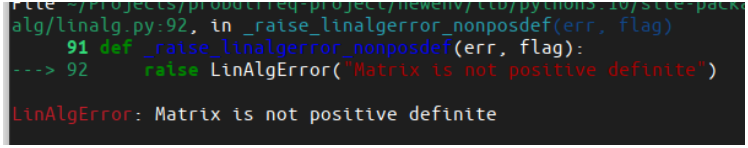
Practical Probabilistic Numerical Solvers

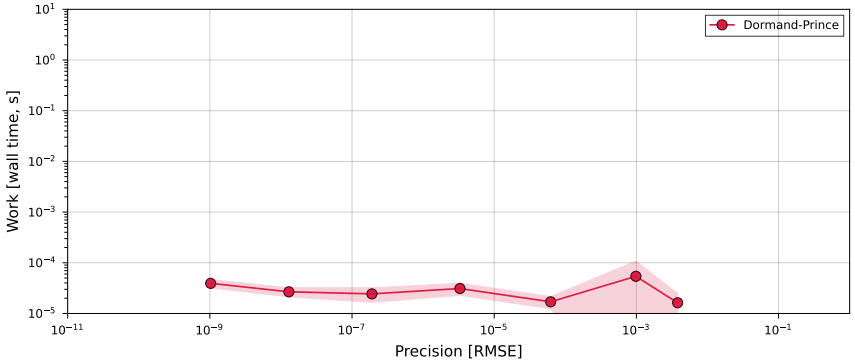
Nico Krämer

Technical University of Denmark

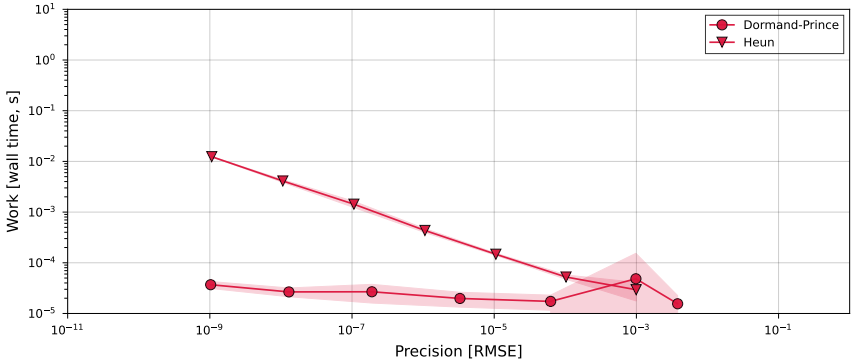
I have bad news.



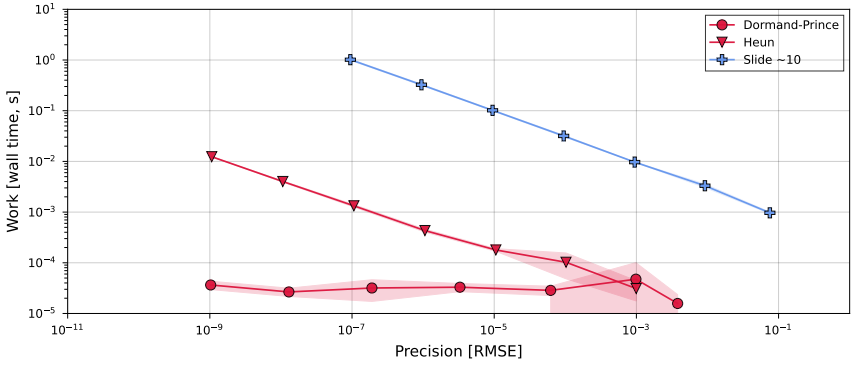
Work vs. precision



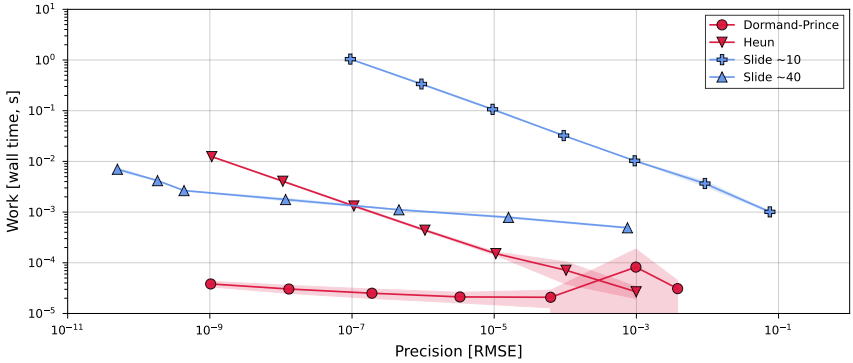
Work vs. precision



Work vs. precision



Work vs. precision



Practical Probabilistic Numerical Solvers (now for real)

Nico Krämer

Technical University of Denmark

Probabilistic solvers in a nutshell

- ◇ Initial value problem (IVP):

$$\frac{\mathrm{d}}{\mathrm{d}t} y(t) = f(y(t), t), \quad y(0) = y_0$$

- ◇ Hypothesize $Y \sim \mathrm{GP}$. Grid $t_0, ..., t_N$. Approximate y by estimating (variations of)

$$p\left(Y(\cdot) \;\middle|\; \left\{ \dot{Y}(t_n) = f(Y(t_n), t_n) \right\}_{n=0}^{N}, \; Y(t_0) = y_0 \right)$$

(WLOG $t_0 = 0$) *Tronarp et al., Cockayne et al.*

- ◇ We call this *probabilistic IVP solution*
- ◇ Same for BVPs, PDEs, mass-matrices, etc. *Krämer & Hennig, Krämer et al., Bosch et al.*
- ◇ Gauss–Markov prior $p(y) \Rightarrow$ probabilistic solution in $O(N)$ *Schober et al.*

Probabilistic solvers in a nutshell

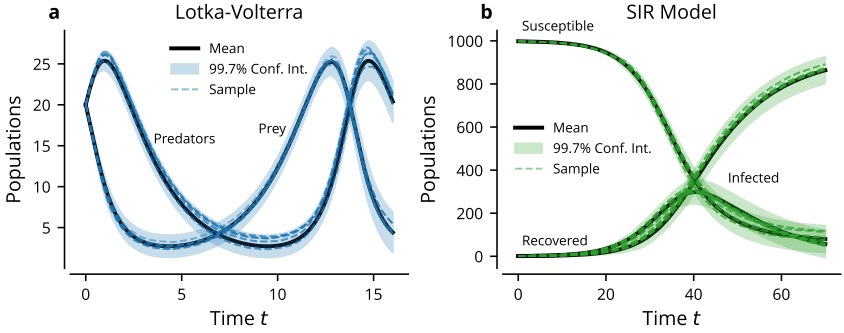


Figure: Krämer & Hennig

$$Y \sim \mathrm{GP}$$

Stochastic differential equation priors

$$\frac{\mathrm{d}^{\nu+1}}{\mathrm{d}t^{\nu+1}} Y(t) - \sum_{k=0}^{\nu} A_k \frac{\mathrm{d}^k}{\mathrm{d}t^k} Y(t) = \sigma w(t) \quad + \text{Gaussian initial condition}$$

Stochastic differential equation priors

$$\frac{\mathrm{d}}{\mathrm{d}t}\begin{pmatrix} Y(t) \\ \frac{\mathrm{d}}{\mathrm{d}t}Y(t) \\ \vdots \\ \frac{\mathrm{d}^\nu}{\mathrm{d}t^\nu}Y(t) \end{pmatrix} = \begin{pmatrix} 0 & I & & \\ & \ddots & \ddots & \\ & & 0 & I \\ -A_0 & \dots & \dots & -A_\nu \end{pmatrix} \begin{pmatrix} Y(t) \\ \frac{\mathrm{d}}{\mathrm{d}t}Y(t) \\ \vdots \\ \frac{\mathrm{d}^\nu}{\mathrm{d}t^\nu}Y(t) \end{pmatrix} + \sigma \begin{pmatrix} 0 \\ \vdots \\ 0 \\ I \end{pmatrix} w(t)$$

