

Smart Models at MNZ@HTWK

Sara Grundel, Jochen Merker

Mathematisch-Naturwissenschaftliches Zentrum (MNZ), HTWK Leipzig
sara.grundel@htk-leipzig.de, jochen.merker@htk-leipzig.de

30.09.2026

- Which kind of models are smart?

- Which kind of models are smart?
- From abstract theory to specific applications

- Which kind of models are smart?
- From abstract theory to specific applications
- Model reduction for energy networks

Mathematical MSO - a success story

Mathematical modeling of real-world systems or processes

- fosters understanding,
- allows simulation to predict,
- allows optimization to improve.

Mathematical MSO - a success story

Mathematical modeling of real-world systems or processes

- fosters understanding,
- allows simulation to predict,
- allows optimization to improve.

Mathematical modeling, simulation and optimization (MSO) is a very successful development cycle, and numerical methods seem to have outperformed computational power in history.

Mathematical MSO - a success story

Mathematical modeling of real-world systems or processes

- fosters understanding,
- allows simulation to predict,
- allows optimization to improve.

Mathematical modeling, simulation and optimization (MSO) is a very successful development cycle, and numerical methods seem to have outperformed computational power in history.

What's new?

Increase of data availability (due to internet and sensors everywhere, or generation of data by simulation) allows training and productive use of deep learning models (DLM) and artificial intelligences (AI).

Smart models & digital twins

As an extension of MSO, smart models or digital twins

- should be hierarchical models, which can be simulated with different degrees of realism depending on use case, objective and data quality,
 - should allow interconnection to other models via interfaces sharing not only accuracies but also sensitivities or uncertainties,
 - should allow seamless integration in a data rich or AI environment,
- and all this automatically without much adaption of code^a.

^afor another list of requirements see e.g. V.Mehrmann, B.Unger: Control of port-Hamiltonian differential-algebraic systems and applications. Acta Numerica **32** (2023), 395–515.

Smart models & digital twins

As an extension of MSO, smart models or digital twins

- should be hierarchical models, which can be simulated with different degrees of realism depending on use case, objective and data quality,
 - should allow interconnection to other models via interfaces sharing not only accuracies but also sensitivities or uncertainties,
 - should allow seamless integration in a data rich or AI environment,
- and all this automatically without much adaption of code^a.

^afor another list of requirements see e.g. V.Mehrmann, B.Unger: Control of port-Hamiltonian differential-algebraic systems and applications. Acta Numerica **32** (2023), 395–515.

Example

- Importance of this research topic is stressed e.g. by increase of professorships for data-driven PDEs.

Smart models & digital twins

As an extension of MSO, smart models or digital twins

- should be hierarchical models, which can be simulated with different degrees of realism depending on use case, objective and data quality,
- should allow interconnection to other models via interfaces sharing not only accuracies but also sensitivities or uncertainties,
- should allow seamless integration in a data rich or AI environment, and all this automatically without much adaption of code^a.

^afor another list of requirements see e.g. V.Mehrmann, B.Unger: Control of port-Hamiltonian differential-algebraic systems and applications. Acta Numerica **32** (2023), 395–515.

Example

- Importance of this research topic is stressed e.g. by increase of professorships for data-driven PDEs.
- An example for effectiveness of code by multiple dispatch is e.g. seamless integration of uncertainties in ODE solvers in Julia.

Closed static systems

can mathematically often be modeled by reflexive Banach spaces X as state space and abstract stationary equations

$$Ax = b$$

with $A : X \rightarrow X^*$, $b \in X^*$,

Closed static systems

can mathematically often be modeled by reflexive Banach spaces X as state space and abstract stationary equations

$$Ax = b$$

with $A : X \rightarrow X^*$, $b \in X^*$, which in the potential case $A = \partial\Phi_A$ can be derived from a variational principle

$$\Phi_A(x) - \langle b, x \rangle = \min_x$$

Closed static systems

can mathematically often be modeled by reflexive Banach spaces X as state space and abstract stationary equations

$$Ax = b$$

with $A : X \rightarrow X^*$, $b \in X^*$, which in the potential case $A = \partial\Phi_A$ can be derived from a variational principle

$$\Phi_A(x) - \langle b, x \rangle = \min_x$$

and have solutions if A is bounded, pseudomonotone, coercive (Brézis) or if A is radially continuous, monotone, coercive (Browder-Minty).

Basic abstract theory - open static systems

Open static systems with control u and observation y

can mathematically be modeled by abstract input-output systems

$$Ax = b - Bu$$

$$y = B^*x$$

with continuous linear $B : U \rightarrow X^*$,

Open static systems with control u and observation y

can mathematically be modeled by abstract input-output systems

$$Ax = b - Bu$$

$$y = B^*x$$

with continuous linear $B : U \rightarrow X^*$, inducing in case of well-posedness an input-output map

$$Fu := B^*A^{-1}(b - Bu) = y$$

Basic abstract theory - open static systems

Open static systems with control u and observation y

can mathematically be modeled by abstract input-output systems

$$Ax = b - Bu$$

$$y = B^*x$$

with continuous linear $B : U \rightarrow X^*$, inducing in case of well-posedness an input-output map

$$Fu := B^*A^{-1}(b - Bu) = y$$

and allowing a Dirac formulation $(x, Ax - b, u, y) \in D$ with constant Dirac structure D , i.e. a maximally isotropic subspace $D = D^\perp$,

$$D = \{(x, x^*, u, y) \in X \oplus X^* \oplus U \oplus U^* \mid x^* = -Bu, y = B^*x\}$$

expressing power conservation $\langle Ax - b, x \rangle + \langle y, u \rangle = 0$.

Input-output systems

can be closed by a feedback loop $u = Ry$ leading to $(A + BRB^*)x = b$

Input-output systems

can be closed by a feedback loop $u = Ry$ leading to $(A + BRB^*)x = b$
or by closing the port $y = 0$ leading to an implicit system

$$Ax = b - Bu$$

$$0 = B^*x$$

allowing a (forward) Dirac formulation,

Input-output systems

can be closed by a feedback loop $u = Ry$ leading to $(A + BRB^*)x = b$
or by closing the port $y = 0$ leading to an implicit system

$$Ax = b - Bu$$

$$0 = B^*x$$

allowing a (forward) Dirac formulation,
or can be interconnected e.g. by parallel interconnection

$$A_1x_1 = b_1 - B_1\iota_1u$$

$$A_2x_2 = b_2 - B_2\iota_2u$$

$$y = \iota_1^*B_1^*x_1 + \iota_2^*B_2^*x_2$$

with $(\iota_1, \iota_2) : U \rightarrow U_1 \oplus U_2$ allowing a (backward) Dirac formulation.

