Symbolic Regression for Prognosis with application on ozone forecast based on the Toar database

MARKUS ABEL

Ambrosys GmbH

Potsdam University

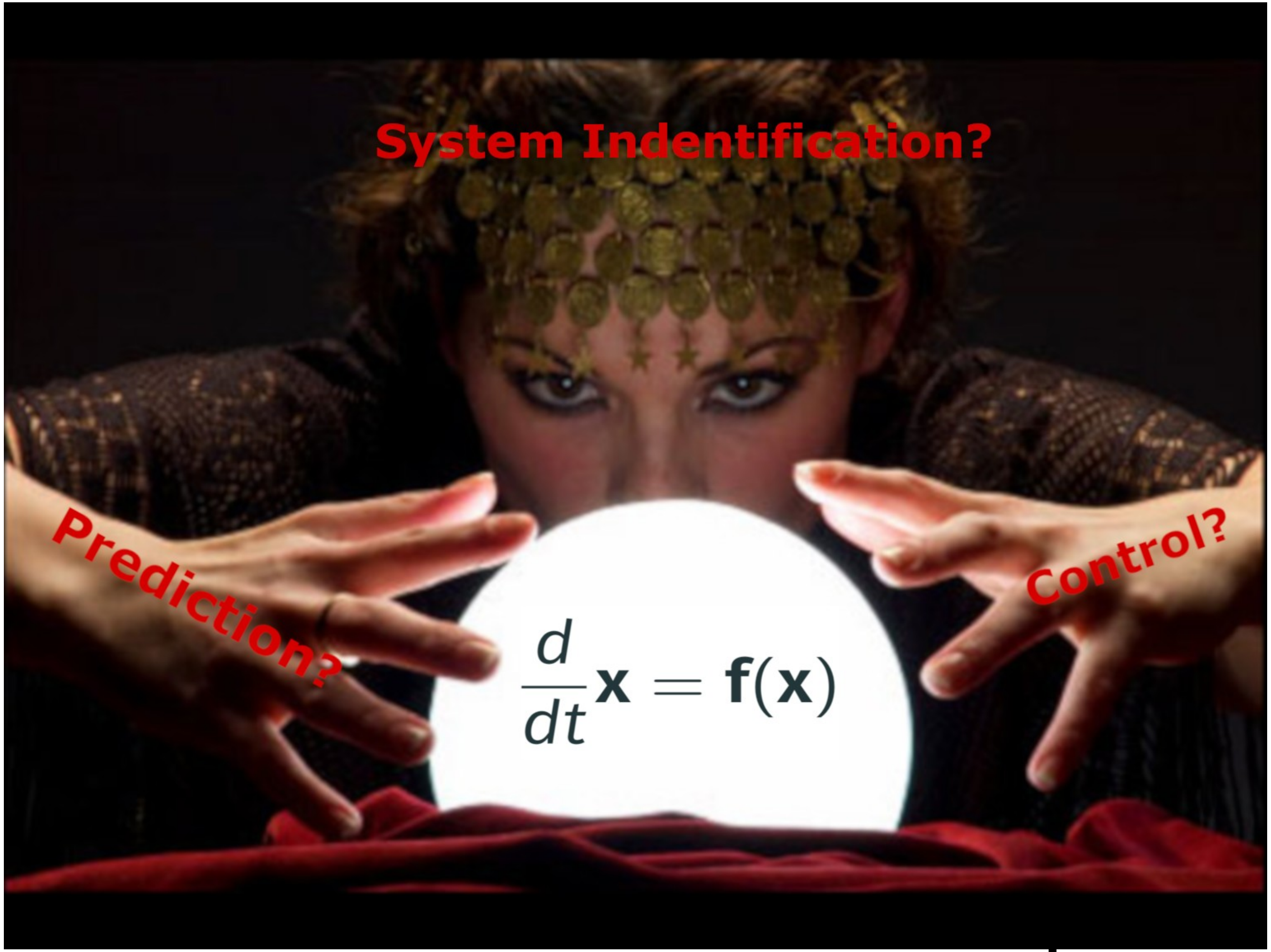
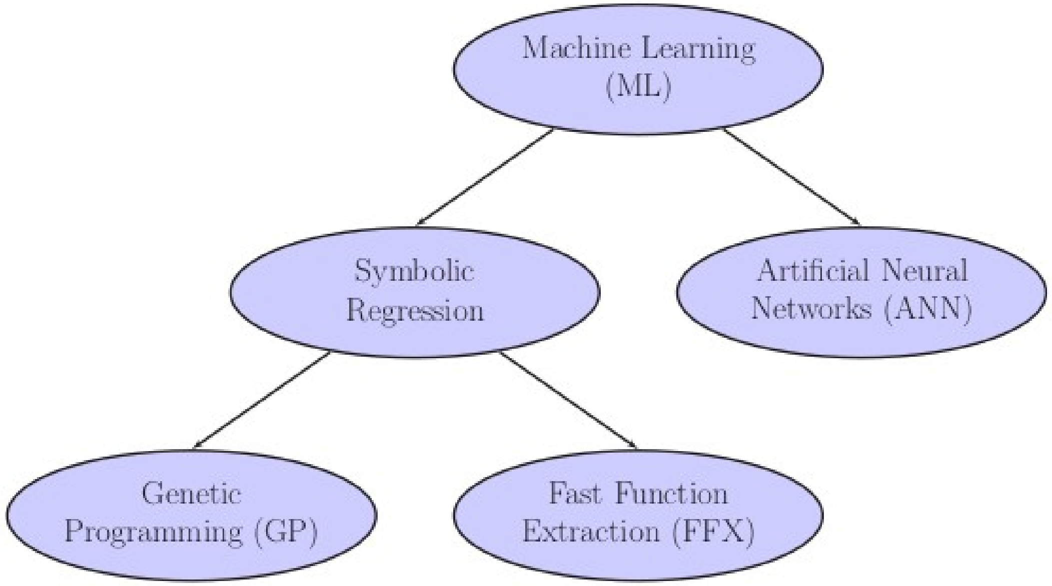


Image with kind permission of Markus Quade

SYMBOLIC REGRESSION ONTOLOGY



The idea of Symbolic Regression is

- To find relations
- To find functional dependencies
- To allow rigorous analysis

WHY SYMBOLIC REGRESSION

Equations describe physical behaviour

- Find Equations directly from data
- Use suitable input variables (= features)
- Causality can be used
- Constraints and multiobjectivity
- Prediction, Control and
- Preprocessing of results by rigorous analysis is possible

$$\frac{\partial \rho}{\partial t} + \frac{\partial}{\partial x_j}\left[\rho u_j\right] = 0$$

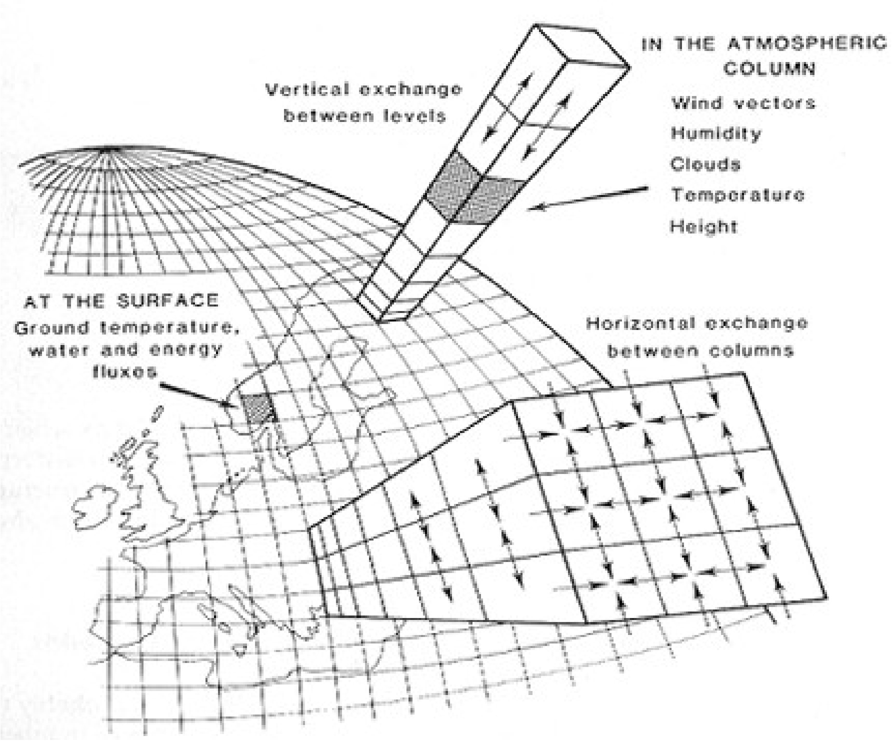
$$\frac{\partial}{\partial t}(\rho u_i) + \frac{\partial}{\partial x_j}\left[\rho u_i u_j + p\delta_{ij} - \tau_{ji}\right] = 0, \quad i = 1,2,3$$

$$\frac{\partial}{\partial t}(\rho e_0) + \frac{\partial}{\partial x_j}\left[\rho u_j e_0 + u_j p + q_j - u_i \tau_{ij}\right] = 0$$

Force balance
or
Conservation Laws

Can be built in

WHY SYMBOLIC REGRESSION IN CLIMATE



$$\frac{\partial \rho}{\partial t} + \frac{\partial}{\partial x_j}[\rho u_j] = 0$$

$$\frac{\partial}{\partial t}(\rho u_i) + \frac{\partial}{\partial x_j}[\rho u_i u_j + p\delta_{ij} - \tau_{ji}] = 0, \quad i = 1,2,3$$

$$\frac{\partial}{\partial t}(\rho e_0) + \frac{\partial}{\partial x_j}[\rho u_j e_0 + u_j p + q_j - u_i \tau_{ij}] = 0$$