Stochastic differential equation priors

$$\frac{\mathrm{d}}{\mathrm{d}t}\mathbf{Y}(t) = \mathbf{A}\mathbf{Y}(t) + \sigma\mathbf{B}w(t)$$

where

$$\mathbf{Y}(t) = \begin{pmatrix} Y(t) \\ \frac{\mathrm{d}}{\mathrm{d}t}Y(t) \\ \vdots \\ \frac{\mathrm{d}^\nu}{\mathrm{d}t^\nu}Y(t) \end{pmatrix}, \quad \mathbf{A} = \begin{pmatrix} 0 & I & & \\ & \ddots & \ddots & \\ & & 0 & I \\ -A_0 & \dots & \dots & -A_\nu \end{pmatrix}, \quad \mathbf{B} = \begin{pmatrix} 0 \\ \vdots \\ 0 \\ I \end{pmatrix}$$

Our favourite example: integrated Wiener processes

- ν-times integrated Wiener process

$$\frac{\mathrm{d}^{\nu+1}}{\mathrm{d}t^{\nu+1}} Y(t) = \sigma w(t) \tag{1}$$

  (i.e. $A_0 = A_1 = ... = 0$)
- Corresponds to spline models

see works by Wahba

- It is just fantastic

Integrated Wiener processes

$A_0 = A_1 = ... = 0$:

$$\frac{\mathrm{d}}{\mathrm{d}t}\mathbf{Y}(t) = \mathbf{A}\mathbf{Y}(t) + \sigma\mathbf{B}w(t)$$

where

$$\mathbf{Y}(t) = \begin{pmatrix} Y(t) \\ \frac{\mathrm{d}}{\mathrm{d}t}Y(t) \\ \vdots \\ \frac{\mathrm{d}^\nu}{\mathrm{d}t^\nu}Y(t) \end{pmatrix}, \quad \mathbf{A} = \begin{pmatrix} 0 & I & & \\ & \ddots & \ddots & \\ & & 0 & I \\ 0 & \dots & 0 & 0 \end{pmatrix}, \quad \mathbf{B} = \begin{pmatrix} 0 \\ \vdots \\ 0 \\ I \end{pmatrix}$$

Discretised integrated Wiener process

$$p(\mathbf{Y}(t+\Delta t) \mid \mathbf{Y}(t), \sigma) = \mathsf{N}((\Phi(\Delta t) \otimes I)\mathbf{Y}(t), \Sigma(\Delta t, \sigma) \otimes I),$$

with

e.g. Schober et al., Kersting et al.

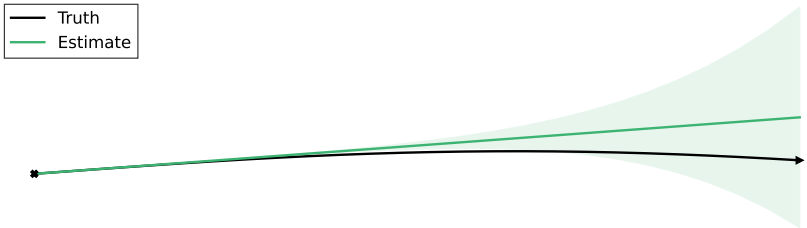
$$\Phi(\Delta t) = 1_{i \leq j} \left(\frac{(\Delta t)^{j-i}}{(j-i)!} \right)_{ij}, \quad \Sigma(\Delta t, \sigma) = \sigma^2 \left(\frac{\Delta t^{2\nu+1-i-j}}{(2\nu+1-i-j)(\nu-i)!(\nu-j)!} \right)_{ij}$$

for example with $\nu = 2$:

$$\Phi(\Delta t) = \begin{pmatrix} 1 & \Delta t & (\Delta t)^2/2 \\ 0 & 1 & \Delta t \\ 0 & 0 & 1 \end{pmatrix}, \quad \Sigma(\Delta t, \sigma) = \sigma^2 \begin{pmatrix} (\Delta t)^5/20 & (\Delta t)^4/8 & (\Delta t)^3/6 \\ (\Delta t)^4/8 & (\Delta t)^3/6 & (\Delta t)^2/2 \\ (\Delta t)^3/6 & (\Delta t)^2/2 & \Delta t \end{pmatrix}$$

Discretised integrated Wiener process

$$\begin{pmatrix} Y(t+\Delta t) \\ \frac{\mathrm{d}}{\mathrm{d}t} Y(t+\Delta t) \\ \frac{\mathrm{d}^2}{\mathrm{d}t^2} Y(t+\Delta t) \end{pmatrix} = \begin{pmatrix} 1 & \Delta t & (\Delta t)^2/2 \\ 0 & 1 & \Delta t \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} Y(t) \\ \frac{\mathrm{d}}{\mathrm{d}t} Y(t) \\ \frac{\mathrm{d}^2}{\mathrm{d}t^2} Y(t) \end{pmatrix} + \mathsf{N}(0, \Sigma(\Delta t, \sigma))$$



Sequential IVP solution

- Probabilistic IVP solution

$$p\left(Y \;\middle|\; \left\{\dot{Y}(t_n) = f(Y(t_n), t_n)\right\}_{n=0}^{N},\; Y(t_0) = y_0\right)$$

- Implement as

$$p\left(Y, \frac{\mathrm{d}}{\mathrm{d}t}Y, ..., \frac{\mathrm{d}^\nu}{\mathrm{d}t^\nu}Y \;\middle|\; \left\{\dot{Y}(t_n) = f(Y(t_n), t_n)\right\}_{n=0}^{N},\; Y(t_0) = y_0, \sigma\right)$$

Sequential IVP solution

- Probabilistic IVP solution

$$p\left(Y \;\middle|\; \left\{\dot{Y}(t_n) = f(Y(t_n), t_n)\right\}_{n=0}^{N},\; Y(t_0) = y_0\right)$$

- Implement as

$$p\left(Y, \frac{\mathrm{d}}{\mathrm{d}t}Y, ..., \frac{\mathrm{d}^\nu}{\mathrm{d}t^\nu}Y \;\middle|\; \left\{\dot{Y}(t_n) = f(Y(t_n), t_n)\right\}_{n=0}^{N},\; Y(t_0) = y_0, \sigma\right)$$

Sequential IVP solution

- Probabilistic IVP solution

$$p\left(Y \;\middle|\; \left\{\dot{Y}(t_n) = f(Y(t_n), t_n)\right\}_{n=0}^{N},\; Y(t_0) = y_0\right)$$

- Implement as

$$p\left(Y, \frac{\mathrm{d}}{\mathrm{d}t}Y, ..., \frac{\mathrm{d}^\nu}{\mathrm{d}t^\nu}Y \;\middle|\; \left\{\dot{Y}(t_n) = f(Y(t_n), t_n)\right\}_{n=0}^{N},\; Y(t_0) = y_0, \sigma\right)$$

Sequential IVP solution

- Probabilistic IVP solution

$$p\left(Y \;\middle|\; \left\{\dot{Y}(t_n) = f(Y(t_n), t_n)\right\}_{n=0}^{N},\; Y(t_0) = y_0\right)$$

- Implement as

$$p\left(\mathbf{Y} \;\middle|\; \left\{\dot{Y}(t_n) = f(Y(t_n), t_n)\right\}_{n=0}^{N},\; Y(t_0) = y_0, \sigma\right)$$

Sequential IVP solution

- Probabilistic IVP solution

$$p\left(Y \;\middle|\; \left\{\dot{Y}(t_n) = f(Y(t_n), t_n)\right\}_{n=0}^{N},\; Y(t_0) = y_0\right)$$

- Implement as

$$p\left(\{\mathbf{Y}(t_n)\}_{n=0}^{N} \;\middle|\; \left\{\dot{Y}(t_n) = f(Y(t_n), t_n)\right\}_{n=0}^{N},\; Y(t_0) = y_0, \sigma\right)$$

