{\partial a}\text{Reg}(a) = \frac{1}{2N_p} a^T (R + R^T) - \frac{M}{N_x} \mathbf{1}^{1 \times N_x} \text{diag}\left(\frac{1}{D_0 + M_a a} \right) M_a \tag{10}$$

$$= \frac{1}{2N_p} a^T (R + R^T) - \frac{M}{N_x} \left(\frac{1}{D_0 + M_a a} \right)^T M_a \tag{11}$$

$$\frac{\partial^2}{\partial a^2}\text{Reg}(a) = \frac{1}{2N_p} (R + R^T) + \frac{M}{N_x} M_a^T \text{diag}\left(\frac{1}{D_0 + M_a a} \right)^2 M_a \tag{12}$$

B. The optimality conditions

Defining the adjoint state as:

$$\dot{\lambda} = -f_x^T \lambda \tag{13}$$

$$\lambda(t_f) = Q(Cx(t_f) - y_m) \tag{14}$$

and along the state trajectories described by (1), the cost variation (6) becomes:

$$\delta J_a = \left[\frac{\partial}{\partial a}\text{Reg}(a) + \int_{t_0}^{t_f} \lambda^T f_a \, dt \right] \delta a \tag{15}$$

Thus, choosing:

$$\delta a = -\tau \left[\frac{\partial}{\partial a}\text{Reg}(a) + \int_{t_0}^{t_f} \lambda^T f_a \, dt \right]^T \tag{16}$$

with $\tau > 0$ ensures that $\delta J_a \leq 0$.

Starting from an initial a^0 satisfying the inequality constraints possibly expressed in the regularization term, the gradient descent algorithm is obtained as follows.

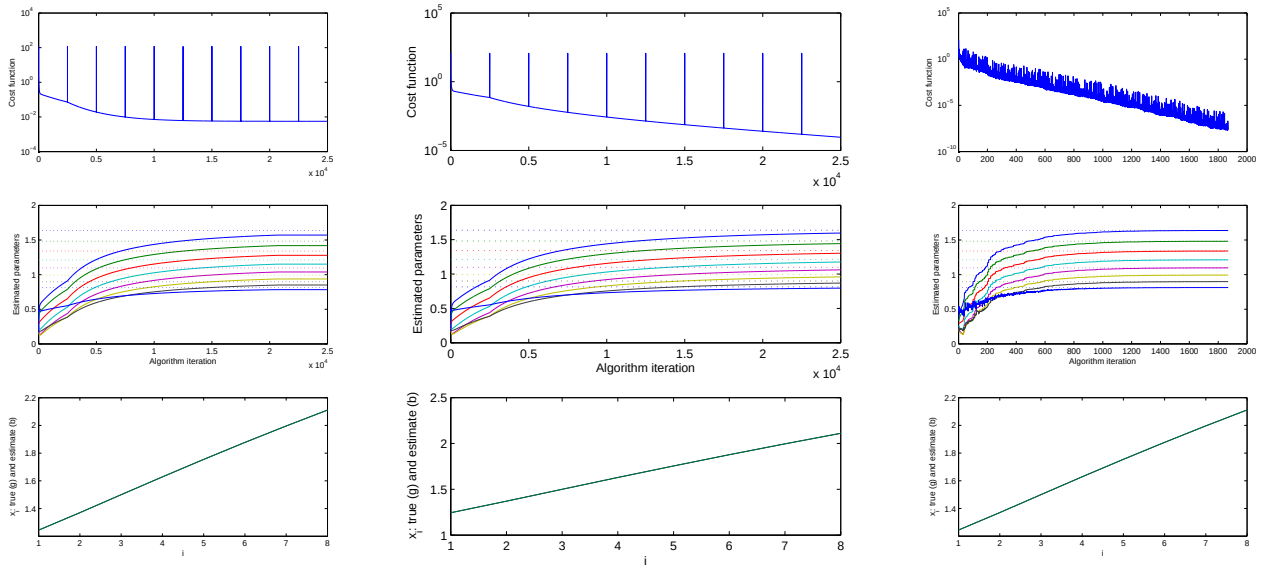
Require: $\bar{a} = a^0$ s.t. $\text{Reg}(a)$ exists
while $|\delta J_a| > \epsilon_{grad}$ **do**
  Solve for $\bar{x}$ with $\bar{a}$ using (1)
  Compute λ_i from (13)-(14)
  Compute δa from (16)
  Update $\bar{a} \doteq \bar{a} + \delta a$
  Compute δJ_a from (15)
end while

Note that ϵ_{grad} is chosen to obtain the desired precision and τ is the tunning parameter for the convergence speed (typically reduced when the solution is approached). The simulation results, for a diffusive equation discretized in 10 spatial steps, are presented in Fig. 1(a) and Fig. 1(b) for two different choices of the regularization function.