Truncated Equations → Closure Problem

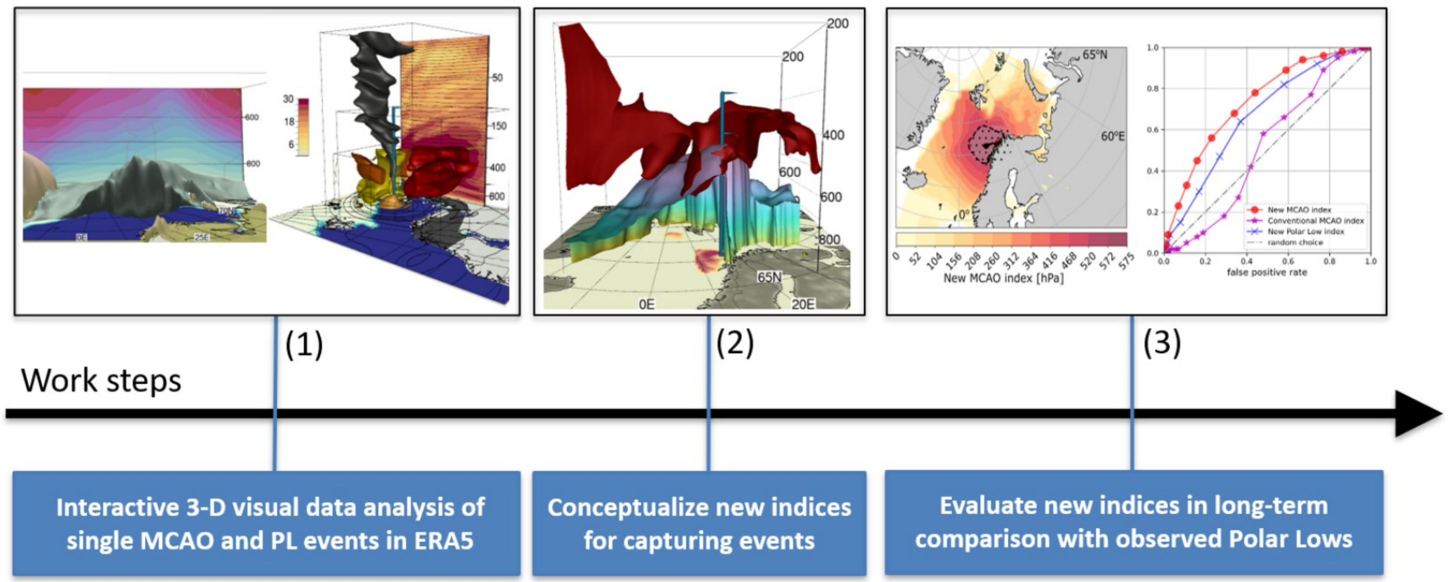
Different Equations?

Predictions based on ML → Stability ?

Control of climate → what is the optimum?

ambrosys

WHY SYMBOLIC REGRESSION IN CLIMATE



- Equations from data:
- Compute fields
- Run Regression schemes
- Obtain Equations
- Quantify noise

Predict, Control, Improve

From Meyer et al https://doi.org/10.5194/wcd-2021-20

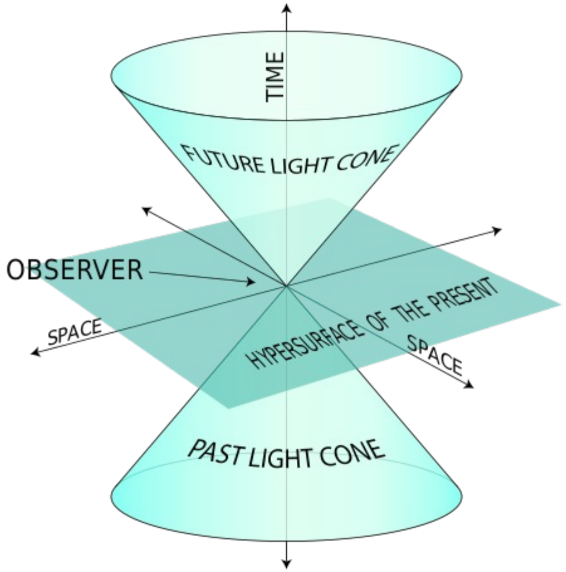
SYMBOLIC REGRESSION AND CAUSALITY

Without sophistications, we have a narrow understanding

If a system is initialized in state A, its evolution is described by physical laws in form of equations

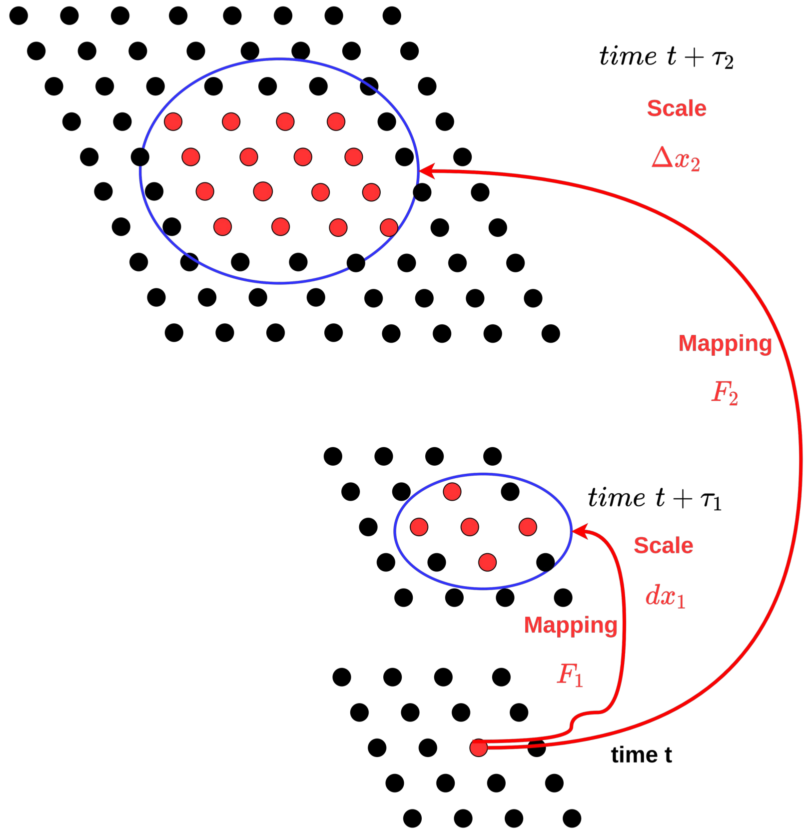
Local “coupling”
Global “coupling”

Differential equations are mainly local
Difference equations are still local in a coarse grained sense

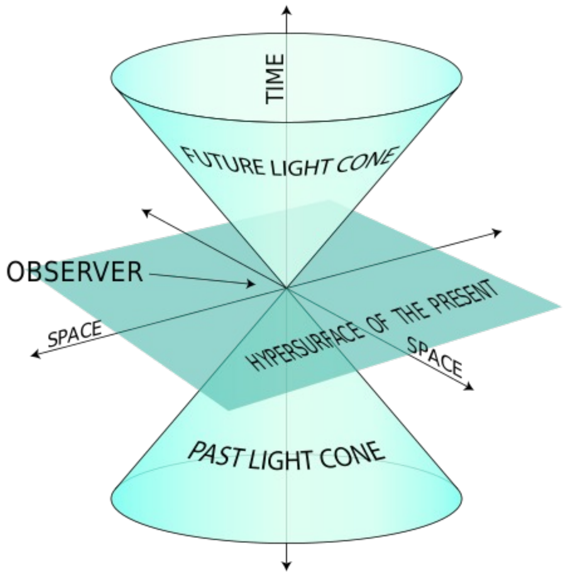


Graphics from wikipedia.org

SYMBOLIC REGRESSION AND CAUSALITY



cf. Andreas Gerhardus et al.



Find the map s(x,t) from t to T → *Propagator*

Find the law to map s(x,t) from t to T → Difference Equations

SYMBOLIC REGRESSION ANSATZ

Find an equation of the form

$$x(t+\tau)=F(x,\nabla x,\ldots)$$

or

$$\frac{\partial x}{\partial t}=F(x,\nabla x,\ldots)$$

Where x can be a vector

Holds for local conditioning

For nondifferential formulation:

differentials → differences

SYMBOLIC REGRESSION ANSATZ

Noisy data or unexplained sub(grid)-scales

$$x(t+\tau)=F(x,\nabla x,\ldots)+\epsilon$$

or

$$\frac{\partial x}{\partial t}=F(x,\nabla x,\ldots)+\epsilon$$

Assumptions:

- noise is identical in space and time
- stationary system
- no spatial variance

Holds for local conditioning

For nondifferential formulation
differentials → differences

CONCRETE METHODS: GENERALIZED REGRESSION

Let us simplify for a second.
consider the optimization
problem

$$y = F(x_1, x_2, x_3 ....)$$

$$min_a \{ \frac{1}{N} \sum_i (y_i - a_0 - x_i^T a)^2 \}$$

with condition $\sum_i |a_i| \leq t$, t is a regularizer

or determine

$$\min_a ||y - a_0 - ax||_2^2 + \lambda ||a||_1$$

I drop mathematical rigor in favor of readability

Regularization yields minimal number of coefficients:

Lasso:

$||a|| = \sum_{m,n} |a_{mn}| \leq t$, t is a regularizer

Ridge:

$||a||_2^2$ is used for regularizing

Elastic net:

$\lambda_1 ||a||_1 + \lambda_2 ||a||_2^2$ is used for regularizing

GENERALIZED LINEAR REGRESSION: EXAMPLE

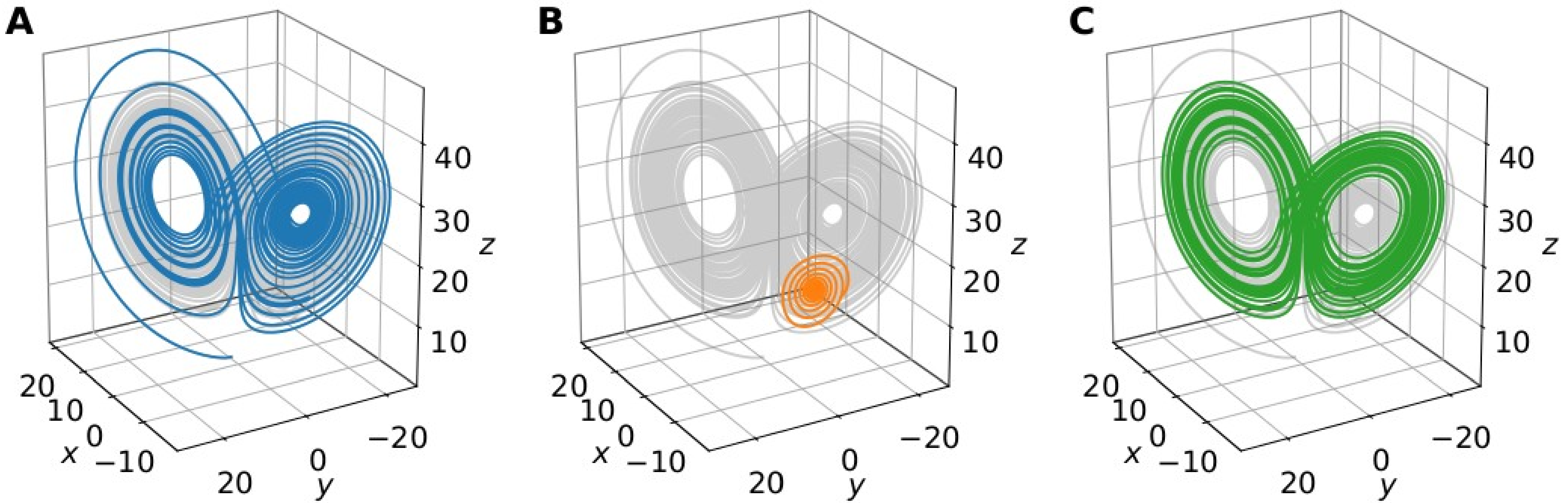


Figure 2.5: Lorenz system: Colors and parameters as in Fig. 2.4. In **A**, **B**, and **C** we show the first, second, and third segments of the trajectory in color with the concatenated trajectory in grey. The system changes from a butterfly attractor to a stable fixed point and back to a butterfly attractor.

t_{detected}	t_{update}	Equations
0.00	10.0	$\dot{x} = -10.0x + 10.0y$ $\dot{y} = 27.96x - 0.99y - 1.0xz$ $\dot{z} = -2.67z + 1.0xy$
40.01	41.0	$\dot{x} = -10.0x + 10.0y$ $\dot{y} = 15.0x - 1.0y - 1.0xz$ $\dot{z} = -2.67z + 1.0xy$
80.02	81.0	$\dot{x} = -10.0x + 10.0y$ $\dot{y} = 27.98x - 1.0y - 1.0xz$ $\dot{z} = -2.67z + 1.0xy$

Lorenz system: detection and update times, along with identified equations. The detection time coincides up to the second digit with the true switching time. The rapidly identified model agrees well with the true model structure and parameters. Coefficients are rounded to the second digit.

METHODS: GENETIC PROGRAMMING

Conventional Regression: search for coefficients in

$\mathbb{R}^N$, where N is the max. number of coefficients

Generalized Regression: search for coefficients in the space of functions with certain properties, e.g. square integrable, etc.

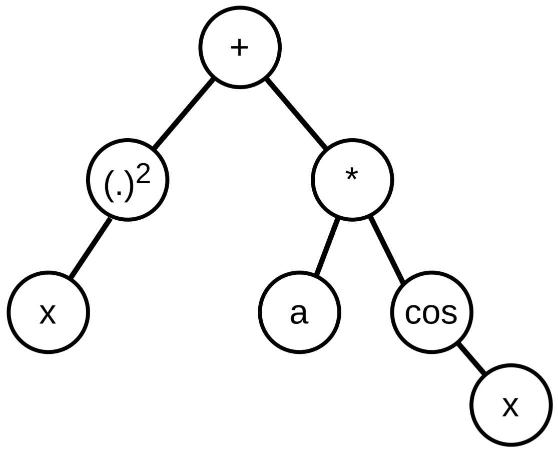
The optimization for the example problem reads then:

$$min_{f \in N} \| y - f(x) \|$$

How to search the space of functions direct, instead through power series?

Represent functions as trees

Representations are not unique



$$f(x) = x^2 + a\cos(x)$$

METHODS: GENETIC PROGRAMMING

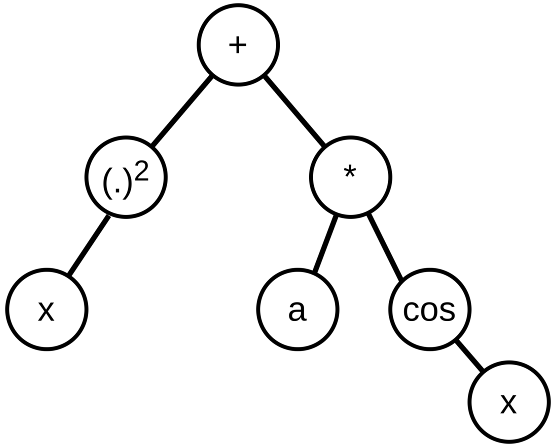
Search is stochastic with genetic principles:

Survival of the fittest:

- Create “population” as sample of function space
- Select best functions according to score
- Mutate the trees
- Crossover trees
- Iterate

Represent functions as trees

Representations are not unique



$$f(x) = x^2 + a\cos(x)$$

METHODS: GENETIC PROGRAMMING

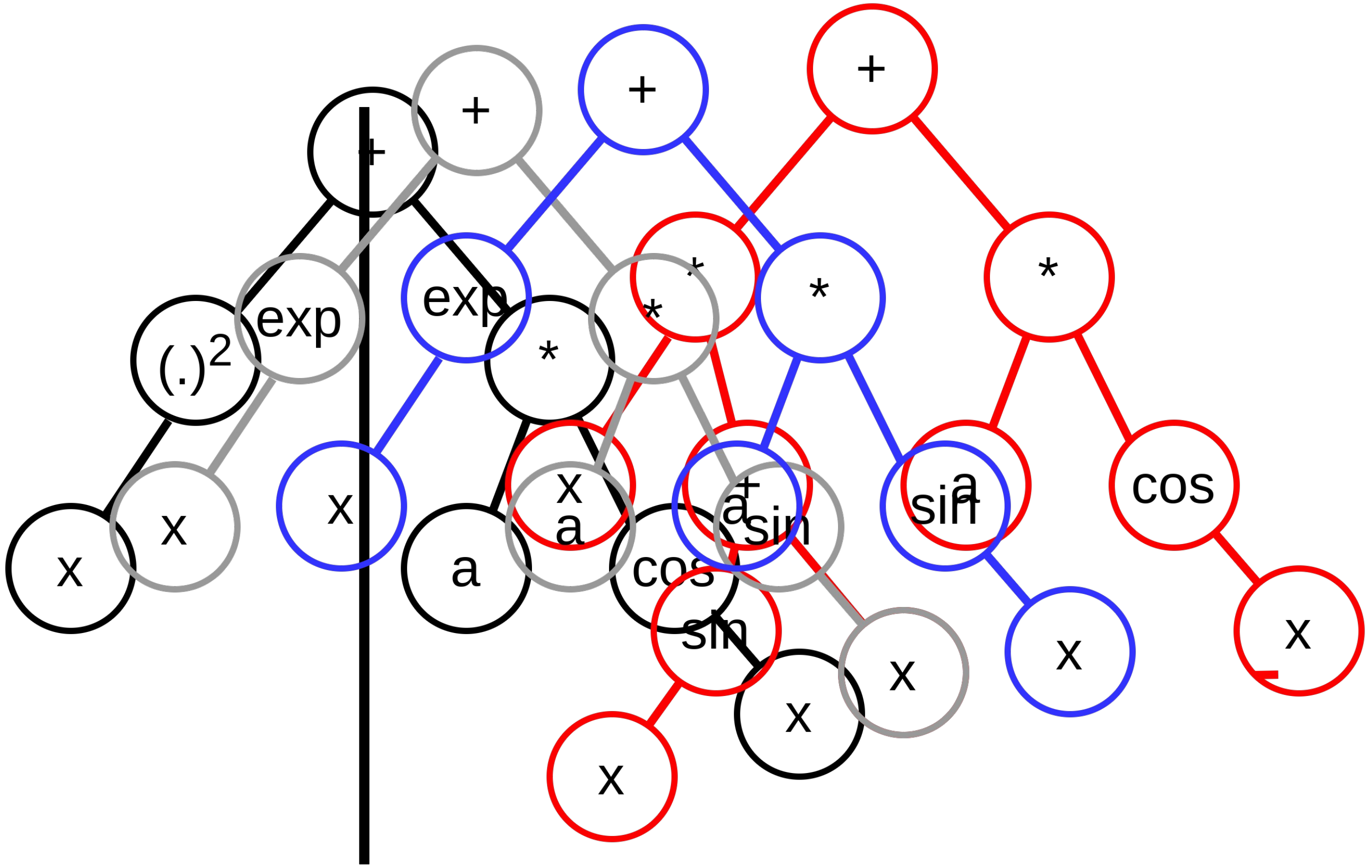
Search is stochastic with genetic principles:

Survival of the fittest:

- **Create “population”**
- Select best functions according to score
- Mutate the trees
- Crossover trees
- Iterate

Represent functions as trees

Representations are not unique



METHODS: GENETIC PROGRAMMING

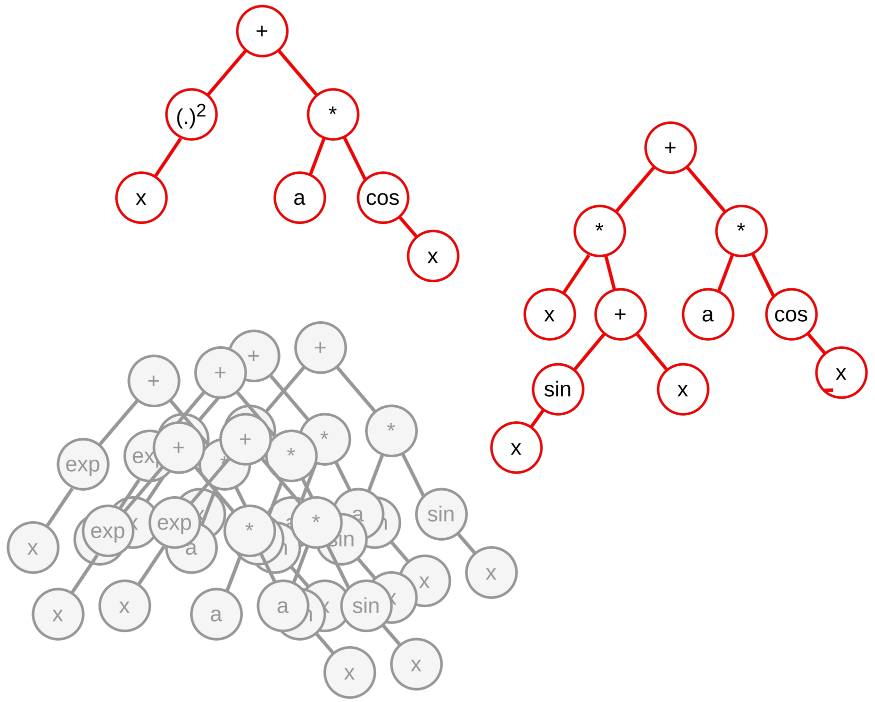
Search is stochastic with genetic principles:

Survival of the fittest:

- Create “population” as sample of function space
- **Select best functions according to score**
- Mutate the trees
- Crossover trees
- Iterate

Represent functions as trees

Representations are not unique

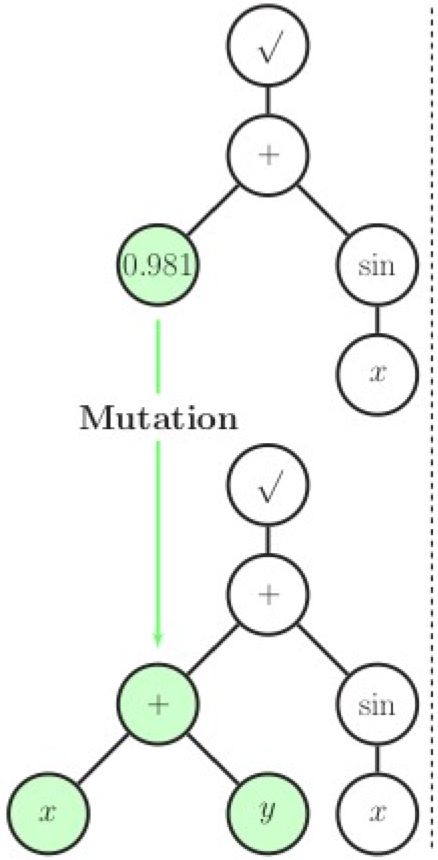


METHODS: GENETIC PROGRAMMING

Search is stochastic with genetic principles:

Survival of the fittest:

- Create “population” as sample of function space
- Select best functions according to score
- **Mutate the trees**
- Crossover trees
- Iterate

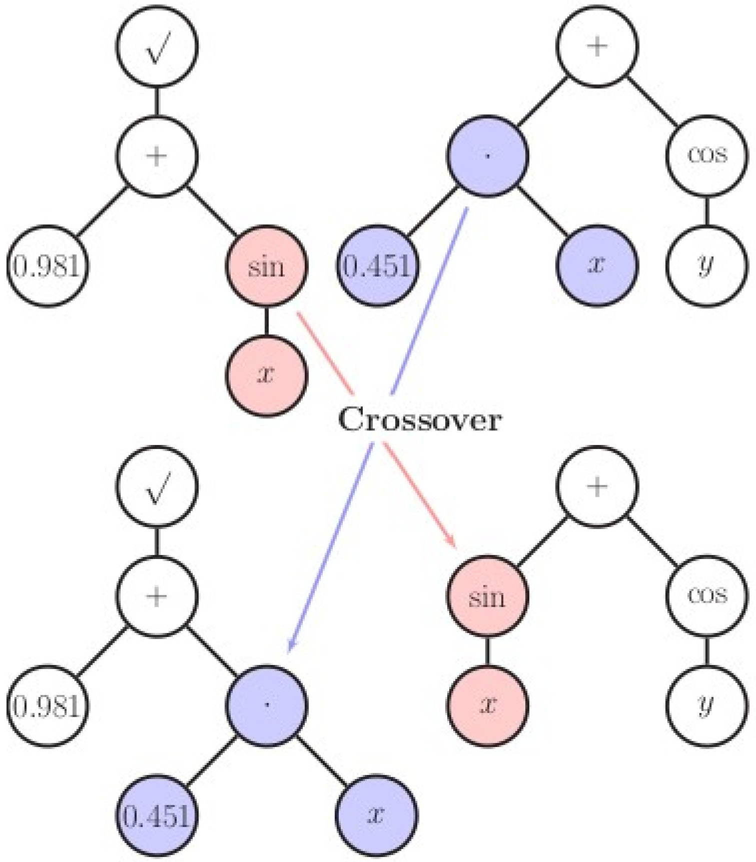


METHODS: GENETIC PROGRAMMING

Search is stochastic with genetic principles:

Survival of the fittest:

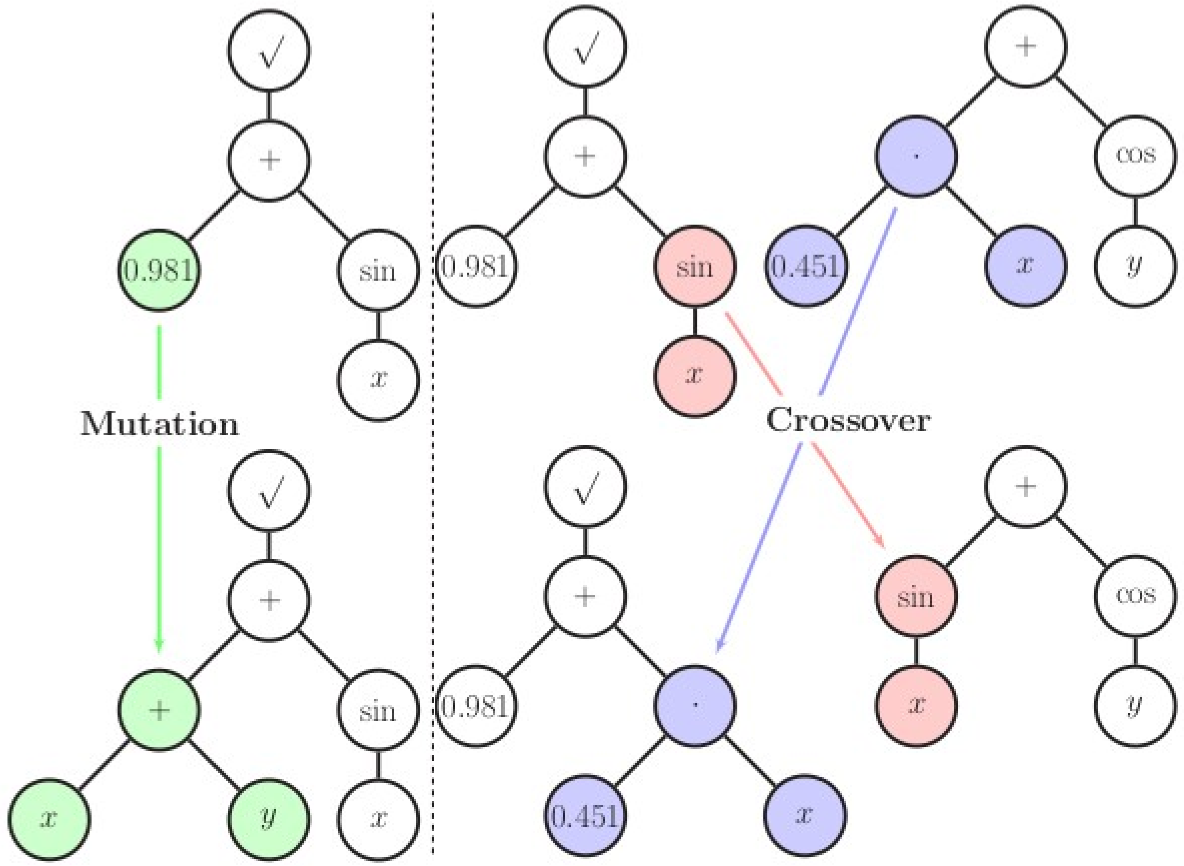
- Create “population” as sample of function space
- Select best functions according to score
- Mutate the trees
- **Crossover trees**
- Iterate: tournament algorithm