Deep learning models (DLM)

are input-output maps F obtained from approximating or interpolating “big” training data (u_i, y_i) , $i = 1, \dots, N$, by “deep” learning, i.e. by assuming a certain shape $F = F_w$ with “many” weights w , e.g. a large feed-forward neural network, and approximately solving for a loss functional Φ

$$\frac{1}{N} \sum_{i=1}^N \Phi(F_w u_i - y_i) = \min_w !,$$

e.g. $\Phi(y) := \|y\|_{U^*}^2$ so that expected quadratic loss is minimized.

Deep learning models

Deep learning models (DLM)

are input-output maps F obtained from approximating or interpolating “big” training data (u_i, y_i) , $i = 1, \dots, N$, by “deep” learning, i.e. by assuming a certain shape $F = F_w$ with “many” weights w , e.g. a large feed-forward neural network, and approximately solving for a loss functional Φ

$$\frac{1}{N} \sum_{i=1}^N \Phi(F_w u_i - y_i) = \min_w !,$$

e.g. $\Phi(y) := \|y\|_{U^*}^2$ so that expected quadratic loss is minimized.

Example

Deep learning models F trained by simulation-generated data (u_i, y_i) can be used as surrogate models replacing simulation by predicting for an input u the output $y = Fu$.

Physics-informed neural networks (PINN)

Physics-informed neural networks (PINN)

Instead of training with simulation-generated data (u_i, y_i) , i.e.

$$\frac{1}{N} \sum_{i=1}^N \|F_w u_i - y_i\|_{U^*}^2 = \min_w \text{! subject to}$$
$$Ax_i = b - Bu_i, y_i = B^* x_i,$$

Physics-informed neural networks (PINN)

Instead of training with simulation-generated data (u_i, y_i) , i.e.

$$\frac{1}{N} \sum_{i=1}^N \|F_w u_i - y_i\|_{U^*}^2 = \min_w!$$
 subject to

$$Ax_i = b - Bu_i, \quad y_i = B^* x_i,$$

alternatively residuals may be penalized, i.e.

$$\frac{1}{N} \sum_{i=1}^N (\|F_w u_i - B^* x\|_{U^*}^2 + \gamma \|Ax - b + Bu_i\|_{X^*}^2) = \min_{w,x}!.$$

Parameter estimation

Note similarity to parameter estimation, i.e.

for $\Phi(y - \hat{y}) := \|y - \hat{y}\|^2$ to the stationary optimal control problem

$$\Phi(y - \hat{y}) + \Psi(k) = \min_k! \quad \text{subject to}$$

$$A_k x = b, \quad y = B^* x, \quad k \in K,$$

for estimation of model parameters k with regularizer Ψ^a .

^aJ.M., A.Matas: Estimation of discontinuous parameters in linear elliptic equations by a regularized inverse problem. PDE in Appl. Math. **5** (2022) 100384.

Parameter estimation

Note similarity to parameter estimation, i.e.

for $\Phi(y - \hat{y}) := \|y - \hat{y}\|^2$ to the stationary optimal control problem

$$\Phi(y - \hat{y}) + \Psi(k) = \min_k! \quad \text{subject to}$$

$$A_k x = b, \quad y = B^* x, \quad k \in K,$$

for estimation of model parameters k with regularizer Ψ^a .

^aJ.M., A.Matas: Estimation of discontinuous parameters in linear elliptic equations by a regularized inverse problem. PDE in Appl. Math. **5** (2022) 100384.

Example

Estimate thermal conductivity $k \in BV(\Omega)$ from temperature measurements $\hat{y}$ in bore holes $\Omega' \subset \Omega$ to obtain a sub-surface map via

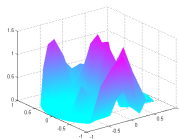
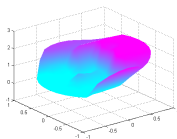
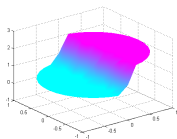
$$\|y - \hat{y}\|_{L^2(\Omega')}^2 + \gamma TV(k) = \min_k!$$

$$s.t. \quad -\operatorname{div}(k \nabla y) = b \text{ in } \Omega, \quad y = 0 \text{ on } \partial\Omega, \quad r \leq k \leq R \text{ in } \Omega.$$

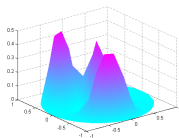
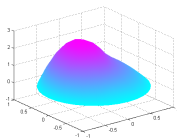
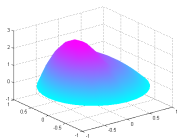
Example

Numerical solution for meshed 2D domain $\Omega := B_1(0)$, constant $b := 10$, $\gamma := 10^{-4}$ and solution $\hat{y}$ on $\Omega' := B_{\frac{1}{10}}(\pm\frac{1}{2}, 0)$ to

$$k(x, y) := \begin{cases} \frac{1}{2} & x < 0 \\ 2 & x > 0 \end{cases} \text{ for } r := \frac{1}{2}, R := 2:$$



original k , estimated $\hat{k}$, pointwise error



original u , estimated $\hat{u}$, pointwise error

Closed dynamic processes

can mathematically often be modeled on reflexive Banach spaces X, Y with a dense intersection and abstract evolution equations

$$\frac{d}{dt}Mx + Ax = b$$

with $M : Y \rightarrow Y^*$ (e.g. duality mapping), $b(t) \in X^* + Y^*$,

^aJ.M., M.Krüger: On a variational principle in thermodynamics, Continuum Mechanics and Thermodynamics **25** (2013), 779–793

Closed dynamic processes

can mathematically often be modeled on reflexive Banach spaces X, Y with a dense intersection and abstract evolution equations

$$\frac{d}{dt}Mx + Ax = b$$

with $M : Y \rightarrow Y^*$ (e.g. duality mapping), $b(t) \in X^* + Y^*$,

- dissipative for $b = 0$ due to $\frac{d}{dt}\Phi_M^*(Mx) = -\langle Ax, x \rangle \leq 0$ or $\frac{d}{dt}\Phi_A(x) = -\langle \frac{d}{dt}Mx, \frac{d}{dt}x \rangle \leq 0$ allowing a Dirac formulation $(x(t), Ax(t), M^{-1}p(t), \frac{dp}{dt}(t)) \in \{(x, x^*, y, y^*) \mid y^* = -x^*, x = y\}^a$,