Choosing τ such that the variation evolves as a percent q of the cost, i.e.:

$$\tau = \frac{\frac{q}{100}|J|}{\left| \left| \frac{\partial}{\partial a}\text{Reg}(a) + \int_{t_0}^{t_f} \lambda^T f_a \, dt \right| \right|^2} \tag{17}$$

we obtain the results presented on Fig. 1(c) (without regularization).



(a) Quadratic regulation

(b) Rugosity regularization that vanishes

(c) No Reg. τ such that the $\delta J \approx 2J$ ($q = 200$)

Fig. 1. Diffusivity estimation example (8 states) with the gradient descent method. The peaks on the cost function are due to attempts to increase τ that are cancelled due to a gradient increase.

C. Gradient descent approach with the sensitivity computation

Another approach consists in using the state sensitivity to the estimated parameter in order to compute the gradient and evolution law. The gradient can be computed from (3) as:

$$\nabla J = (y(t_f) - y_m)^T Q \frac{\partial y}{\partial a}(t_f) + \frac{\partial}{\partial a}\text{Reg}(a) \tag{18}$$

where $\partial y/\partial a$ is the sensitivity of the model output with respect to a. From the fact that the system (1)-(2) is continuously differentiable, this sensitivity can be evaluated using the so-called "ODE-method" by computing the dynamics:

$$\dot{x} = f(x, a, u) \tag{19}$$

$$\frac{d}{dt}\left[\frac{\partial x}{\partial a}\right] = f_x(a)\frac{\partial x}{\partial a} + f_a(x, u) \tag{20}$$

$$\frac{\partial y}{\partial a} = C\frac{\partial x}{\partial a} \tag{21}$$

with the initial sensitivity estimated as $\partial x/\partial a(0) = f_x^{-1} f_a(x(0), u(0))$.

The optimal parameter a^* is obtained by moving along the steepest slope $-\nabla J(a)$ with a step α, which as to be small enough to ensure that

$$\dot{a} = -\alpha \nabla J(a) \tag{22}$$

converges to a^*. This step is chosen according to the *damped Newton's method* [Madsen et al., 1999] and writes as

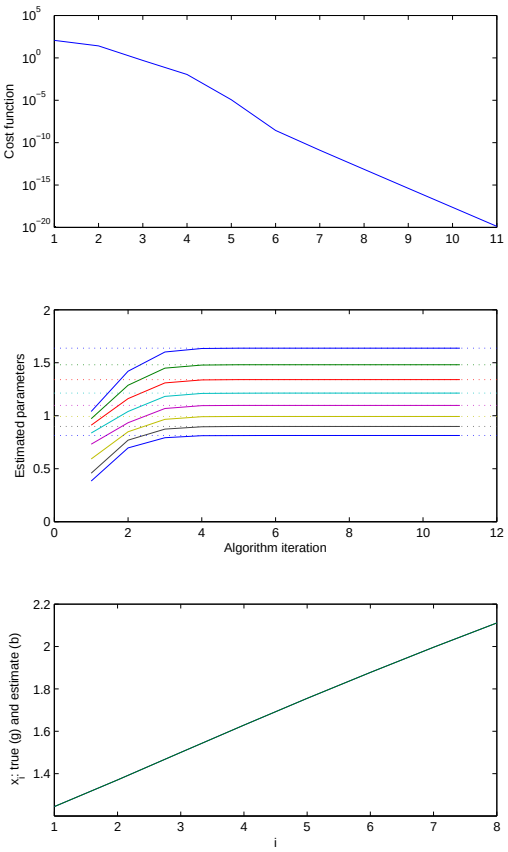
$$\alpha \doteq (\Psi J(a) + \upsilon I)^{-1}$$

where υ is a positive constant introduced to ensure strict positiveness and $\Psi J(a)$ is the pseudo-Hessian, derived using the Gauss-Newton approximation as