METHODS: GENETIC PROGRAMMING

Algorithm 1 Top level description of a GP algorithm

procedure MAIN
  $G_0 \leftarrow \text{random}(\lambda)$
  $\text{evaluate}(G_0)$
  $t \leftarrow 1$
  repeat
    $O_t \leftarrow \text{breed}(G_{t-1}, \lambda)$
    $\text{evaluate}(O_t)$
    $G_t \leftarrow \text{select}(O_t, G_{t-1}, \mu)$
    $t \leftarrow t + 1$
  until $t > T$ **or** $G_t = \text{good}()$
end procedure



GENETIC PROGRAMMING VS. NEURAL NETWORKS

Both Represent a map input → target data

Network structure allows very flexible design of data processing

Graph structure as well

Difference: complexity is much higher in Genetic Programming, since functions are mixed

Networks may be big to represent simple functions (e.g. 100 neurons for a sine function),but they are universal

Genetic graphs converge slow, gradients are not applicable

GP is handy for equations,
It is not useful for unknown mappings

Image recognition → no good with GP
Classification→ only for particular problems with GP (more effort)

GP can grow to huge complex models
ANN have fixed complexity (in the network)

→ GP needs to be stopped after certain depth → restriction in complexity

Key in all symbolic regression is the “library”

LATENT VARIABLES

Both Represent a map input → target data

Network structure allows very flexible design of data processing

Graph structure as well

Difference: complexity is much higher in Genetic Programming, since functions are mixed

Networks may be big to represent simple functions (e.g. 100 neurons for a sine function),but they are universal

Genetic graphs converge slow, gradients are not applicable

GENETIC PROGRAMMING: FRONT PROPAGATION

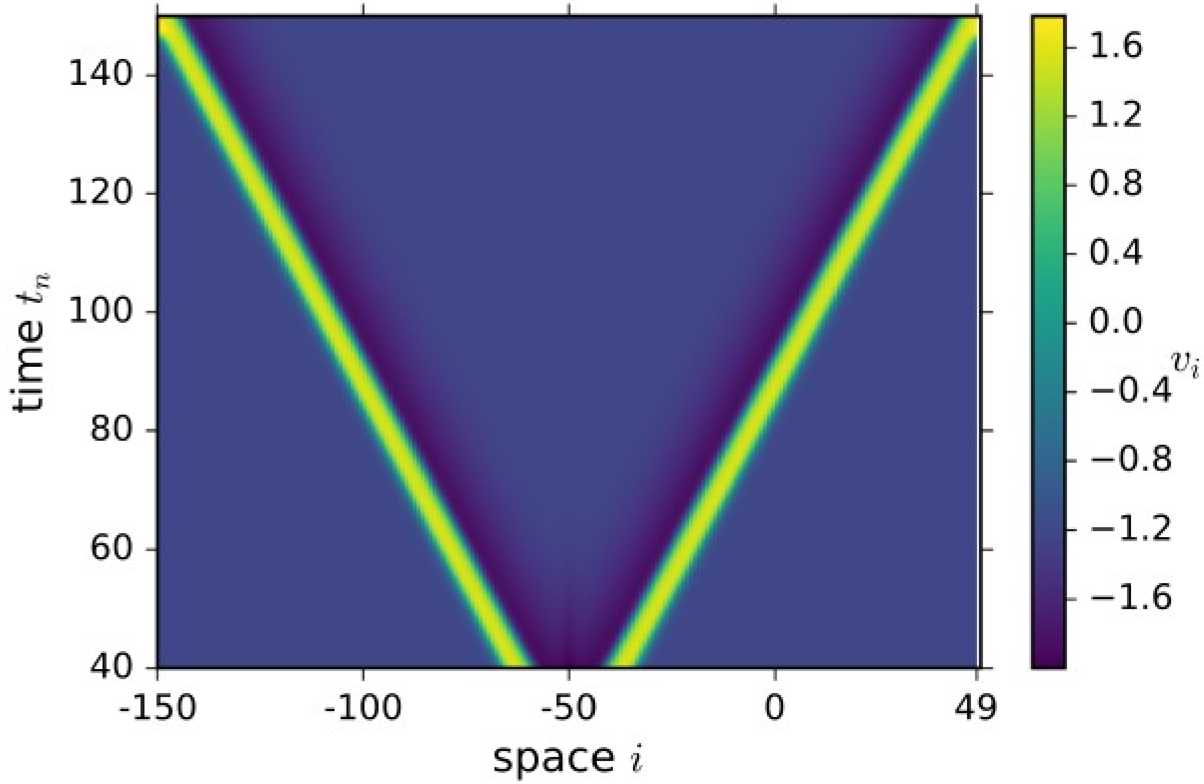
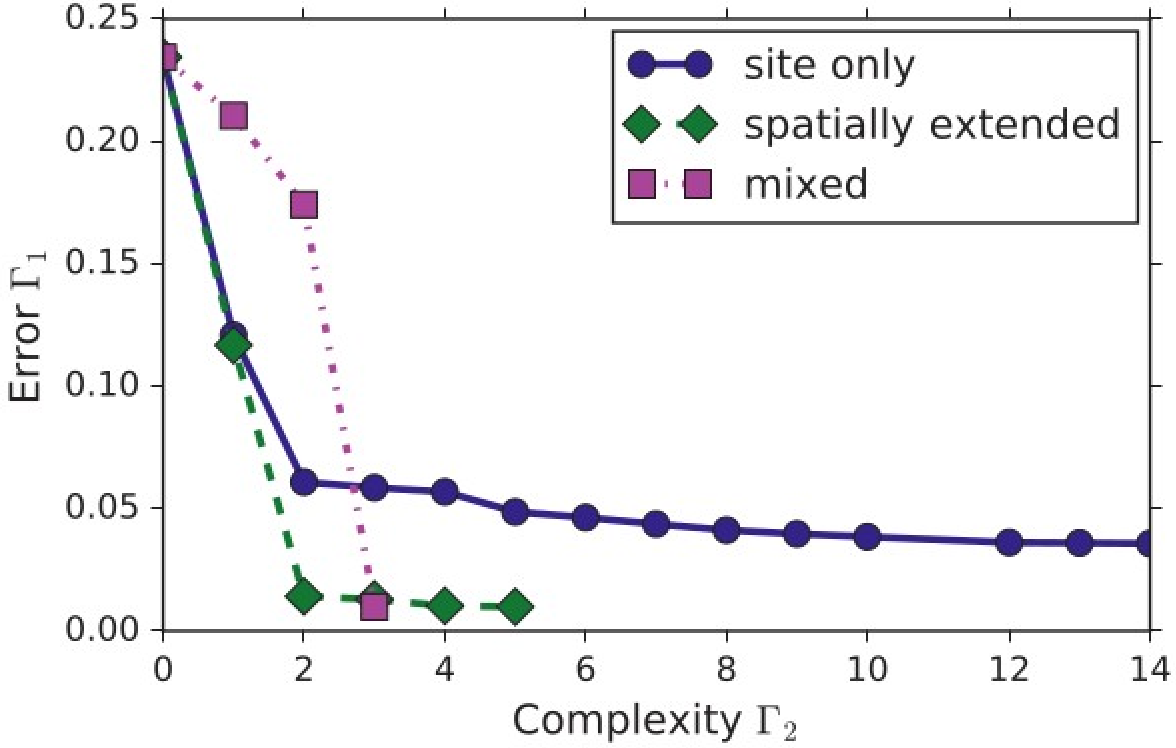


FIG. 7. Space-time plot of the pulse evolution. v_i is color coded. The front velocity is $v_f = 1.28$. Pulse width (full width half maximum) $\tau_P = 8.4$.

$$\dot{v}_i = v_i - \frac{v_i^3}{3} - w_i + I_i + D \sum_{i,j} A_{ij}(v_j - v_i),$$

$$\dot{w}_i = \varepsilon(v_i + a - bw_i),$$

GENETIC PROGRAMMING: FRONT PROPAGATION



$$\dot{v}_i = v_i - \frac{v_i^3}{3} - w_i + I_i + D \sum_{i,j} A_{ij}(v_j - v_i),$$

$$\dot{w}_i = \varepsilon(v_i + a - bw_i),$$

Make the problem multi-objective

- small modeling error
- small complexity

GENETIC PROGRAMMING: FRONT PROPAGATION

Study results for different features

$$\dot{v}_i = v_i - \frac{v_i^3}{3} - w_i + I_i + D \sum_{i,j} A_{ij}(v_j - v_i),$$

$$\dot{w}_i = \varepsilon(v_i + a - bw_i),$$

TABLE II. Coupled spiking oscillators, method GP. Formulas of the most accurate models for seed 42.

Temporal site only	$v_{0,0}^2/[v_{0,-10} + \surd(-v_{0,-10}(v_{0,0} - v_{0,-30})\{v_{0,-30}/\sin(v_{0,-10} + v_{0,-20}) + \exp(v_{0,-30}) - [\sin(v_{0,-30})]^{1/2} + \cos[(v_{0,-30})^{1/2} v_{0,-40}]\})]$
Spatially extended	$0.208 v_{0,0} + 0.792 v_{-2,0} + 0.0274\,362\,547\,430\,272 \exp(-v_{-4,0}) \sin(v_{-2,0})$
Temporal spatial	$0.878 v_{-4,-30} + 0.124\,496 v_{-4,-40}$

GENETIC PROGRAMMING: POSTPROCESSING

(Dynamical systems) equations of form

$$\partial_t u = F(u, \nabla u)$$

Can be studied rigorously for

- Stability
- Bifurcations
- Attracting, repelling regions
- ...