^aJ.M., M.Krüger: On a variational principle in thermodynamics, Continuum Mechanics and Thermodynamics **25** (2013), 779–793

Closed dynamic processes

can mathematically often be modeled on reflexive Banach spaces X, Y with a dense intersection and abstract evolution equations

$$\frac{d}{dt}Mx + Ax = b$$

with $M : Y \rightarrow Y^*$ (e.g. duality mapping), $b(t) \in X^* + Y^*$,

- dissipative for $b = 0$ due to $\frac{d}{dt}\Phi_M^*(Mx) = -\langle Ax, x \rangle \leq 0$ or $\frac{d}{dt}\Phi_A(x) = -\langle \frac{d}{dt}Mx, \frac{d}{dt}x \rangle \leq 0$ allowing a Dirac formulation $(x(t), Ax(t), M^{-1}p(t), \frac{dp}{dt}(t)) \in \{(x, x^*, y, y^*) \mid y^* = -x^*, x = y\}^a$,
- conservative in case of a Hamiltonian equation $\frac{dx}{dt} = J dH(x)$ with linear $J = -J^*$ so that $\frac{d}{dt}H(x) = 0$ for Hamiltonian H on X , allowing a Dirac formulation $(\frac{dx}{dt}, dH(x)) \in \{(x, x^*) \mid x = Jx^*\}$.

^aJ.M., M.Krüger: On a variational principle in thermodynamics, Continuum Mechanics and Thermodynamics **25** (2013), 779–793

Open dynamic processes with control u and observation y

can mathematically be modeled by abstract input-output systems

$$\begin{aligned}\frac{d}{dt}Mx + Ax &= b - Bu \\ y &= B^*x\end{aligned}$$

in the dissipative case,

Open dynamic processes with control u and observation y

can mathematically be modeled by abstract input-output systems

$$\begin{aligned}\frac{d}{dt}Mx + Ax &= b - Bu \\ y &= B^*x\end{aligned}$$

in the dissipative case, similar in the conservative case even for additional resistive parts given by positive semi-definite $R = R^*$ in dissipative port-Hamiltonian descriptor systems

$$\begin{aligned}M \frac{dx}{dt} &= (J - R) dH(x) - Bu \\ y &= B^* dH(x),\end{aligned}$$

both allowing a Dirac formulation.

Generative AI

predict next state x_{new} and output y from actual state x_{old} and input u

$$x_{new} := F(x_{old}, u)$$

$$y := B^* x_{new},$$

where weights w of $F = F_w$ are learned from training data $((x_i, u_i), y_i)$

$$\frac{1}{N} \sum_{i=1}^N \Phi(B^* F_w(x_i, u_i) - y_i) = \min_w!$$

e.g. for $\Phi(y) := \|y\|_{U^*}^2$ expected quadratic loss is minimized.

Generative AI

predict next state x_{new} and output y from actual state x_{old} and input u

$$x_{new} := F(x_{old}, u)$$

$$y := B^* x_{new},$$

where weights w of $F = F_w$ are learned from training data $((x_i, u_i), y_i)$

$$\frac{1}{N} \sum_{i=1}^N \Phi(B^* F_w(x_i, u_i) - y_i) = \min_w!$$

e.g. for $\Phi(y) := \|y\|_{U^*}^2$ expected quadratic loss is minimized.

Example

In reservoir computing only weights W_{out} of output layer $F(x, u) := W_{out}^T \Phi(x, u)$, $\Phi(x, u) = \sigma(W_{state}x + W_{in}u)$, are fitted to data, while W_{state} , resp. W_{in} , are fixed (and often randomly chosen).

Dirac structures persist

Why emphasize possibility of a Dirac formulation?

Because Dirac formulations of models persist under coordinate changes, restriction to submanifolds (i.e. by imposing constraints) or projection onto quotient manifolds (i.e. reduction e.g. by symmetry).

Dirac structures persist

Why emphasize possibility of a Dirac formulation?

Because Dirac formulations of models persist under coordinate changes, restriction to submanifolds (i.e. by imposing constraints) or projection onto quotient manifolds (i.e. reduction e.g. by symmetry).

Lemma

Pull-back $F^*D := \{(x, F^*y^*) \in X \times X^* \mid (Fx, y^*) \in D\}$ of $D \subset Y \oplus Y^*$ and push-forward $F_*D := \{(Fx, y^*) \in Y \times Y^* \mid (x, F^*y^*) \in D\}$ of $D \subset X \oplus X^*$ by a continuous linear map $F : X \rightarrow Y$ (and compositions) are again constant Dirac structures.

Dirac structures persist

Why emphasize possibility of a Dirac formulation?

Because Dirac formulations of models persist under coordinate changes, restriction to submanifolds (i.e. by imposing constraints) or projection onto quotient manifolds (i.e. reduction e.g. by symmetry).

Lemma

Pull-back $F^*D := \{(x, F^*y^*) \in X \times X^* \mid (Fx, y^*) \in D\}$ of $D \subset Y \oplus Y^*$ and push-forward $F_*D := \{(Fx, y^*) \in Y \times Y^* \mid (x, F^*y^*) \in D\}$ of $D \subset X \oplus X^*$ by a continuous linear map $F : X \rightarrow Y$ (and compositions) are again constant Dirac structures.

- Generalizes to (integrable smooth) Dirac structures in Courant algebroids (e.g. $TX \oplus TX^*$) over manifolds X and bundle maps.

^aCorollary 3.7 of C. Blohmann: Removable Presymplectic Singularities and the Local Splitting of Dirac Structures. IMRN **23** (2017), 7344–7374.

Dirac structures persist

Why emphasize possibility of a Dirac formulation?

Because Dirac formulations of models persist under coordinate changes, restriction to submanifolds (i.e. by imposing constraints) or projection onto quotient manifolds (i.e. reduction e.g. by symmetry).

Lemma

Pull-back $F^*D := \{(x, F^*y^*) \in X \times X^* \mid (Fx, y^*) \in D\}$ of $D \subset Y \oplus Y^*$ and push-forward $F_*D := \{(Fx, y^*) \in Y \times Y^* \mid (x, F^*y^*) \in D\}$ of $D \subset X \oplus X^*$ by a continuous linear map $F : X \rightarrow Y$ (and compositions) are again constant Dirac structures.

- Generalizes to (integrable smooth) Dirac structures in Courant algebroids (e.g. $TX \oplus TX^*$) over manifolds X and bundle maps.
- Local normal forms of Dirac formulations exist^a.

^aCorollary 3.7 of C. Blohmann: Removable Presymplectic Singularities and the Local Splitting of Dirac Structures. IMRN **23** (2017), 7344–7374.