Sequential IVP solution

- Probabilistic IVP solution

$$p\left(Y \;\middle|\; \left\{\dot{Y}(t_n) = f(Y(t_n), t_n)\right\}_{n=0}^{N},\; Y(t_0) = y_0\right)$$

- Implement as

$$p\left(\{\mathbf{Y}(t_n)\}_{n=0}^{N} \;\middle|\; \left\{\dot{Y}(t_n) - f(Y(t_n), t_n) = 0\right\}_{n=0}^{N},\; Y(t_0) = y_0, \sigma\right)$$

Sequential IVP solution

- Probabilistic IVP solution

$$p\left(Y \;\middle|\; \left\{\dot{Y}(t_n) = f(Y(t_n), t_n)\right\}_{n=0}^{N},\; Y(t_0) = y_0\right)$$

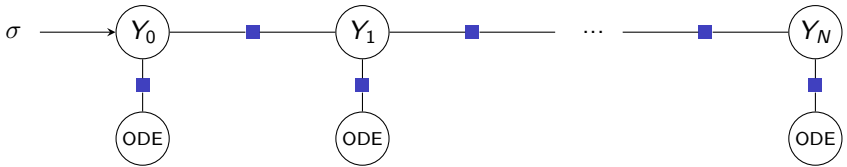
- Implement as

$$p\left(\mathbf{Y}(t_{0:N}) \mid (\mathcal{R}\mathbf{Y})(t_{0:N}) = 0, Y(t_0) = y_0, \sigma\right)$$

Sequential solution

$$
\begin{aligned}
&p(\mathbf{Y}(t_{0:N}) \mid (\mathcal{R}\mathbf{Y})(t_{0:N}) = 0, Y(t_0) = y_0, \sigma) \\
&\qquad = p(\mathbf{Y}(t_N) \mid (\mathcal{R}\mathbf{Y})(t_{0:N}) = 0, Y(t_0) = y_0, \sigma) \\
&\qquad\qquad \times \prod_{n=0}^{N-1} p((\mathbf{Y}(t_n) \mid \mathbf{Y}(t_{n+1}), (\mathcal{R}\mathbf{Y})(t_{0:n}) = 0, Y(t_0) = y_0, \sigma)
\end{aligned}
$$

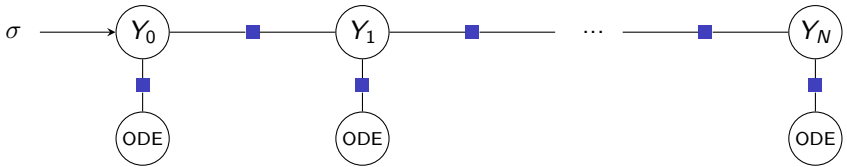
e.g. Tronarp, Bosch, Hennig



Sequential solution

$$
\begin{aligned}
&p(\mathbf{Y}(t_{0:N}) \mid (\mathcal{R}\mathbf{Y})(t_{0:N}) = 0, Y(t_0) = y_0, \sigma) \\
&\qquad = p(\mathbf{Y}(t_N) \mid (\mathcal{R}\mathbf{Y})(t_{0:N}) = 0, Y(t_0) = y_0, \sigma) \\
&\qquad\qquad \times \prod_{n=0}^{N-1} p((\mathbf{Y}(t_n) \mid \mathbf{Y}(t_{n+1}), (\mathcal{R}\mathbf{Y})(t_{0:n}) = 0, Y(t_0) = y_0, \sigma)
\end{aligned}
$$

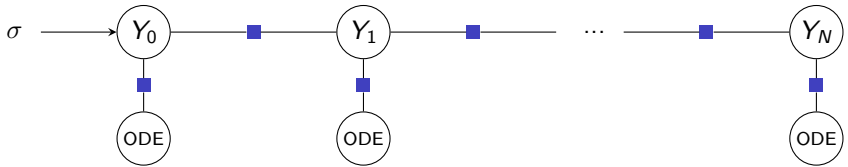
e.g. Tronarp, Bosch, Hennig



Sequential solution

$$
\begin{aligned}
p(\text{IVP solution}) &= p(\text{Terminal-value solution}) \times \prod_{n=0}^{N-1} p(\text{backward-transitions}) \\
&= [\text{Simulate forward-in-time}] \times [\text{Byproducts of forward simulation}] \\
&= [\text{Always interesting}] \times [\text{Optional}]
\end{aligned}
$$

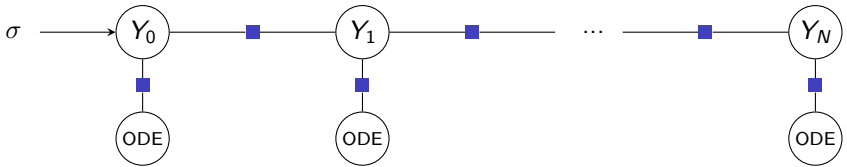
e.g. Tronarp, Bosch, Hennig



Sequential solution

$$
\begin{aligned}
&p(\mathbf{Y}(t_{0:N}) \mid (\mathcal{R}\mathbf{Y})(t_{0:N}) = 0, Y(t_0) = y_0, \sigma) \\
&\qquad = p(\mathbf{Y}(t_N) \mid (\mathcal{R}\mathbf{Y})(t_{0:N}) = 0, Y(t_0) = y_0, \sigma) \\
&\qquad\qquad \times \prod_{n=0}^{N-1} p((\mathbf{Y}(t_n) \mid \mathbf{Y}(t_{n+1}), (\mathcal{R}\mathbf{Y})(t_{0:n}) = 0, Y(t_0) = y_0, \sigma)
\end{aligned}
$$

e.g. Tronarp, Bosch, Hennig

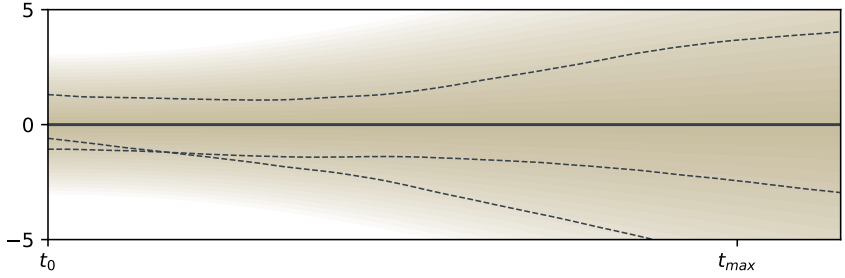


Algorithm: Sequential IVP solver

1. Initialise boundary conditions: $p(\mathbf{Y}(t_0) \mid Y(t_0) = y_0, \sigma)$
2. Initialise first grid point: $p(\mathbf{Y}(t_0) \mid Y(t_0) = y_0, (\mathcal{R}\mathbf{Y})(t_0) = 0, \sigma)$
3. For all $(t_n, t_{n+1} = t_n + \Delta t_n]$:
   - 3.1 Extrapolate: Compute
     - $p(\mathbf{Y}(t_{n+1}) \mid \mathcal{R}\mathbf{Y}(t_{0:n}) = 0, Y(t_0) = y_0, \sigma)$
     - $p(\mathbf{Y}(t_n) \mid \mathbf{Y}(t_{n+1}), \mathcal{R}\mathbf{Y}(t_{0:n}) = 0, Y(t_0) = y_0, \sigma)$
   - 3.2 Linearise: $\mathcal{R}\mathbf{Y}(t_{n+1}) \approx H\mathbf{Y}(t_{n+1}) + b$
   - 3.3 Correct: $\approx p(\mathbf{Y}(t_{n+1}) \mid \mathcal{R}\mathbf{Y}(t_{0:n+1}) = 0, Y(t_0) = y_0, \sigma)$
4. Do something fun with the terminal-value solution and (optionally) the backward transitions

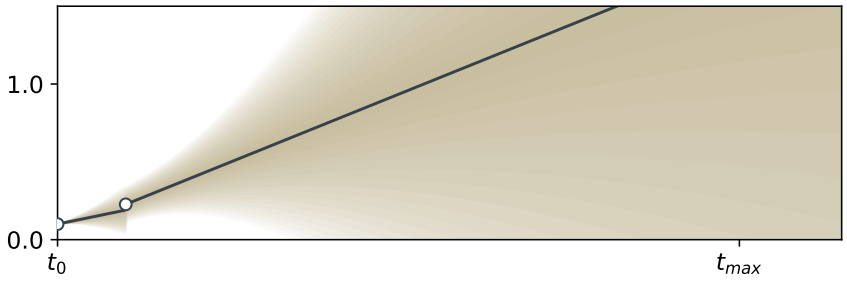
Like Kalman filtering, sum-product algorithm, sequential least-squares, etc.