GENETIC PROGRAMMING: EXPLAINABILITY AND POSTPROCESSING

$$\ddot{x}_0 = f_{\mathrm{vdP}}(x_0, \dot{x}_0) + c\,(\dot{x}_1 - \dot{x}_0) + u(\vec{x}),$$
$$\ddot{x}_1 = f_{\mathrm{vdP}}(x_1, \dot{x}_1) + c\,(\dot{x}_0 - \dot{x}_1) + u(\vec{x}).$$

obtain optimal control u by GP

Two Coupled Oscillators: Optimal solutions for forced sy

$\lvert\Omega_0 - \Omega_1\rvert$	length	expression
0.0	2	$\cos(\dot{x}_1)$
0.0	2	$\cos(\dot{x}_0)$
0.0	2	$-\dot{x}_0$
0.0	2	$\sin(\dot{x}_1)$
0.0	2	$-\dot{x}_1$
0.0	2	$\sin(\dot{x}_0)$

GENETIC PROGRAMMING:
EXPLAINABILITY AND POSTPROCESSING

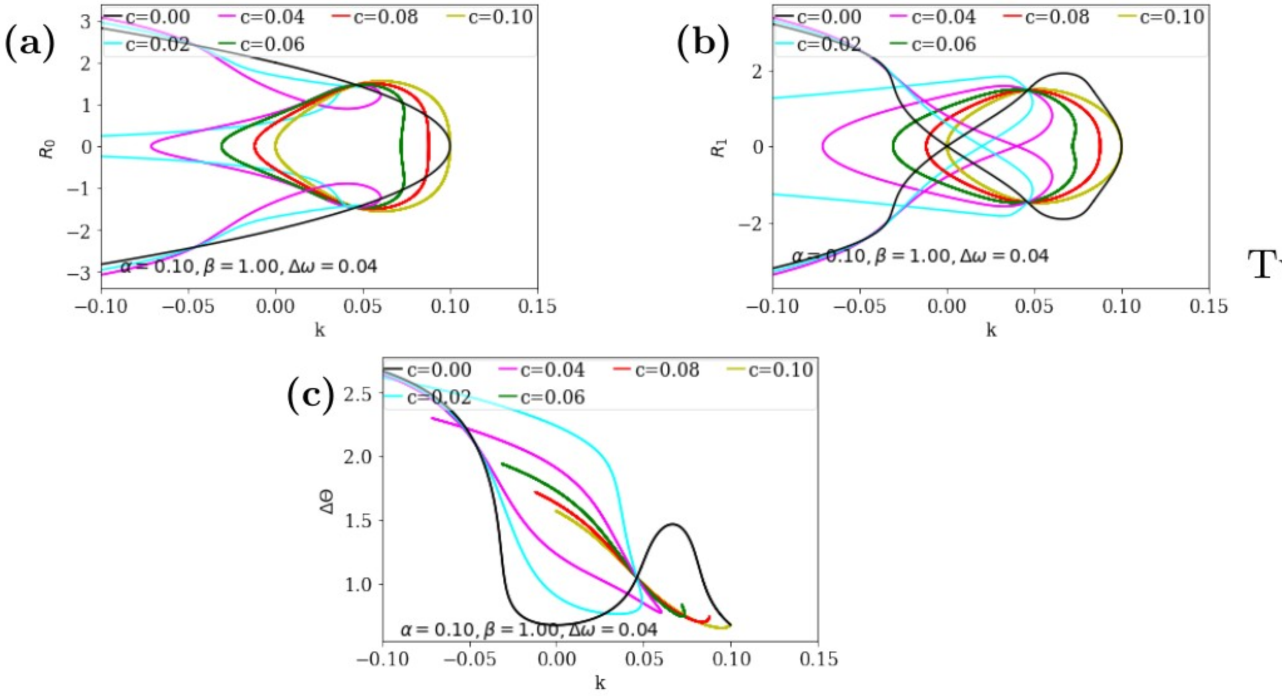


Figure 12: Stationary solutions for the coupled system (19). Solution variables **a** R_0, **b** R_1 and **c** $\Delta\Theta$ are plotted against varying control k and a set of fixed couplings c. For $c =$ (the scenario of Fig. 10 is recovered (black line). With increasing coupling, the control term becomes relatively weaker and eventually coupling dominates the dynamics.

Study stability with varying additional term found by GP solution

Two Coupled Oscillators: Optimal solutions for forced s

$\lvert\Omega_0 - \Omega_1\rvert$	length	expression
0.0	2	$\cos(\dot{x}_1)$
0.0	2	$\cos(\dot{x}_0)$
0.0	2	$-\dot{x}_0$
0.0	2	$\sin(\dot{x}_1)$
0.0	2	$-\dot{x}_1$
0.0	2	$\sin(\dot{x}_0)$

SYMBOLIC REGRESSION: OZONE DATA

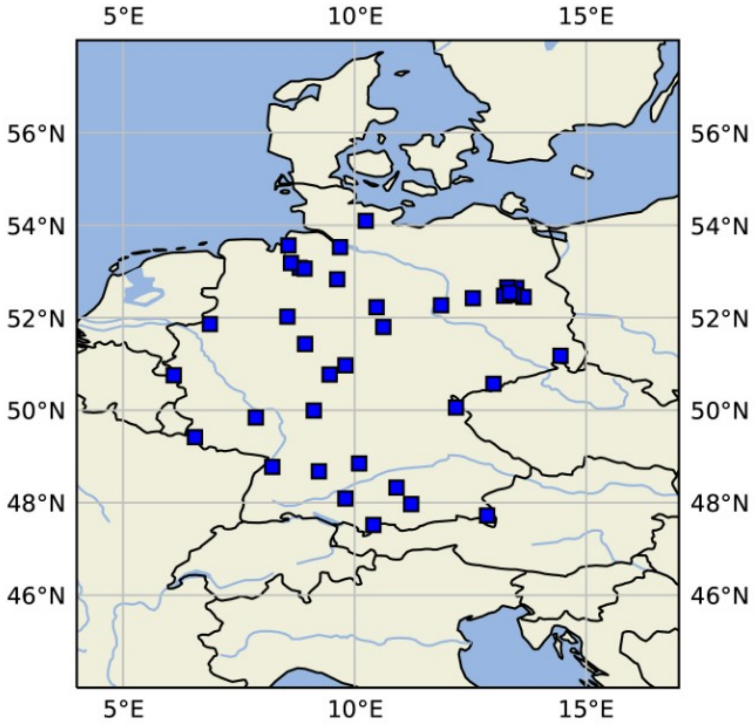


Figure 6. Map of central Europe showing the locations of some sample measurement stations as blue squares created by `PlotStationMap`.

Chemistry measurements over Germany

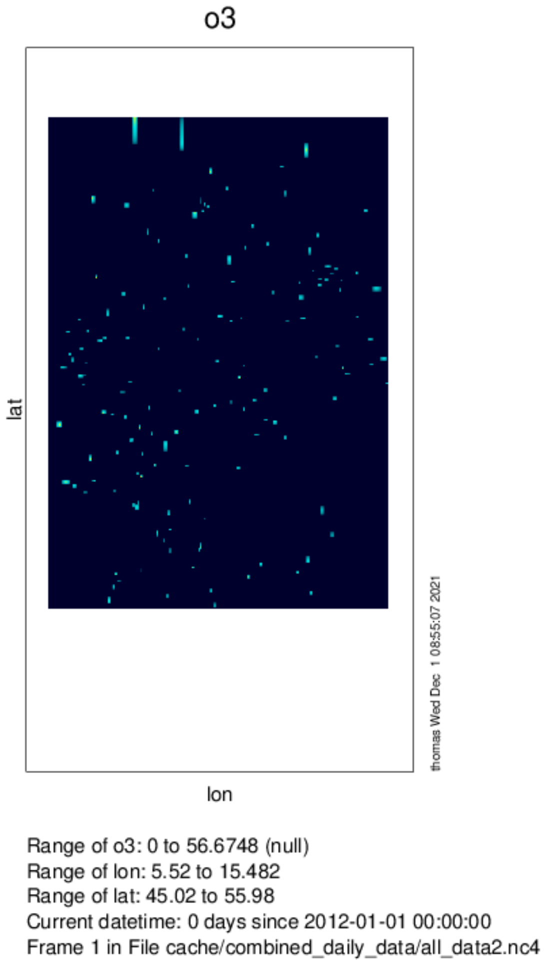
→ O_3, NO_x, …

Plus gridded DWD Icon data (v,u)

- Outside of cities
  - → daily data
  - → ~340 points in space
  - → ~2 y
  - → ~250,000 data points
- Assume spatio-temporal locality
- Compute velocity gradient from icon data
- Interpolate Ozone on grid

From Leufen et. al., https://doi.org/10.5194/gmd-14-1553-2021

SYMBOLIC REGRESSION: OZONE DATA



Simplest Model: consider ozone advection:

$$\partial_t c = \nabla (v\, c) + f(c)$$
$$\partial_t c = (v \nabla) c + (c \nabla) v + f(c)$$