Smart models should be ...

hierarchical:

If state spaces X are infinite-dimensional or operators are complicated, approximate (FEM-)solutions $x_V \in V$ to (Galerkin) projections

$$A_V x_V = b_V$$

onto finite-dimensional $V \subset X$ can be numerically calculated

Smart models should be ...

hierarchical:

If state spaces X are infinite-dimensional or operators are complicated, approximate (FEM-)solutions $x_V \in V$ to (Galerkin) projections

$$A_V x_V = b_V$$

onto finite-dimensional $V \subset X$ can be numerically calculated resp. for

$$\frac{d}{dt} M_V x_V + A_V x_V = b_V, \quad x(t_0) = x_0 \in V,$$

approximate solutions $x_V(t) \in V \subset X \cap Y$.

Smart models should be ...

hierarchical:

If state spaces X are infinite-dimensional or operators are complicated, approximate (FEM-)solutions $x_V \in V$ to (Galerkin) projections

$$A_V x_V = b_V$$

onto finite-dimensional $V \subset X$ can be numerically calculated resp. for

$$\frac{d}{dt} M_V x_V + A_V x_V = b_V, \quad x(t_0) = x_0 \in V,$$

approximate solutions $x_V(t) \in V \subset X \cap Y$. For a nested sequence

$$V_1 \subset V_2 \subset \dots \subset V_k \subset \dots$$

with embeddings $I_k^{k+1} : V_k \rightarrow V_{k+1}$ (prolongation, $(I_k^{k+1})^*$ restriction) satisfying $(I_k^{k+1})^* A_{V_{k+1}} I_k^{k+1} = A_{V_k}$ we have a whole hierarchy of discretizations $A_{V_k} : V_k \rightarrow (V_k)^*$ on different resolution levels.

Smart models should be ...

adaptive:

- For specific use case (i.e. the equation) and a required accuracy the discretization V_k is chosen so that x_{V_k} just has this accuracy
↪ usually implemented in software (e.g. Matlab, FEniCS)

Smart models should be ...

adaptive:

- For specific use case (i.e. the equation) and a required accuracy the discretization V_k is chosen so that x_{V_k} just has this accuracy
 $\curvearrowright$ usually implemented in software (e.g. Matlab, FEniCS)
- Possibly the objective only is partial knowledge of x (e.g. on $\partial\Omega$) and accuracy is demanded only for this part
 $\nrightarrow$ usually not implemented in software.

Smart models should be ...

adaptive:

- For specific use case (i.e. the equation) and a required accuracy the discretization V_k is chosen so that x_{V_k} just has this accuracy
 $\curvearrowright$ usually implemented in software (e.g. Matlab, FEniCS)
- Possibly the objective only is partial knowledge of x (e.g. on $\partial\Omega$) and accuracy is demanded only for this part
 $\nrightarrow$ usually not implemented in software.
- Possibly parameters are uncertain, i.e. data quality is low, and objective is to know whether demanded accuracy can be reached
 $\nrightarrow$ usually not implemented in software.

Smart models should be ...

able to compute sensitivities, i.e. infinitesimal changes of output

- w.r.t. input, e.g. for $Ax = b$ and r.h.s. as input $\frac{\partial x}{\partial b} = \left(\frac{\partial A}{\partial x}(x)\right)^{-1}$,

Smart models should be ...

able to compute sensitivities, i.e. infinitesimal changes of output

- w.r.t. input, e.g. for $Ax = b$ and r.h.s. as input $\frac{\partial x}{\partial b} = \left(\frac{\partial A}{\partial x}(x)\right)^{-1}$,
- w.r.t. parameters, e.g. for $A_k x = b$ the sensitivity $\frac{\partial x}{\partial k}$ solves $\frac{\partial A}{\partial x}(k, x)Z = -\frac{\partial A}{\partial k}(k, x)$,

Smart models should be ...

able to compute sensitivities, i.e. infinitesimal changes of output

- w.r.t. input, e.g. for $Ax = b$ and r.h.s. as input $\frac{\partial x}{\partial b} = (\frac{\partial A}{\partial x}(x))^{-1}$,
- w.r.t. parameters, e.g. for $A_k x = b$ the sensitivity $\frac{\partial x}{\partial k}$ solves $\frac{\partial A}{\partial x}(k, x)Z = -\frac{\partial A}{\partial k}(k, x)$,
- w.r.t. initial value, e.g. for $\frac{d}{dt}Mx + Ax = b$, $x(t_0) = x_0$, the sensitivity $\frac{\partial x}{\partial x_0}(t)$ solves $\frac{d}{dt}\frac{\partial M}{\partial x}(x(t))Y + \frac{\partial A}{\partial x}(x(t))Y = 0$, $Y(t_0) = \text{Id}$,

Smart models should be ...

able to compute sensitivities, i.e. infinitesimal changes of output

- w.r.t. input, e.g. for $Ax = b$ and r.h.s. as input $\frac{\partial x}{\partial b} = (\frac{\partial A}{\partial x}(x))^{-1}$,
 - w.r.t. parameters, e.g. for $A_k x = b$ the sensitivity $\frac{\partial x}{\partial k}$ solves $\frac{\partial A}{\partial x}(k, x)Z = -\frac{\partial A}{\partial k}(k, x)$,
 - w.r.t. initial value, e.g. for $\frac{d}{dt}Mx + Ax = b$, $x(t_0) = x_0$, the sensitivity $\frac{\partial x}{\partial x_0}(t)$ solves $\frac{d}{dt}\frac{\partial M}{\partial x}(x(t))Y + \frac{\partial A}{\partial x}(x(t))Y = 0$, $Y(t_0) = \text{Id}$,
- ↪ partially implemented in software (JAX-FEM)

Smart models should be ...

able to compute sensitivities, i.e. infinitesimal changes of output

- w.r.t. input, e.g. for $Ax = b$ and r.h.s. as input $\frac{\partial x}{\partial b} = \left(\frac{\partial A}{\partial x}(x)\right)^{-1}$,
 - w.r.t. parameters, e.g. for $A_k x = b$ the sensitivity $\frac{\partial x}{\partial k}$ solves $\frac{\partial A}{\partial x}(k, x)Z = -\frac{\partial A}{\partial k}(k, x)$,
 - w.r.t. initial value, e.g. for $\frac{d}{dt}Mx + Ax = b$, $x(t_0) = x_0$, the sensitivity $\frac{\partial x}{\partial x_0}(t)$ solves $\frac{d}{dt}\frac{\partial M}{\partial x}(x(t))Y + \frac{\partial A}{\partial x}(x(t))Y = 0$, $Y(t_0) = \text{Id}$,
- ↪ partially implemented in software (JAX-FEM)

able to compute uncertainties

i.e. (parameters like $\sigma, \dots$ of) the distribution of the output for finite changes of input, parameters, initial values, ... according to some distributions ↯ usually not implemented in software, instead e.g. ensemble simulations

Sensitivities of DLM / AI w.r.t. training data

are defined as

$\frac{\partial}{\partial y} F_{(u,y)}(u)|_{y=F(u)}$, where $F_{(u,y)}$ denotes the DLM obtained by adding to training data for DLM F one artificial data point $(u, y)^a$.