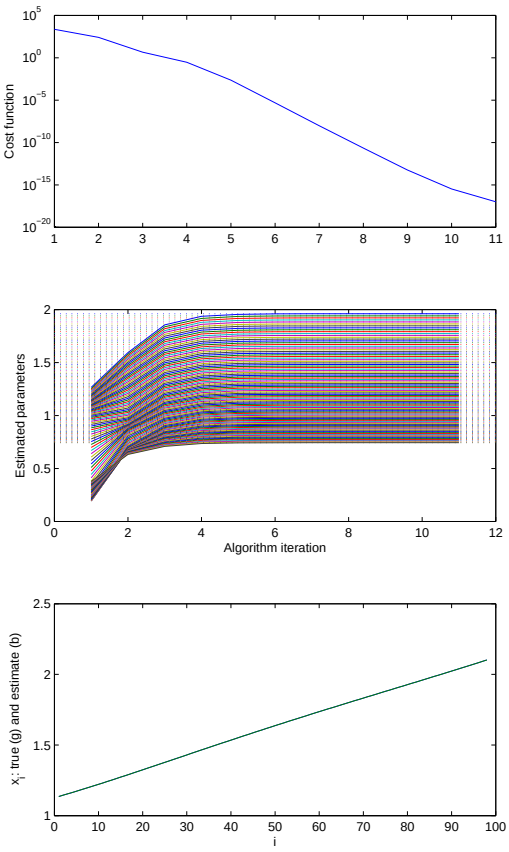
$$\Psi J = \frac{\partial y}{\partial a}(t_f)^T Q \frac{\partial y}{\partial a}(t_f) + \frac{\partial^2}{\partial a^2}\text{Reg}(a) \tag{23}$$

Remark 1: The convergence of the previous algorithm, commonly used in least square problems, is ensured from the fact that $\Psi J(a) \geq 0$ and the use of the positive constant υ to compensate the singularity point $\Psi J(\cdot) = 0$.

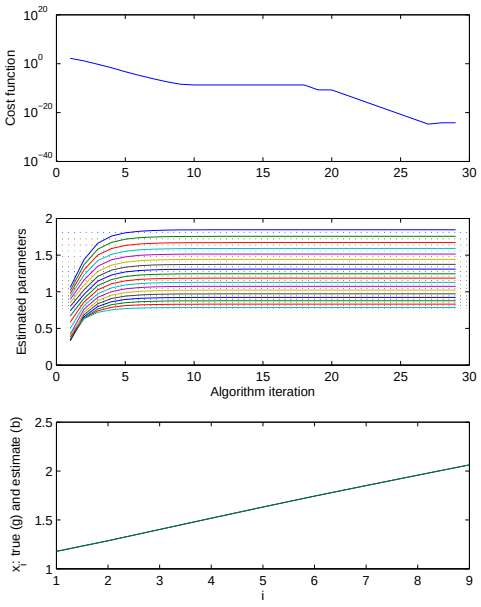
The simulation results for the diffusion example are presented in Fig. 2(a) and Fig. 2(b), without any use of a regularization function (to check the robustness while maximizing the convergence speed). The regularization becomes necessary when we have more parameters to identify than state measurements. Using a rugosity-based approach, we obtain the results presented in Fig. 2(c) and Fig. 2(d). Note that the rugosity coefficient has to be increased when the number of measurements is reduced. Also, reducing the rugosity when we get closer to the solution decreases the cost function but does not reduced the difference between the estimated and true parameters (sequences where the cost function becomes flat and then decreases again on the figures, which corresponds to iterations where R is divided by 100).



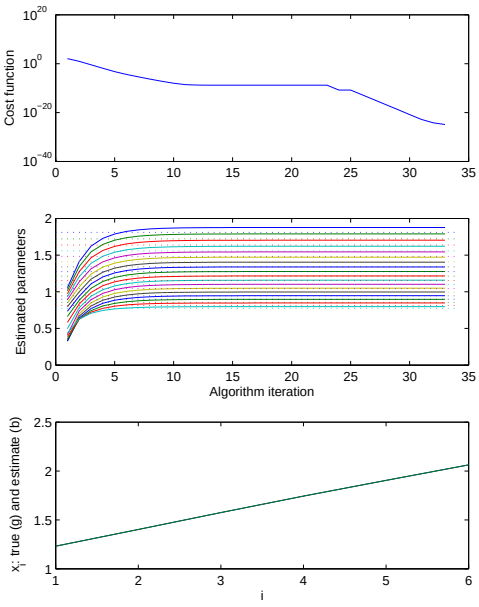
(a) 8 states, no regularization



(b) 98 states, no regularization



(c) 18 states, $n_{meas} = n_{param}/2$, $\searrow$ rugosity



(d) 18 states, $n_{meas} = n_{param}/3$, $\searrow$ rugosity

Fig. 2. Diffusivity estimation example (8 states) with the gradient computed based on the sensitivity of the state with respect to the parameter.