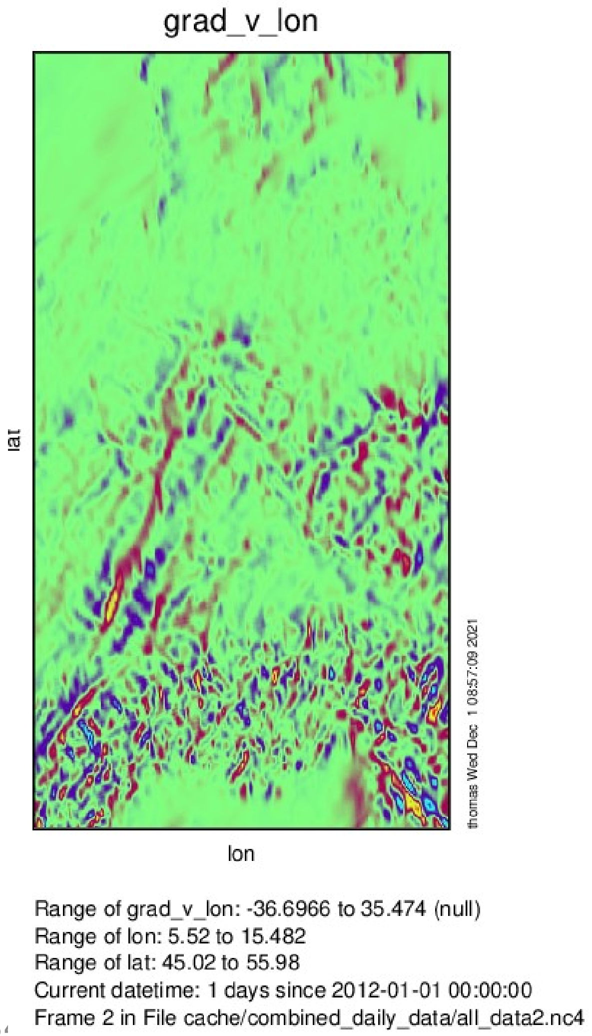
$$for\ v \in \mathbb{R}^3 : \nabla v = 0$$

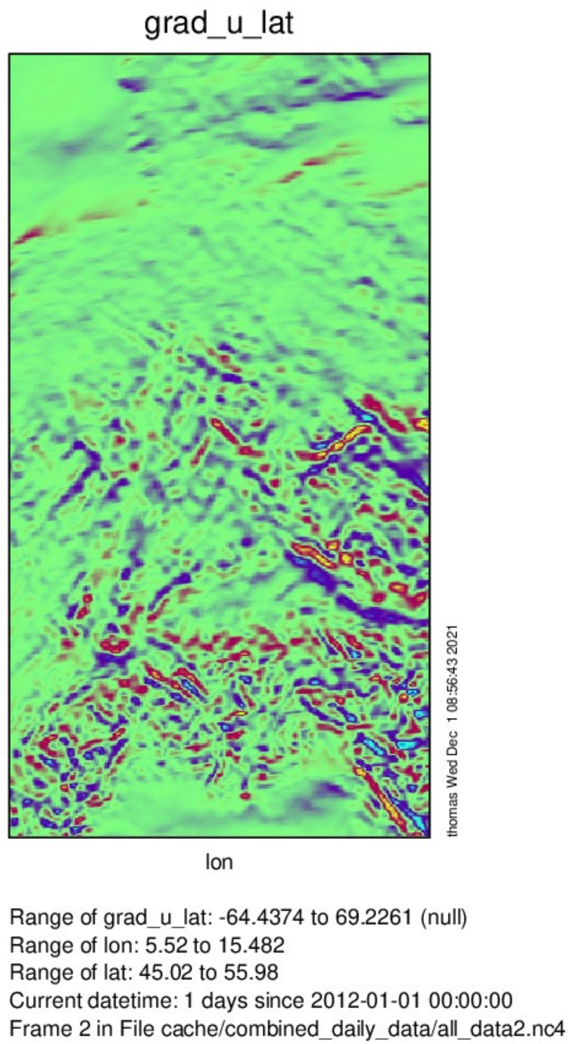
c is the ozone concentration, v velocity

More complex chemistry described by a vector of chemistry components

The result in terms of coefficients of a power series is

SYMBOLIC REGRESSION: OZONE DATA





Simplest Model: consider ozone advection:

$$\partial_t c = \nabla(v c) + f(c)$$
$$\partial_t c = (v \nabla) c + (c \nabla) v + f(c)$$

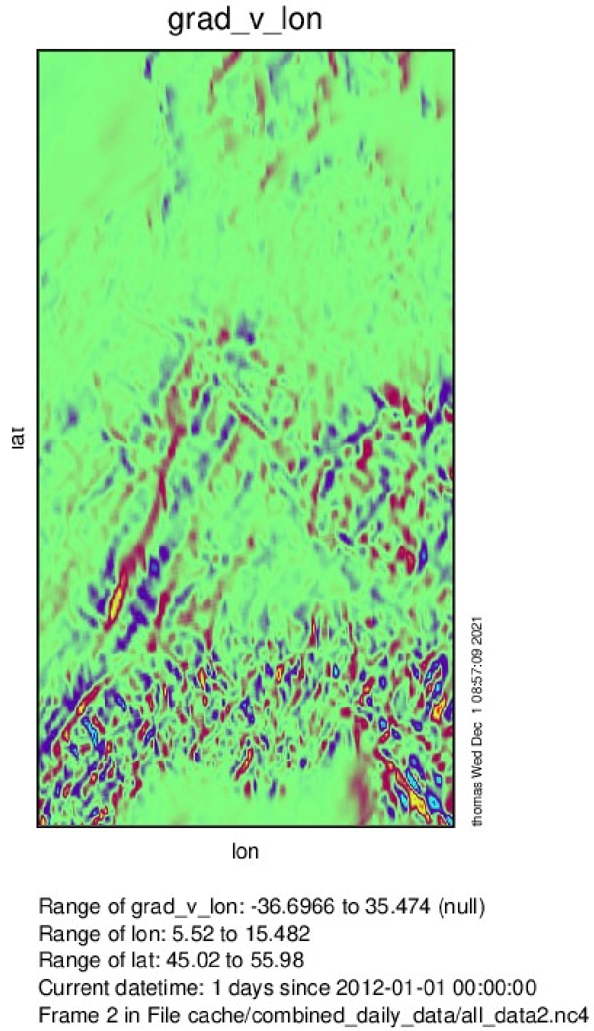
$$for\ v \in \mathbb{R}^3 : \nabla v = 0$$

Preprocessing:

- compute div v on grid
- Interpolate ozone data to grid
- Remove NANs

- Scale variables to mean zero and standard deviation 1
- Run Lasso

SYMBOLIC REGRESSION: DATA PROCESSING



COSMO data

- Wind and irradiance fromDWD (`U_10M` and `V_10m`, `SWDIFDS_RAD.2D`)
- Transform from rotated grid to lat lon grid, resolution of 0.055°x0.055° (roughly 6km x 6km) according to COSMO guidelines.
- Merge wind data for given timespan (1995 -2014).
- Calculate "differentials" as finite differences, for latitude / longitude
- Calculate daily mean values

TOAR data

- Request all timeseries IDs for non-urban stations in Germany
- Download for 1990-2020 with aggregation over 8hrs daily max
- Project to COSMO grid
- Compute "time derivative" as finite difference (normalized to 1 day)
- Regrid (coarsen to 60km) to generate nearest neighbour pairs
- Compute "gradient" as finite difference

Merge data

- Interpolate all data to grid (roughly 60 km x 60 km)
- merge data and drop NaN

SYMBOLIC REGRESSION: OZONE DATA

Mixed powers for c and div v:

[(0, 0), (0, 1), (0, 2), (1, 0), (1, 1), (1, 2), (2, 0), (2, 1), (2, 2)]

[0. -0. -0.0013563 -0. -0. -0.0005478 0.00038443 -0.00086633 0.00028896]

Coefficients for $(\nabla v)^2, \ , c(\nabla v)^2, \ , c^2, \ , c^2 \nabla v, c^2, (\nabla v)^2$

Are nonzero.

The coefficient is basically zero → **why?**

Simplest Model: consider ozone advection:

$$\partial_t c = \nabla (v c) + f(c)$$
$$\partial_t c = (v \nabla) c + (c \nabla) v + f(c)$$

$$\mathit{for}\ v \in \mathbb{R}^3 : \nabla v = 0$$

Preprocessing:

- compute div v on grid
- Interpolate ozone data to grid
- Remove NANs

- Scale variables to mean zero and standard deviation 1
- Run Lasso

SYMBOLIC REGRESSION: AVERAGING AND THE CLOSURE PROBLEM

Averaging over one day:

$$\partial_t c = \nabla(v\,c) + f(c)$$
$$\langle \partial_t c \rangle = \langle \nabla (v\,c) \rangle + \langle f(c) \rangle$$
$$\partial_t \langle c \rangle = \nabla \langle v\,c \rangle + \langle f(c) \rangle$$

$$with\, \langle . \rangle = \int_T (.)$$

This is a closure problem: find <vc> in terms of <v> and <c>

- Luckily, there exists theory on passive (and active) scalar advection.

Physics:
- which timescales are involved?
- Which length scales are involved?