^aor $F(x, u, y)$ for generative AI, J.M., G.Schuldt (2021)

Sensitivities of DLM / AI w.r.t. training data

are defined as

$\frac{\partial}{\partial y} F_{(u,y)}(u)|_{y=F(u)}$, where $F_{(u,y)}$ denotes the DLM obtained by adding to training data for DLM F one artificial data point $(u, y)^a$.

^aor $F(x, u, y)$ for generative AI, J.M., G.Schuldt (2021)

Example

In case of linear (and ridge) regression for $F(u) = W_{out}\Phi(u)$, with $A^T = (\Phi(u_1), \dots, \Phi(u_N))$ the sensitivity $\frac{\partial}{\partial y} F_{(u,y)}(u)|_{y=F(u)}$ is a multiple of Id by the factor^a

$$\frac{\Phi(u)^T (A^T A)^{-1} \Phi(u)}{1 + \Phi(u)^T (A^T A)^{-1} \Phi(u)} \in [0, 1].$$

^aJ.M., W.Schimmel (2023), similarly for generative AI, J.M., M. Schwinger (2025)

Claim

The higher this sensitivity factor at a state x and input u , the less training data is available near (x, u) and the less reliable is the prediction $F(x, u)$ of a generative AI.

Claim

The higher this sensitivity factor at a state x and input u , the less training data is available near (x, u) and the less reliable is the prediction $F(x, u)$ of a generative AI.

Example

This claim has been numerically validated for Hénon-Heiles system

$$\dot{x} = p_x, \dot{y} = p_y, \dot{p}_x = -x - 2xy, \dot{p}_y = -y - x^2 + y^2,$$

with Hamiltonian $H(x, y, p_x, p_y) := \frac{1}{2}(x^2 + y^2 + p_x^2 + p_y^2) + x^2y - \frac{1}{3}y^3$, where energy $E \in [0, \frac{1}{6}]$ is considered as input, for two different sets of training data with 20 000 data points badly resp. well covering the energy surface $H = E$.

Hénon-Heiles: Training data

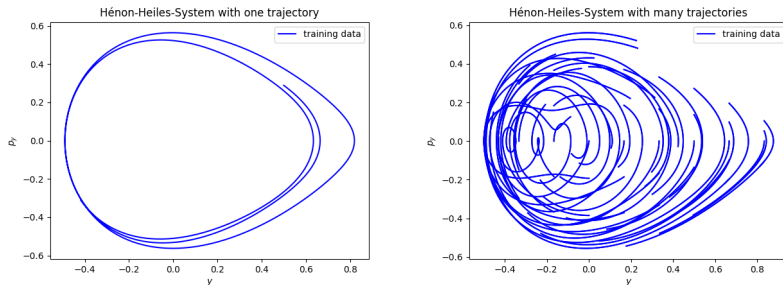
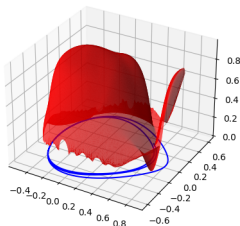


Figure 1: Left set of training data is a single bounded trajectory (starting at $(0, \frac{1}{2}, p_x, \frac{1}{\sqrt{12}})$ with p_x determined by fixed energy E), the right set contains 200 trajectories (of length 100) better covering the energy surface, visualization in the y - p_y -plane.

Hénon-Heiles: Sensitivities over training data

Sensitivity on Hénon-Heiles-System with one trajectory



Sensitivity on Hénon-Heiles-System with many trajectories

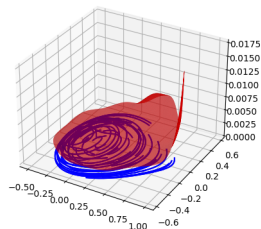


Figure 2: Sensitivities over y - p_y -plane (in red) for the former two sets of training data.

Quasi-static thermo-elasto-plastic deformation

of incompressible Newtonian solids¹ exhibiting Prandtl-Reuss perfect plasticity with elasticity domain $S \subset \{\sigma \in \text{Sym} \mid \text{tr}(\sigma) = 0\}$ bounded, $0 \in S^\circ$, coupled to heat-transfer via thermal expansion E is governed by

$$-\text{div}(\sigma) = -\nabla p + F, \quad \sigma = C\epsilon^{el}, \quad \epsilon^{el} + \epsilon^{pl} + \theta E = \nabla^{sym} u$$

$$\text{div}(u) = 0$$

$$\frac{\partial \epsilon^{pl}}{\partial t} \in \partial \delta_S(\text{dev}(\sigma)) = N_S(\text{dev}(\sigma)) \Leftrightarrow \partial \delta_S^* \left(\frac{\partial \epsilon^{pl}}{\partial t} \right) \ni \text{dev}(\sigma)$$

$$\rho_0 c(\theta) \frac{\partial \theta}{\partial t} - \text{div}(k(\theta) \nabla \theta) = \delta_S^* \left(\frac{\partial \epsilon^{pl}}{\partial t} \right)$$

s.t. initial conditions $u(0) = u_0$, $\epsilon^{pl}(0) = \epsilon_0^{pl}$, $\theta(0) = \theta_0$, boundary conditions $\sigma \cdot \nu = 0$ on Γ_N , $u = u_D$ on Γ_D and $k(\theta) \frac{\partial \theta}{\partial \nu} = f$ on $\partial \Omega$, and has been explored with orthotropic C for embossing of paper².