Algorithm: Sequential IVP solver



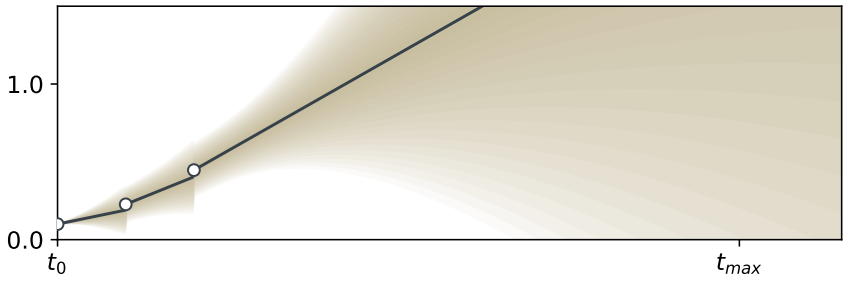
https://www.youtube.com/watch?v=vlMibwkanQw

Algorithm: Sequential IVP solver



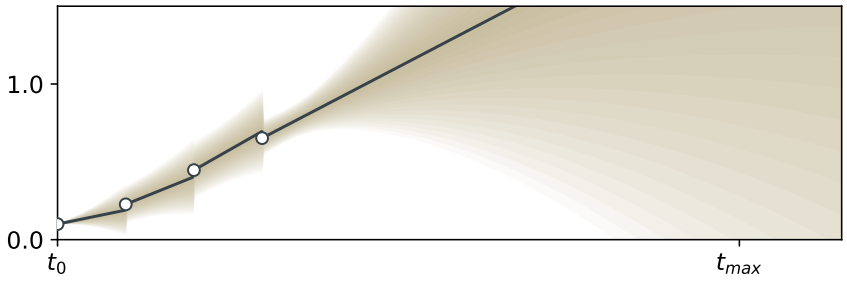
https://www.youtube.com/watch?v=vlMibwkanQw

Algorithm: Sequential IVP solver



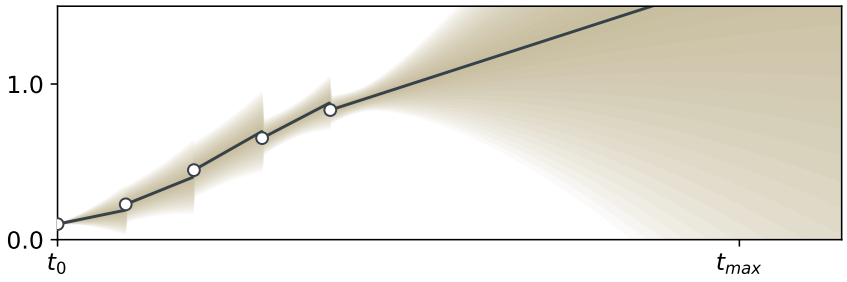
https://www.youtube.com/watch?v=vlMibwkanQw

Algorithm: Sequential IVP solver



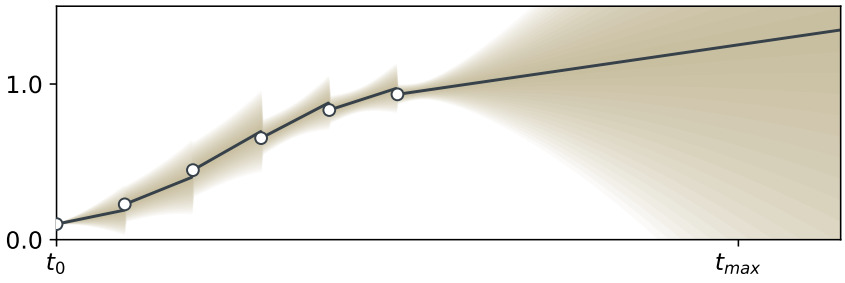
https://www.youtube.com/watch?v=vlMibwkanQw

Algorithm: Sequential IVP solver



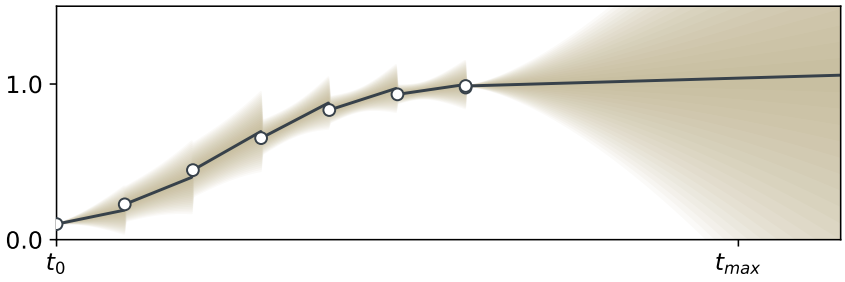
https://www.youtube.com/watch?v=vlMibwkanQw

Algorithm: Sequential IVP solver



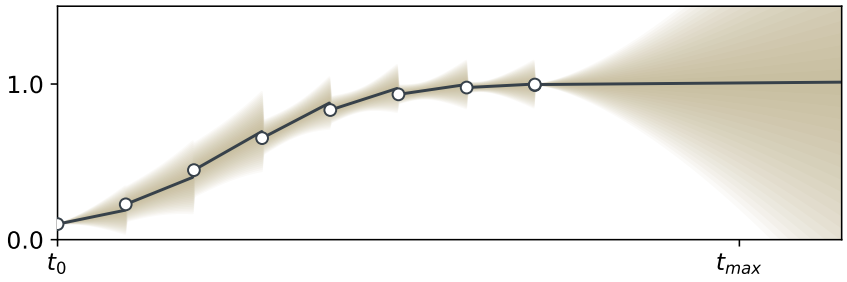
https://www.youtube.com/watch?v=vlMibwkanQw

Algorithm: Sequential IVP solver



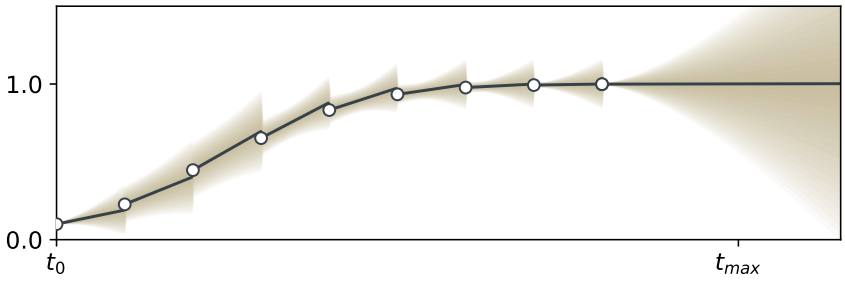
https://www.youtube.com/watch?v=vlMibwkanQw

Algorithm: Sequential IVP solver



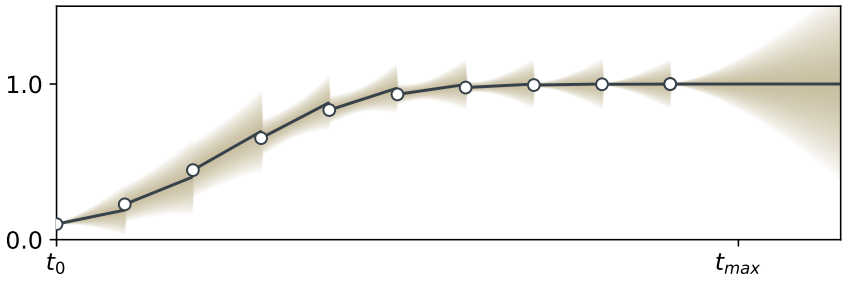
https://www.youtube.com/watch?v=vlMibwkanQw