ACTIVE SCALAR ADVECTION: SCALES

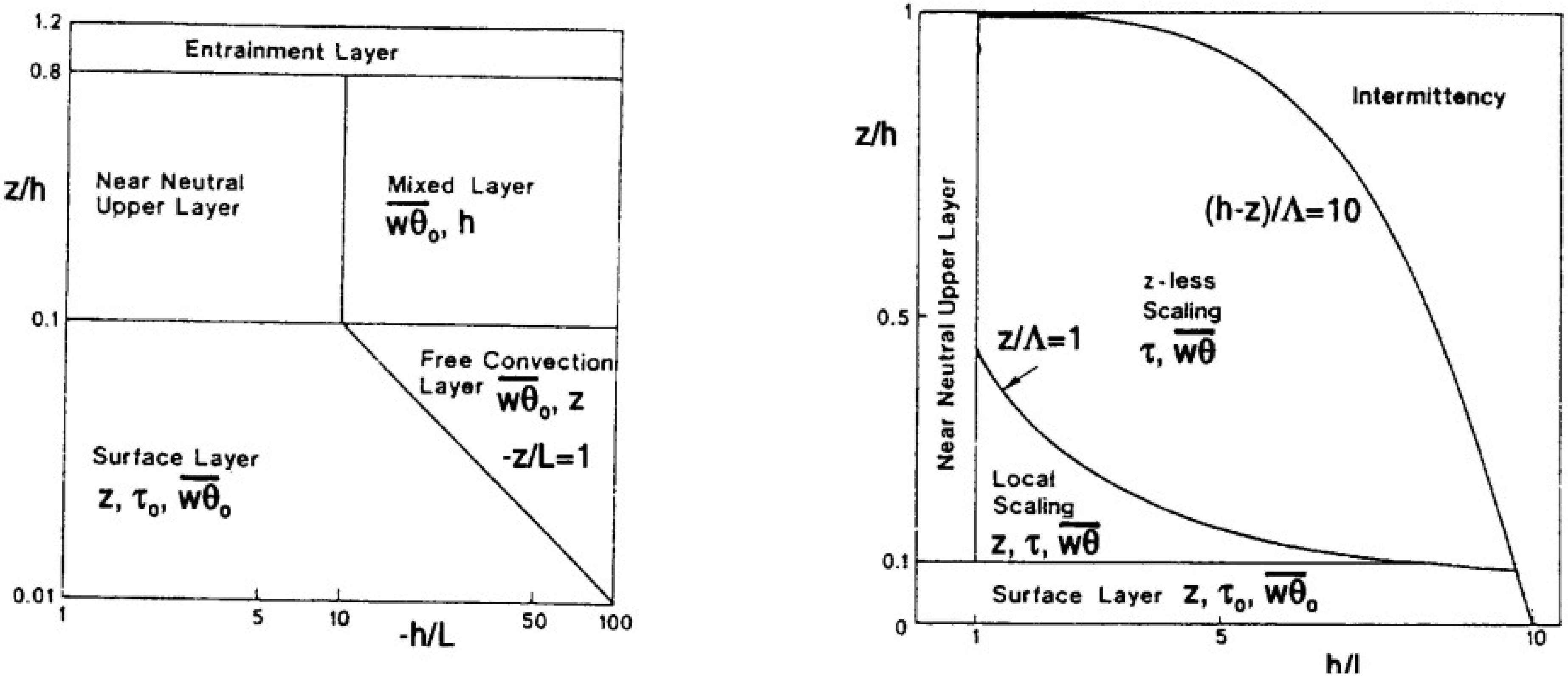


Fig. 1. Scaling regions in the unstable ($\overline{w'\theta'}_o > 0$) (above) and stable ($\overline{w'\theta'}_o < 0$) (below) boundary layer; in each scaling region the characteristic parameters have been indicated (the buoyancy parameter (g/T_0) has been omitted as scaling parameter); the lines denote the boundaries between the various scaling regimes (Holtslag and Nieuwstadt, 1986).

F.T.M. Nieuwstadt, P.G. Duynkerke / Atmospheric Research 40 (1996) 111–142

ACTIVE SCALAR ADVECTION: TIME SCALES

Forcing time (irradiance and Coriolis):

$$T \sim O(0.5\,d)$$

Boundary Layer time: $T \sim O(l/v)$

with l and v turbulence length scales

Ozone decay time $T_{Ozone} = O(1\,d) \ldots O(100\,d)$

... dependent on reactant concentration

Machine learning cannot yield better
in average

Statistics:

- For long time scales:

$$p(v,c) \sim p(v)\,p(c) \sim f(x) = e^{-\frac{(v-\langle v\rangle)^2}{2\sigma_v^2}}\, e^{-\frac{(c-\langle c\rangle)^2}{2\sigma_v^2}}$$

- Because correlation decays
- Consequence:
  - we need finer grained data
  - Or situations with higher correlation

  - For the gradient, closure by Kraichnan ‘89
    A difusion equation needs to be solved, i.e. gaussian distribution, with space and time dependent coefficients.
  - Coarse-graining: Gaussian approximation

F.T.M. Nieuwstadt, P.G. Duynkerke / Atmospheric Research 40 (1996) 111–142

ACTIVE SCALAR ADVECTION: TIME SCALES

Forcing time (irradiance and Coriolis):

$$T \sim O(0.5\,d)$$

Boundary Layer time: $T \sim O(l/v)$

with l and v turbulence length scales

Ozone decay time $T_{Ozone} = O(1\,d) \ldots O(100\,d)$

... dependent on reactant concentration

Machine learning cannot yield better *in average*

Statistics:

- For long time scales:

$$p(v,c) \sim p(v)\,p(c) \sim f(x) = \mathrm{e}^{-\frac{(v-\langle v\rangle)^2}{2\sigma_v^2}}\,\mathrm{e}^{-\frac{(c-\langle c\rangle)^2}{2\sigma_v^2}}$$

- Because correlation decays
- Consequence:
  - we need finer grained data
  - Or situations with higher correlation

  - For the gradient, closure by Kraichnan '89 A difusion equation needs to be solved, i.e. gaussian distribution, with space and time dependent coefficients.
  - Coarse-graining: Gaussian approximation

F.T.M. Nieuwstadt, P.G. Duynkerke / Atmospheric Research 40 (1996) 111–142

CONCLUSION

- Symbolic regression is suitable for equation
- Recovery from data
- Spatio-temporal data yield derivatives, gradients, divergence,…
- Noise can be found and nicely modeled
- Structured and unstructured data can be treated
- Pattern recognition is not well suited
- Constraints can be built into the objectives
- Strong point: analytical postprocessing with mathematical tools

MORE ON SYMBOLIC REGRESSION FOR DYNAMICAL SYSTEMS

PHYSICAL REVIEW E **94**, 012214 (2016)

Prediction of dynamical systems by symbolic regression

Markus Quade* and Markus Abel
Universität Potsdam, Institut für Physik und Astronomie, Karl-Liebknecht-Straße 24/25, 14476
and Ambrosys GmbH, David-Gilly-Straße 1, 14469 Potsdam, Germany

Kamran Shafi and Robert K. Niven
School of Engineering and Information Technology, University of New South Wales, Canberra ACT 2600, Australia

Sparse identification of nonlinear dynamics for rapid model re-covery

Chaos **28**, 063116 (2018); https://doi.org/10.1063/1.5027470

Markus Quade[1,2,a)], Markus Abel[1,2], J. Nathan Kutz[3], *and* Steven L. Brunton[4]

PHYSICAL REVIEW E VOLUME 57, NUMBER 3 MARCH 1998

Identification of continuous, spatiotemporal systems

H. Voss,[1] M. J. Bünner,[2] and M. Abel[1]
dam, Am Neuen Palais 10, D-14415, Potsdam, Germany
me, Nöthnitzer Strasse 38, D–01187 Dresden, Germany
30 October 1997)

Physica D: Nonlinear Phenomena
Volume 412, November 2020, 132582

Machine learning control — explainable and analyzable methods

Markus Quade [a], Thomas Isele [b], Markus Abel [a, b, c]

LETTERS 25 OCTOBE

l Binary-Fluid Convection Data

…ss*
Institut für Physik, Universität Potsdam, 14469 Potsdam, Germany

Paul Kolodner
Bell Laboratories, Lucent Technologies, Inc., Murray Hill, New Jersey 07974-0636

Markus Abel and Jürgen Kurths
Institut für Physik, Universität Potsdam, 14469 Potsdam, Germany
(Received 29 January 1999)

PASSIVE AND ACTIVE SCALAR ADVECTION

VOLUME 63, NUMBER 24 PHYSICAL REVIEW LETTERS 11 DECEMBER 1989

Probability Distribution of a Stochastically Advected Scalar Field

Hudong Chen and Shiyi Chen

Center for Nonlinear Studies, Los Alamos National Laboratory, Los Alamos, New Mexico 87545

Robert H. Kraichnan

303 Potrillo Drive, Los Alamos, New Mexico 87545

(Received 7 August 1989)

CONCLUSION

METHODS AND SKILLS

DATA SCIENCE

IT & IT-ARCHITECTURE

FRAMEWORKS, ,LIBARIES
Tensorflow, Torch, Scikit-learn, Apache Spark

GENERAL METHODS
all around time series, physical modeling, Deep learning, GANs, symbolic regression, genetic methods, nonlinear Analysis, meta-heuristics

POSITIONING & MAP
QGIS, ArcGIS, Postgis, Geo-Positioning, Navigation, Extreme values, Event-Detection, Uncertainty quantification

PREDICTION
Intraday-, short-, medium-, and longterm prediction. Predictive maintenance, nonlinear timeseries

PROGRAMMING LANGUAGES
C++, Python, , Scala, Javascript

ARCHITECTURE
service-oriented architecture, distributed data, microservices

CLOUD & CONTAINERIZING
AWS, HPC infrastructure, docker, kubernetes, singularity

DATA BASES
PostgreSQL, OracleDB, Cassandra, MongoDB, KAFKA, Prometheus

SEARCH AND OPERATIONS
Elasticsearch / Beats / Kibana, Prometheus

VISUALISATION
grafana, tableau, python

we follow clean code / clean architecture-principles and functional style