¹for compressible viscous materials, see T. Roubíček, Thermodynamics of perfect plasticity (2011)

²ESF Young Researcher Group SmartKMU (2019-22) led by S. Schönfelder FING@HTWK

Visco-elastic incompressible flow with free boundary

modeled by Navier-Stokes³ coupled to Oldroyd's non-Newtonian fluid model⁴, where $\frac{\nabla \sigma}{\partial t} := \frac{\partial \sigma}{\partial t} + \operatorname{div}(\sigma \otimes v) + [\sigma, \omega]$ with commutator $[\sigma, \omega]$ of elastic stress σ and vorticity $\omega := \frac{1}{2}(\nabla v - (\nabla v)^T)$, is governed by

$$\rho_0 \left(\frac{\partial v}{\partial t} - \operatorname{div}(v \oplus v) \right) = -\nabla p + \operatorname{div}(\mu_0 \nabla^{\operatorname{sym}} v) + \operatorname{div}(\sigma)$$

$$\frac{\nabla \sigma}{\partial t} + a\sigma = b \nabla^{\operatorname{sym}} v, \quad \operatorname{div}(v) = 0$$

s.t. boundary conditions $v = 0$ on $\Gamma_D(t)$,

$$(-p + \mu_0 \nabla^{\operatorname{sym}} v + \sigma)\nu = \sigma_s \kappa \nu \text{ on } \Gamma_{\operatorname{free}}(t),$$

and initial conditions $v(0) = v_0$ with $\operatorname{div}(v_0) = 0$, $\sigma(0) = \sigma_0$, where κ is mean curvature of $\Gamma_{\operatorname{free}}$, $\sigma_s \geq 0$ surface tension, μ_0 minimal viscosity and $a, b \geq 0$ are related to relaxation time and elasticity.

³for free boundary non-Newtonian, but not visco-elastic two-phase flow, see H. Abels (2007), for pressure reconstruction Abels, Daube, Kraus (2009), for weak-strong uniqueness Fischer, Hensel (2019)

⁴for visco-elastic non-Newtonian, but no free boundary, see P.L. Lions, N. Masmoudi (2000)

On well-posedness of incompressible Navier-Stokes

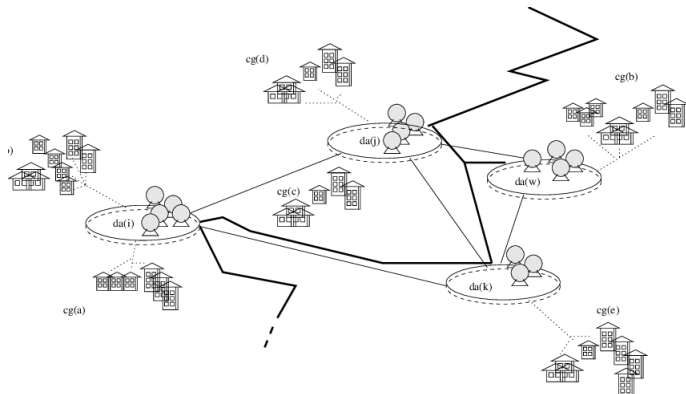
Uniqueness of (very) weak solutions to incompressible Navier-Stokes has been disproved⁵ for initial $v_0 \in L^p_{\text{div}}(\Omega)$, $p < 3$, and recently Open AI has published a solution of the corresponding Millenium problem. In contrast, an appropriate non-linear stress-strain relation allows to prove uniqueness, and for the linear case and strong initial values there is a computable time up to which uniqueness can be guaranteed.

Theorem (J.M.(2023))

For every initial value $v_0 \in H^1_{\text{div}}(\Omega)$ and every $3 < p \leq 4$, there is a time $T > 0$ such that $v \in L^q(0, T; L^p(\Omega, \mathbb{R}^3))$ for the exponent $q := \frac{2p}{p-3}$ determined by equality in the Prodi-Serrin condition, and moreover $v \in L^\infty(0, T; H^1_{\text{div}}(\Omega))$ and $\Delta v \in L^2(0, T; L^2_{\text{div}}(\Omega))$.

⁵see Buckmaster, Vicol (2019), for non-Newtonian flow with p -Laplacian, $p < \frac{6}{5}$, see Burczak, Modena, Székelyhidi (2020), for forced Navier-Stokes equations with $v_0 = 0$ see Albritton, Brué, Colombo (2022).

Are gas network models smart?



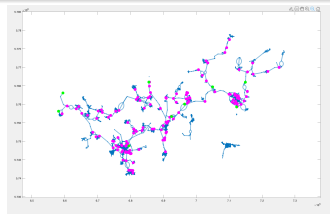
Compressible Euler Equation - 1D

$$\begin{aligned}\frac{\partial \rho}{\partial t} + \frac{\partial \rho u}{\partial x} &= 0 \\ \frac{\partial(\rho u)}{\partial t} + \frac{\partial(\rho u^2 + p(\rho))}{\partial x} + g \rho h_x &= -f \rho u |u| \\ \frac{\partial(\rho(\frac{1}{2}u^2 + e))}{\partial t} + \frac{\partial(\rho u(\frac{1}{2}u^2 + e) + pu)}{\partial x} &= -k(T - T_w)\end{aligned}$$

Quantities of interest are here $\rho, u, e : \mathbb{R} \times [0, L] \rightarrow \mathbb{R}$.

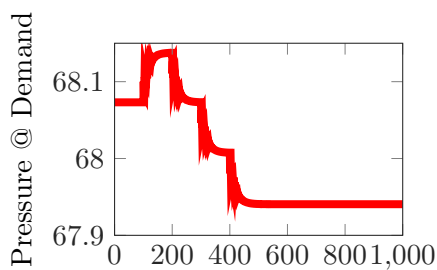
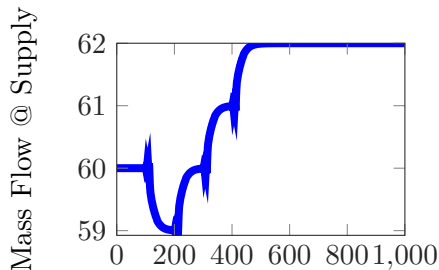
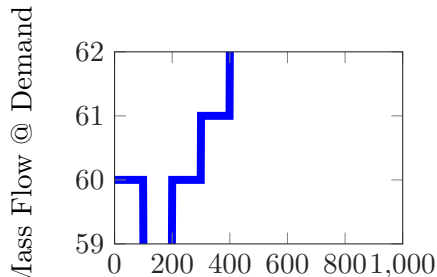
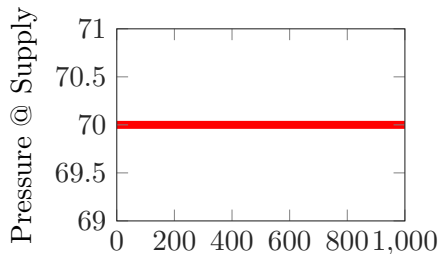
Mathematical Model