Algorithm: Sequential IVP solver



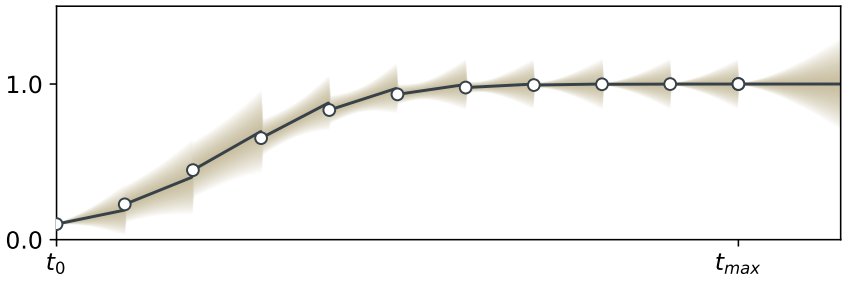
https://www.youtube.com/watch?v=vlMibwkanQw

Algorithm: Sequential IVP solver



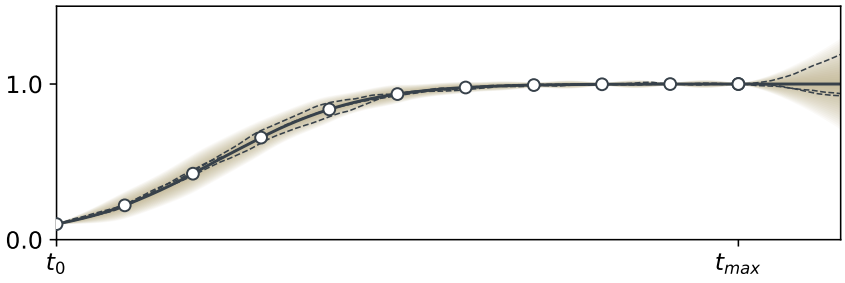
https://www.youtube.com/watch?v=vlMibwkanQw

Algorithm: Sequential IVP solver



https://www.youtube.com/watch?v=vlMibwkanQw

Algorithm: Sequential IVP solver



https://www.youtube.com/watch?v=vlMibwkanQw

Algorithm: Sequential IVP solver

1. Initialise boundary conditions: $p(\mathbf{Y}(t_0) \mid Y(t_0) = y_0, \sigma)$
2. Initialise first grid point: $p(\mathbf{Y}(t_0) \mid Y(t_0) = y_0, (\mathcal{R}\mathbf{Y})(t_0) = 0, \sigma)$
3. For all $(t_n, t_{n+1} = t_n + \Delta t_n]$:
   - 3.1 Extrapolate: Compute
     - $p(\mathbf{Y}(t_{n+1}) \mid \mathcal{R}\mathbf{Y}(t_{0:n}) = 0, Y(t_0) = y_0, \sigma)$
     - $p(\mathbf{Y}(t_n) \mid \mathbf{Y}(t_{n+1}), \mathcal{R}\mathbf{Y}(t_{0:n}) = 0, Y(t_0) = y_0, \sigma)$
   - 3.2 Linearise: $\mathcal{R}\mathbf{Y}(t_{n+1}) \approx H\mathbf{Y}(t_{n+1}) + b$
   - 3.3 Correct: $\approx p(\mathbf{Y}(t_{n+1}) \mid \mathcal{R}\mathbf{Y}(t_{0:n+1}) = 0, Y(t_0) = y_0, \sigma)$
4. Do something fun with the terminal-value solution and (optionally) the backward transitions

Where's the Kalman filter?

1. Initialise boundary conditions: $p(\mathbf{Y}(t_0) \mid Y(t_0) = y_0, \sigma)$
2. Initialise first grid point: $p(\mathbf{Y}(t_0) \mid Y(t_0) = y_0, (\mathcal{R}\mathbf{Y})(t_0) = 0, \sigma)$
3. For all $(t_n, t_{n+1} = t_n + \Delta t_n]$:
   - 3.1 Extrapolate: Compute
     - $p(\mathbf{Y}(t_{n+1}) \mid \mathcal{R}\mathbf{Y}(t_{0:n}) = 0, Y(t_0) = y_0, \sigma)$
     - $p(\mathbf{Y}(t_n) \mid \mathbf{Y}(t_{n+1}), \mathcal{R}\mathbf{Y}(t_{0:n}) = 0, Y(t_0) = y_0, \sigma)$
   - 3.2 Linearise: $\mathcal{R}\mathbf{Y}(t_{n+1}) \approx H\mathbf{Y}(t_{n+1}) + b$
   - 3.3 Correct: $\approx p(\mathbf{Y}(t_{n+1}) \mid \mathcal{R}\mathbf{Y}(t_{0:n+1}) = 0, Y(t_0) = y_0, \sigma)$
4. Do something fun with the terminal-value solution
   and (optionally) the backward transitions

Where's the Rauch–Tung–Striebel smoother?

1. Initialise boundary conditions: $p(\mathbf{Y}(t_0) \mid Y(t_0) = y_0, \sigma)$
2. Initialise first grid point: $p(\mathbf{Y}(t_0) \mid Y(t_0) = y_0, (\mathcal{R}\mathbf{Y})(t_0) = 0, \sigma)$
3. For all $(t_n, t_{n+1} = t_n + \Delta t_n]$:
   - 3.1 Extrapolate: Compute
     - $p(\mathbf{Y}(t_{n+1}) \mid \mathcal{R}\mathbf{Y}(t_{0:n}) = 0, Y(t_0) = y_0, \sigma)$
     - $p(\mathbf{Y}(t_n) \mid \mathbf{Y}(t_{n+1}), \mathcal{R}\mathbf{Y}(t_{0:n}) = 0, Y(t_0) = y_0, \sigma)$
   - 3.2 Linearise: $\mathcal{R}\mathbf{Y}(t_{n+1}) \approx H\mathbf{Y}(t_{n+1}) + b$
   - 3.3 Correct: $\approx p(\mathbf{Y}(t_{n+1}) \mid \mathcal{R}\mathbf{Y}(t_{0:n+1}) = 0, Y(t_0) = y_0, \sigma)$
4. Do something fun with the terminal-value solution and (optionally) the backward transitions

Where's the extended/unscented filter/smoother?

1. Initialise boundary conditions: $p(\mathbf{Y}(t_0) \mid Y(t_0) = y_0, \sigma)$
2. Initialise first grid point: $p(\mathbf{Y}(t_0) \mid Y(t_0) = y_0, (\mathcal{R}\mathbf{Y})(t_0) = 0, \sigma)$
3. For all $(t_n, t_{n+1} = t_n + \Delta t_n]$:
   - 3.1 Extrapolate: Compute
     - $p(\mathbf{Y}(t_{n+1}) \mid \mathcal{R}\mathbf{Y}(t_{0:n}) = 0, Y(t_0) = y_0, \sigma)$
     - $p(\mathbf{Y}(t_n) \mid \mathbf{Y}(t_{n+1}), \mathcal{R}\mathbf{Y}(t_{0:n}) = 0, Y(t_0) = y_0, \sigma)$
   - 3.2 Linearise: $\mathcal{R}\mathbf{Y}(t_{n+1}) \approx H\mathbf{Y}(t_{n+1}) + b$
   - 3.3 Correct: $\approx p(\mathbf{Y}(t_{n+1}) \mid \mathcal{R}\mathbf{Y}(t_{0:n+1}) = 0, Y(t_0) = y_0, \sigma)$
4. Do something fun with the terminal-value solution and (optionally) the backward transitions

Where's Gauss–Newton?