- Equations
- Numerical Methods



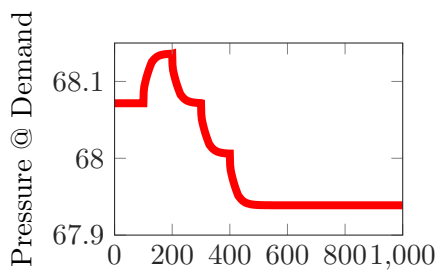
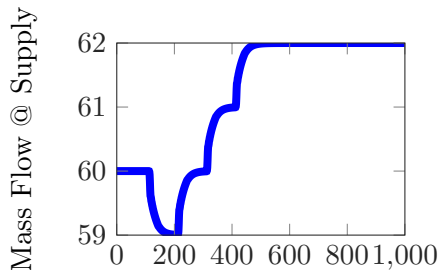
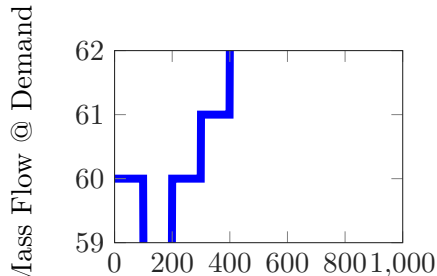
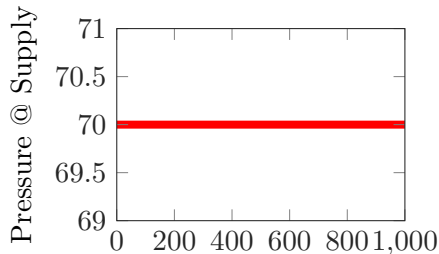
Example - Discretization

Dynamic Simulation Midpoint $\Delta h = 300$



Example - Discretization

Dynamic Simulation Left/Right $\Delta h = 300$



Input Output System

Given an input function u , a differential equation determines the state variable x , which is used to compute the output y .

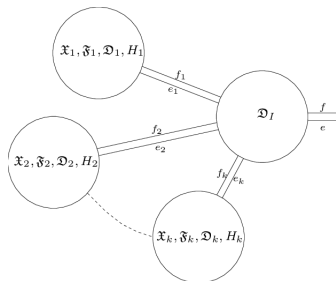
$$\frac{d}{dt}Mx + Ax = b - Bu$$
$$y = C^*x$$

The operators depend

- on the chosen model, including the choice of parameters,
 - on the discretization,
 - on the quantities of interest.
-
- $M = 0$ the stationary problem is included in this framework.
 - Port-Hamiltonian Systems can be formulated this way together with additional conditions.

Why Port-Hamiltonian Systems?

1. **Connection** is formalized with the Dirac structure $\mathcal{D}$



2. **Energy** Port Hamiltonian System have a Hamiltonian H that represents the energy.

Figure 3: Graphic by J. Cervera

Collection of Models

Each has a Hamiltonian, Port Variables and a Dirac Structure.

What about reduced order models?

Given a model of the type

$$\dot{x} = Ax + Bu$$

$$y = C^T x$$

with state $x : \mathbb{R} \rightarrow \mathbb{R}^n$. This defines an input-output map from u to y .
A reduced order model is a model of the type

$$\dot{x}_r = A_r x_r + B_r u$$

$$y_r = C_r^T x_r$$

with $x_r(t) \in \mathbb{R}^r$, with $r < n$. It is again an input-output map from u to y_r .

What about reduced order models?

Given a model of the type

$$\begin{aligned}\dot{x} &= Ax + Bu \\ y &= C^T x\end{aligned}$$

with state $x : \mathbb{R} \rightarrow \mathbb{R}^n$. This defines an input-output map from u to y . A reduced order model is a model of the type

$$\begin{aligned}\dot{x}_r &= A_r x_r + B_r u \\ y_r &= C_r^T x_r\end{aligned}$$

with $x_r(t) \in \mathbb{R}^r$, with $r < n$. It is again an input-output map from u to y_r .

Optimal reduced system

Theoretically, given a measure of similarity in the space of input-output maps and a size of the space r , an optimal reduced system exists.

Linear reduced order models

In practice we can compute projection matrices V and W such that

$$A_r = W^T A V, \quad B_r = W^T B, \quad C_r = V^T C$$

by given algorithms such as

- Iterative Rational Krylov Algorithm
- Balanced Truncation
- Proper Orthogonal Decomposition

Nonlinear Models

Things become as always more complicated.

What about Differential Algebraic Equations (DAE) - M singular

$$M\dot{x} = Ax + Bu \Rightarrow W^T M V \dot{x}_r = W^T A V x_r + W^T B u$$

The Gas Network Model DAE

N. Banagaaya et al found a way to keep the DAE index when projecting a linear DAE.

The type of linear equation we look at is given by

$$M\dot{x} = Ax + f(x) + Bu$$

N. Banagaaya, G. Alì, S. Grundel, P. Benner: **Index-aware model-order reduction for a special class of nonlinear differential-algebraic equations**; Journal of Dynamics and Differential Equations, 34(3) 2022

The Gas Network Model DAE

N. Banagaaya et al found a way to keep the DAE index when projecting a linear DAE.

The type of linear equation we look at is given by

$$M\dot{x} = Ax + f(Mx) + Bu$$

N. Banagaaya, G. Alì, S. Grundel, P. Benner: **Index-aware model-order reduction for a special class of nonlinear differential-algebraic equations**; Journal of Dynamics and Differential Equations, 34(3) 2022

The Gas Network Model DAE

N. Banagaaya et al found a way to keep the DAE index when projecting a linear DAE.

The type of linear equation we look at is given by

$$M\dot{x} = Ax + f(x) + Bu$$

- ❶ In this setting the index-aware concept is doable as well
 - ❷ In earlier work we did a decoupling that is just one specific case of this general framework.
-

N. Banagaaya, G. Alì, S. Grundel, P. Benner: **Index-aware model-order reduction for a special class of nonlinear differential-algebraic equations**; Journal of Dynamics and Differential Equations, 34(3) 2022