1. Initialise boundary conditions: $p(\mathbf{Y}(t_0) \mid Y(t_0) = y_0, \sigma)$
2. Initialise first grid point: $p(\mathbf{Y}(t_0) \mid Y(t_0) = y_0, (\mathcal{R}\mathbf{Y})(t_0) = 0, \sigma)$
3. For all $(t_n, t_{n+1} = t_n + \Delta t_n]$:
   3.1 Extrapolate: Compute
   - $p(\mathbf{Y}(t_{n+1}) \mid \mathcal{R}\mathbf{Y}(t_{0:n}) = 0, Y(t_0) = y_0, \sigma)$
   - $p(\mathbf{Y}(t_n) \mid \mathbf{Y}(t_{n+1}), \mathcal{R}\mathbf{Y}(t_{0:n}) = 0, Y(t_0) = y_0, \sigma)$

   3.2 Repeat until convergence:
   - Linearise: $\mathcal{R}\mathbf{Y}(t_{n+1}) \approx H\mathbf{Y}(t_{n+1}) + b$
   - Correct: $\approx p(\mathbf{Y}(t_{n+1}) \mid \mathcal{R}\mathbf{Y}(t_{0:n+1}) = 0, Y(t_0) = y_0, \sigma)$
4. Do something fun with the terminal-value solution and (optionally) the backward transitions

see works by Bell

Where's parameter estimation?

$\frac{\mathrm{d}}{\mathrm{d}t} y(t) = f_\theta(y(t), t)$

1. Initialise boundary conditions: $p(\mathbf{Y}(t_0) \mid Y(t_0) = y_0, \sigma, \theta)$
2. Initialise first grid point: $p(\mathbf{Y}(t_0) \mid Y(t_0) = y_0, (\mathcal{R}\mathbf{Y})(t_0) = 0, \sigma, \theta)$
3. For all $(t_n, t_{n+1} = t_n + \Delta t_n]$:
   - 3.1 Extrapolate: Compute
     - $\diamond$ $p(\mathbf{Y}(t_{n+1}) \mid \mathcal{R}\mathbf{Y}(t_{0:n}) = 0, Y(t_0) = y_0, \sigma, \theta)$
     - $\diamond$ $p(\mathbf{Y}(t_n) \mid \mathbf{Y}(t_{n+1}), \mathcal{R}\mathbf{Y}(t_{0:n}) = 0, Y(t_0) = y_0, \sigma, \theta)$
   - 3.2 Linearise: $\mathcal{R}\mathbf{Y}(t_{n+1}) \approx H\mathbf{Y}(t_{n+1}) + b$
   - 3.3 Correct: $\approx p(\mathbf{Y}(t_{n+1}) \mid \mathcal{R}\mathbf{Y}(t_{0:n+1}) = 0, Y(t_0) = y_0, \sigma, \theta)$
4. Do something fun with the terminal-value solution and (optionally) the backward transitions

e.g. Kersting, Krämer, et al.; Tronarp, Bosch, Hennig.

Can we solve second-order problems?

$\frac{\mathrm{d}^2}{\mathrm{d}t^2} y(t) = f(y(t), t)$

1. Initialise boundary conditions: $p(\mathbf{Y}(t_0) \mid Y(t_0) = y_0, \sigma)$
2. Initialise first grid point: $p(\mathbf{Y}(t_0) \mid Y(t_0) = y_0, (\mathcal{R}\mathbf{Y})(t_0) = 0, \sigma)$
3. For all $(t_n, t_{n+1} = t_n + \Delta t_n]$:
   - 3.1 Extrapolate: Compute
     - $p(\mathbf{Y}(t_{n+1}) \mid \mathcal{R}\mathbf{Y}(t_{0:n}) = 0, Y(t_0) = y_0, \sigma)$
     - $p(\mathbf{Y}(t_n) \mid \mathbf{Y}(t_{n+1}), \mathcal{R}\mathbf{Y}(t_{0:n}) = 0, Y(t_0) = y_0, \sigma)$
   - 3.2 Linearise: $\mathcal{R}\mathbf{Y}(t_{n+1}) \approx H\mathbf{Y}(t_{n+1}) + b$
   - 3.3 Correct: $\approx p(\mathbf{Y}(t_{n+1}) \mid \mathcal{R}\mathbf{Y}(t_{0:n+1}) = 0, Y(t_0) = y_0, \sigma)$
4. Do something fun with the terminal-value solution and (optionally) the backward transitions

Bosch et al.

Can we solve 8th-order problems?

$\frac{\mathrm{d}^8}{\mathrm{d}t^8} y(t) = f(y(t), t)$

1. Initialise boundary conditions: $p(\mathbf{Y}(t_0) \mid Y(t_0) = y_0, \sigma)$
2. Initialise first grid point: $p(\mathbf{Y}(t_0) \mid Y(t_0) = y_0, (\mathcal{R}\mathbf{Y})(t_0) = 0, \sigma)$
3. For all $(t_n, t_{n+1} = t_n + \Delta t_n]$:
   - 3.1 Extrapolate: Compute
     - $p(\mathbf{Y}(t_{n+1}) \mid \mathcal{R}\mathbf{Y}(t_{0:n}) = 0, Y(t_0) = y_0, \sigma)$
     - $p(\mathbf{Y}(t_n) \mid \mathbf{Y}(t_{n+1}), \mathcal{R}\mathbf{Y}(t_{0:n}) = 0, Y(t_0) = y_0, \sigma)$
   - 3.2 Linearise: $\mathcal{R}\mathbf{Y}(t_{n+1}) \approx H\mathbf{Y}(t_{n+1}) + b$
   - 3.3 Correct: $\approx p(\mathbf{Y}(t_{n+1}) \mid \mathcal{R}\mathbf{Y}(t_{0:n+1}) = 0, Y(t_0) = y_0, \sigma)$
4. Do something fun with the terminal-value solution and (optionally) the backward transitions

Bosch et al.

Can we solve mass-matrix problems?

$\mathsf{M}\frac{\mathrm{d}}{\mathrm{d}t}y(t) = f(y(t), t)$

1. Initialise boundary conditions: $p(\mathbf{Y}(t_0) \mid Y(t_0) = y_0, \sigma)$
2. Initialise first grid point: $p(\mathbf{Y}(t_0) \mid Y(t_0) = y_0, (\mathcal{R}\mathbf{Y})(t_0) = 0, \sigma)$
3. For all $(t_n, t_{n+1} = t_n + \Delta t_n]$:
   - 3.1 Extrapolate: Compute
     - $p(\mathbf{Y}(t_{n+1}) \mid \mathcal{R}\mathbf{Y}(t_{0:n}) = 0, Y(t_0) = y_0, \sigma)$
     - $p(\mathbf{Y}(t_n) \mid \mathbf{Y}(t_{n+1}), \mathcal{R}\mathbf{Y}(t_{0:n}) = 0, Y(t_0) = y_0, \sigma)$
   - 3.2 Linearise: $\mathcal{R}\mathbf{Y}(t_{n+1}) \approx H\mathbf{Y}(t_{n+1}) + b$
   - 3.3 Correct: $\approx p(\mathbf{Y}(t_{n+1}) \mid \mathcal{R}\mathbf{Y}(t_{0:n+1}) = 0, Y(t_0) = y_0, \sigma)$
4. Do something fun with the terminal-value solution and (optionally) the backward transitions

Bosch et al.

Where's the boundary value problem solver?

$\frac{\mathrm{d}^2}{\mathrm{d}t^2} y(t) = f(y(t), t),\ y(0) = y_0,\ y(1) = y_1$

1. Initialise boundary conditions: $p(\mathbf{Y}(t_0) \mid Y(t_0) = y_0, Y(t_N) = y_1, \sigma)$
2. Linearise: $\mathcal{R}\mathbf{Y}(t_{0:N}) \approx R_0 + \hat{\mathcal{R}}\mathbf{Y}(t_{0:N})$
3. Initialise first grid point: $p(\mathbf{Y}(t_0) \mid Y(t_0) = y_0, (\mathcal{R}\mathbf{Y})(t_0) = 0, Y(t_N) = y_1, \sigma)$
4. For all $(t_n, t_{n+1} = t_n + \Delta t_n]$:
   - 4.1 Extrapolate: Compute
     - $p(\mathbf{Y}(t_{n+1}) \mid \mathcal{R}\mathbf{Y}(t_{0:n}) = 0, Y(t_0) = y_0, Y(t_N) = y_1, \sigma)$
     - $p(\mathbf{Y}(t_n) \mid \mathbf{Y}(t_{n+1}), \mathcal{R}\mathbf{Y}(t_{0:n}) = 0, Y(t_0) = y_0, Y(t_N) = y_1, \sigma)$
   - 4.2 Correct: $\approx p(\mathbf{Y}(t_{n+1}) \mid \mathcal{R}\mathbf{Y}(t_{0:n+1}) = 0, Y(t_0) = y_0, Y(t_N) = y_1, \sigma)$
5. Rinse and repeat from 2.
6. Do something fun with the terminal-value solution and (optionally) the backward transitions

Krämer & Hennig

Wait!

Algorithm: Sequential IVP solver

1. Initialise boundary conditions: $p(\mathbf{Y}(t_0) \mid Y(t_0) = y_0, \sigma)$
2. Initialise first grid point: $p(\mathbf{Y}(t_0) \mid Y(t_0) = y_0, (\mathcal{R}\mathbf{Y})(t_0) = 0, \sigma)$
3. For all $(t_n, t_{n+1} = t_n + \Delta t_n]$:
   - 3.1 Extrapolate: Compute
     - $p(\mathbf{Y}(t_{n+1}) \mid \mathcal{R}\mathbf{Y}(t_{0:n}) = 0, Y(t_0) = y_0, \sigma)$
     - $p(\mathbf{Y}(t_n) \mid \mathbf{Y}(t_{n+1}), \mathcal{R}\mathbf{Y}(t_{0:n}) = 0, Y(t_0) = y_0, \sigma)$
   - 3.2 Linearise: $\mathcal{R}\mathbf{Y}(t_{n+1}) \approx H\mathbf{Y}(t_{n+1}) + b$
   - 3.3 Correct: $\approx p(\mathbf{Y}(t_{n+1}) \mid \mathcal{R}\mathbf{Y}(t_{0:n+1}) = 0, Y(t_0) = y_0, \sigma)$
4. Do something fun with the terminal-value solution and (optionally) the backward transitions

Algorithm: Sequential IVP solver

1. Initialise boundary conditions: $p(\mathbf{Y}(t_0) \mid Y(t_0) = y_0, \sigma)$
2. Initialise first grid point: $p(\mathbf{Y}(t_0) \mid Y(t_0) = y_0, (\mathcal{R}\mathbf{Y})(t_0) = 0, \sigma)$
3. For all $(t_n, t_{n+1} = t_n + \Delta t_n ???]$:
   - 3.1 Extrapolate: Compute
     - $p(\mathbf{Y}(t_{n+1}) \mid \mathcal{R}\mathbf{Y}(t_{0:n}) = 0, Y(t_0) = y_0, \sigma)$
     - $p(\mathbf{Y}(t_n) \mid \mathbf{Y}(t_{n+1}), \mathcal{R}\mathbf{Y}(t_{0:n}) = 0, Y(t_0) = y_0, \sigma)$
   - 3.2 Linearise: $\mathcal{R}\mathbf{Y}(t_{n+1}) \approx H\mathbf{Y}(t_{n+1}) + b$
   - 3.3 Correct: $\approx p(\mathbf{Y}(t_{n+1}) \mid \mathcal{R}\mathbf{Y}(t_{0:n+1}) = 0, Y(t_0) = y_0, \sigma)$
4. Do something fun with the terminal-value solution and (optionally) the backward transitions

Δt_n ???

Step-size adaptation

- ◇ Choose $t_0, ..., t_N$ adaptively.
- ◇ Ensure that all "local errors" are equal to tolerance.
- ◇ If error at $t_n + \Delta t_n$ is too large:
  1. Estimate the error introduced by the previous step *Schober et al., Bosch et al.*
  2. Assume the local error decays as $(\Delta t)^\rho$
  3. Then, choosing ("integral"-controller) *e.g. Hairer & Wanner*

$$t_{n+1}^{\text{new}} = t_n + \Delta t_n^{\text{new}}, \quad \Delta t_n^{\text{new}} \sim \left(\frac{\text{tol}}{\text{error}}\right)^{1/\rho} \Delta t_{n-1}$$

  will (usually) yield a sufficiently small step
- ◇ Use a more sophisticated version ("proportional-integral" controller) *See works by Gustafsson*

Algorithm: Sequential IVP solver

1. Initialise boundary conditions: $p(\mathbf{Y}(t_0) \mid Y(t_0) = y_0, \sigma)$
2. Initialise first grid point: $p(\mathbf{Y}(t_0) \mid Y(t_0) = y_0, (\mathcal{R}\mathbf{Y})(t_0) = 0, \sigma)$
3. While $t_n + \Delta t_n \leq 1$ (= terminal time):
   - 3.1 Extrapolate: Compute
     - $p(\mathbf{Y}(t_{n+1}) \mid \mathcal{R}\mathbf{Y}(t_{0:n}) = 0, Y(t_0) = y_0, \sigma)$
     - $p(\mathbf{Y}(t_n) \mid \mathbf{Y}(t_{n+1}), \mathcal{R}\mathbf{Y}(t_{0:n}) = 0, Y(t_0) = y_0, \sigma)$
   - 3.2 Linearise: $\mathcal{R}\mathbf{Y}(t_{n+1}) \approx H\mathbf{Y}(t_{n+1}) + b$
   - 3.3 Correct: $\approx p(\mathbf{Y}(t_{n+1}) \mid \mathcal{R}\mathbf{Y}(t_{0:n+1}) = 0, Y(t_0) = y_0, \sigma)$
   - 3.4 Estimate error
   - 3.5 Accept or reject step
   - 3.6 Propose "optimal" Δt_{n+1} and repeat
4. Do something fun with the terminal-value solution and (optionally) the backward transitions

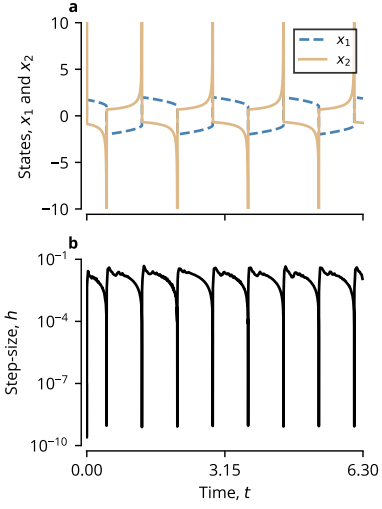
Adaptive steps are mandatory

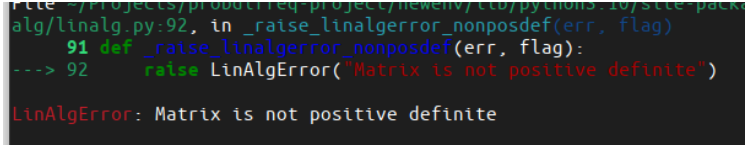
Stiff van-der-Pol system,

$$\frac{\mathrm{d}^2}{\mathrm{d}t^2}y(t) = 10^6(1 - y(t)^2)\frac{\mathrm{d}}{\mathrm{d}t}y(t) - y(t)$$

- ◇ Crazy "spikes"
- ◇ Requires non-uniform steps

see Bosch et al.; Figure: Krämer & Hennig





Algorithm: Sequential IVP solver

1. Initialise boundary conditions: $p(\mathbf{Y}(t_0) \mid Y(t_0) = y_0, \sigma)$
2. Initialise first grid point: $p(\mathbf{Y}(t_0) \mid Y(t_0) = y_0, (\mathcal{R}\mathbf{Y})(t_0) = 0, \sigma)$
3. While $t_n + \Delta t_n \leq 1$ (= terminal time):
   - 3.1 Extrapolate: Compute
     - $p(\mathbf{Y}(t_{n+1}) \mid \mathcal{R}\mathbf{Y}(t_{0:n}) = 0, Y(t_0) = y_0, \sigma)$
     - $p(\mathbf{Y}(t_n) \mid \mathbf{Y}(t_{n+1}), \mathcal{R}\mathbf{Y}(t_{0:n}) = 0, Y(t_0) = y_0, \sigma)$
   - 3.2 Linearise: $\mathcal{R}\mathbf{Y}(t_{n+1}) \approx H\mathbf{Y}(t_{n+1}) + b$
   - 3.3 Correct: $\approx p(\mathbf{Y}(t_{n+1}) \mid \mathcal{R}\mathbf{Y}(t_{0:n+1}) = 0, Y(t_0) = y_0, \sigma)$
   - 3.4 Estimate error
   - 3.5 Accept or reject step
   - 3.6 Propose "optimal" Δt_{n+1} and repeat
4. Do something fun with the terminal-value solution and (optionally) the backward transitions