Example Result

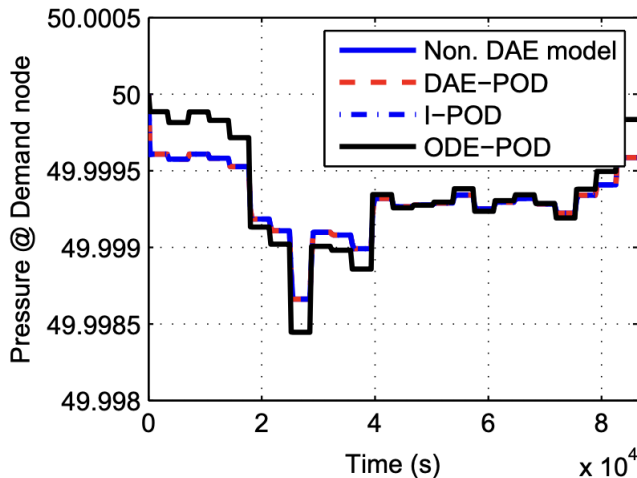


Figure 4: Comparison of the pressure at demand nodes

Choices to make

Model: Equation to describe the network and the components

Space: Solutions Space/Discretization

Solver: Time Integration

Reductor: Reduced Order Model

morgen – **M**odel **O**rders **R**eduction for **G**as and **E**nergy **N**etworks

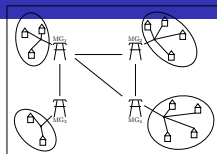
<https://git.io/morgen>

C. Himpe, S. Grundel, P. Benner: **Model Order Reduction for Gas and Energy Networks**; Journal of Mathematics in Industry 11: 13, 2021.

[doi:10.1186/s13362-021-00109-4](https://doi.org/10.1186/s13362-021-00109-4)

Example - Coupled Microgrids

We consider four microgrids (MGs), connected via transportation lines.

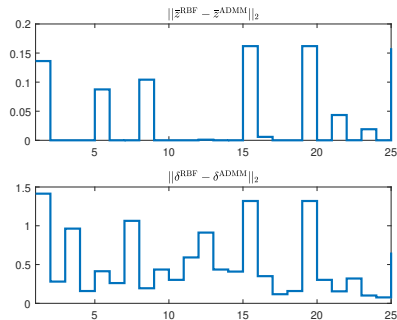
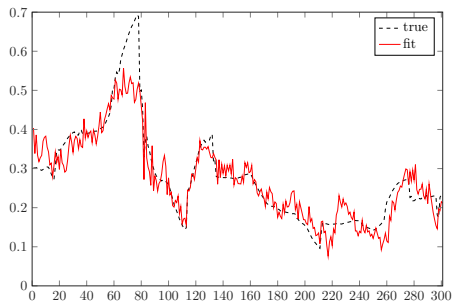


- Each Microgrid has several household, each have a power demand and we can control batteries to optimize the power demand
- Global optimization happens over these 4 microgrids.
- Local Optimization is done via ADMM for example (computationally expensive algorithm)
- Global optimization needs to do the local optimization in each iteration

S. Grundel, P. Sauerteig, K. Worthmann, **Surrogate models for coupled microgrids** Progress in Industrial Mathematics at ECMI 2018,

Approximation with RBF

Comparison of the output of the first component for a random selection of 300 points which are not used to create the surrogate model (left) and for the 25 randomly selected samples both for the output and w.r.t. its impact on the energy exchange (right)



Our numerical case study showed promising results that the exchange coefficients are well approximated—even if individual elements may differ.

Bidirectional Optimization of coupled microgrids

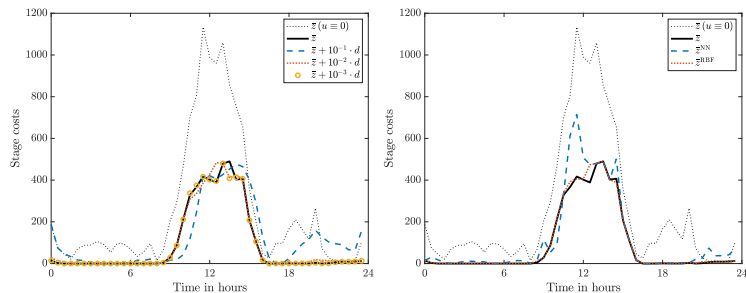


Figure 5: Impact of mapping error (top) and approximation via RBF and NN (bottom) on the costs within 48 consecutive time steps.

M. Baumann, S. Grundel, P. Sauerteig, and K. Worthmann. **Surrogate models in bidirectional optimization of coupled microgrids.** at-Automatisierungstechnik, 67(12), 1035-1046, 2019.

What Energy Systems and Virus Outbreaks have in common?

Optimal control on Dynamical System

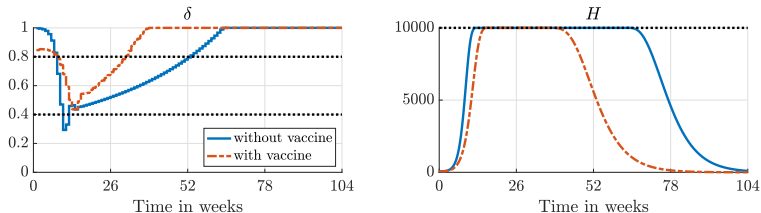


Figure 6: Optimal social distancing policy δ (left) and ICU occupancy $H = \sum_{i=1}^{n_g} H_i + H_i^V$ (right) with and without vaccine. The dotted black lines mark strict social distancing, i.e. $\delta \in [0.4, 0.8]$ (left) and $H^{\max}$ (right).

S. Grundel, S. Heyder, T. Hotz, T.K. Ritschel, P. Sauerteig, K. Worthmann, **How to coordinate vaccination and social distancing to mitigate SARS-CoV-2 outbreaks**. SIAM Journal on Applied Dynamical Systems, 20(2), 1135-1157. (2021)

Conclusion

- A good model needs a precise definition of the problems, such that a right model equation and right numerical techniques can be used to solve the problem.

Conclusion

- A good model needs a precise definition of the problems, such that a right model equation and right numerical techniques can be used to solve the problem.
- A smart model is a collection of good models that anticipates all possible problems and chooses the right one.

Conclusion

- A good model needs a precise definition of the problems, such that a right model equation and right numerical techniques can be used to solve the problem.
- A smart model is a collection of good models that anticipates all possible problems and chooses the right one.
- At MNZ@HTWK-LEIPZIG.DE, we are open to collaboration, please feel free to get in touch with any questions you may have. We will then assess who your issue is most relevant to, and can decide on the next steps.

Conclusion

- A good model needs a precise definition of the problems, such that a right model equation and right numerical techniques can be used to solve the problem.
- A smart model is a collection of good models that anticipates all possible problems and chooses the right one.
- At MNZ@HTWK-LEIPZIG.DE, we are open to collaboration, please feel free to get in touch with any questions you may have. We will then assess who your issue is most relevant to, and can decide on the next steps.

Thanks for your attention!