Algorithm: Sequential IVP solver

1. Initialise boundary conditions: $p(\mathbf{Y}(t_0) \mid Y(t_0) = y_0, \sigma)$
2. Initialise first grid point: $p(\mathbf{Y}(t_0) \mid Y(t_0) = y_0, (\mathcal{R}\mathbf{Y})(t_0) = 0, \sigma)$
3. While $t_n + \Delta t_n \leq 1$ (= terminal time):
   - 3.1 Extrapolate: Compute
     - $p(\mathbf{Y}(t_{n+1}) \mid \mathcal{R}\mathbf{Y}(t_{0:n}) = 0, Y(t_0) = y_0, \sigma)$
     - $p(\mathbf{Y}(t_n) \mid \mathbf{Y}(t_{n+1}), \mathcal{R}\mathbf{Y}(t_{0:n}) = 0, Y(t_0) = y_0, \sigma)$
   - 3.2 Linearise: $\mathcal{R}\mathbf{Y}(t_{n+1}) \approx H\mathbf{Y}(t_{n+1}) + b$
   - 3.3 Correct: $\approx p(\mathbf{Y}(t_{n+1}) \mid \mathcal{R}\mathbf{Y}(t_{0:n+1}) = 0, Y(t_0) = y_0, \sigma)$
   - 3.4 Estimate error
   - 3.5 Accept or reject step
   - 3.6 Propose "optimal" Δt_{n+1} and repeat
4. Do something fun with the terminal-value solution and (optionally) the backward transitions

Algorithm: Sequential IVP solver

1. Initialise boundary conditions: $p(\mathbf{Y}(t_0) \mid Y(t_0) = y_0, \sigma)$
2. Initialise first grid point: $p(\mathbf{Y}(t_0) \mid Y(t_0) = y_0, (\mathcal{R}\mathbf{Y})(t_0) = 0, \sigma)$
3. While $t_n + \Delta t_n \leq 1$ (= terminal time):
   - 3.1 Extrapolate: Compute
     - $p(\mathbf{Y}(t_{n+1}) \mid \mathcal{R}\mathbf{Y}(t_{0:n}) = 0, Y(t_0) = y_0, \sigma)$
     - $p(\mathbf{Y}(t_n) \mid \mathbf{Y}(t_{n+1}), \mathcal{R}\mathbf{Y}(t_{0:n}) = 0, Y(t_0) = y_0, \sigma)$
   - 3.2 Linearise: $\mathcal{R}\mathbf{Y}(t_{n+1}) \approx H\mathbf{Y}(t_{n+1}) + b$
   - 3.3 Correct: $\approx p(\mathbf{Y}(t_{n+1}) \mid \mathcal{R}\mathbf{Y}(t_{0:n+1}) = 0, Y(t_0) = y_0, \sigma)$
   - 3.4 Estimate error
   - 3.5 Accept or reject step
   - 3.6 Propose "optimal" Δt_{n+1} and repeat
4. Do something fun with the terminal-value solution and (optionally) the backward transitions

Algorithm: Sequential IVP solver

1. Initialise boundary conditions: $p(\mathbf{Y}(t_0) \mid Y(t_0) = y_0, \sigma)$
2. Initialise first grid point: $p(\mathbf{Y}(t_0) \mid Y(t_0) = y_0, (\mathcal{R}\mathbf{Y})(t_0) = 0, \sigma)$
3. While $t_n + \Delta t_n \leq 1$ (= terminal time):
   - 3.1 Extrapolate: Compute
     - $p(\mathbf{Y}(t_{n+1}) \mid \mathcal{R}\mathbf{Y}(t_{0:n}) = 0, Y(t_0) = y_0, \sigma)$
     - $p(\mathbf{Y}(t_n) \mid \mathbf{Y}(t_{n+1}), \mathcal{R}\mathbf{Y}(t_{0:n}) = 0, Y(t_0) = y_0, \sigma)$
   - 3.2 Linearise: $\mathcal{R}\mathbf{Y}(t_{n+1}) \approx H\mathbf{Y}(t_{n+1}) + b$
   - 3.3 Correct: $\approx p(\mathbf{Y}(t_{n+1}) \mid \mathcal{R}\mathbf{Y}(t_{0:n+1}) = 0, Y(t_0) = y_0, \sigma)$
   - 3.4 Estimate error
   - 3.5 Accept or reject step
   - 3.6 Propose "optimal" Δt_{n+1} and repeat
4. Do something fun with the terminal-value solution and (optionally) the backward transitions

Algorithm: Sequential IVP solver

1. Initialise boundary conditions: $p(\mathbf{Y}(t_0) \mid Y(t_0) = y_0, \sigma)$
2. Initialise first grid point: $p(\mathbf{Y}(t_0) \mid Y(t_0) = y_0, (\mathcal{R}\mathbf{Y})(t_0) = 0, \sigma)$
3. While $t_n + \Delta t_n \leq 1$ (= terminal time):
   - 3.1 Extrapolate: Compute
     - $p(\mathbf{Y}(t_{n+1}) \mid \mathcal{R}\mathbf{Y}(t_{0:n}) = 0, Y(t_0) = y_0, \sigma)$
     - $p(\mathbf{Y}(t_n) \mid \mathbf{Y}(t_{n+1}), \mathcal{R}\mathbf{Y}(t_{0:n}) = 0, Y(t_0) = y_0, \sigma)$
   - 3.2 Linearise: $\mathcal{R}\mathbf{Y}(t_{n+1}) \approx H\mathbf{Y}(t_{n+1}) + b$
   - 3.3 Correct: $\approx p(\mathbf{Y}(t_{n+1}) \mid \mathcal{R}\mathbf{Y}(t_{0:n+1}) = 0, Y(t_0) = y_0, \sigma)$
   - 3.4 Estimate error
   - 3.5 Accept or reject step
   - 3.6 Propose "optimal" Δt_{n+1} and repeat
4. Do something fun with the terminal-value solution and (optionally) the backward transitions

($\nu = 3$)

$$p(\mathbf{Y}(t_0) \mid \sigma)$$

$$\mathsf{N}\left(\begin{pmatrix}\star\\ \star\\ \star\\ \star\end{pmatrix}, \begin{pmatrix}\star & \star & \star & \star\\ \star & \star & \star & \star\\ \star & \star & \star & \star\\ \star & \star & \star & \star\end{pmatrix}\right)$$

($\nu = 3$)

$$p(\mathbf{Y}(t_0) \mid Y(t_0) = y_0, \sigma)$$

$$\mathsf{N}\left(\begin{pmatrix} y_0 \\ \star \\ \star \\ \star \end{pmatrix}, \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & \star & \star & \star \\ 0 & \star & \star & \star \\ 0 & \star & \star & \star \end{pmatrix}\right)$$

($\nu = 3$)

$$p(\mathbf{Y}(t_0) \mid Y(t_0) = y_0, (\mathcal{R}\mathbf{Y})(t_0) = 0, \sigma)$$

$$\mathsf{N}\left(\begin{pmatrix} y_0 \\ f(y_0, t_0) \\ \star \\ \star \end{pmatrix}, \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & \star & \star \\ 0 & 0 & \star & \star \end{pmatrix}\right)$$

$(\nu = 3)$

$$p(\mathbf{Y}(t_0) \mid Y(t_0) = y_0, (\mathcal{R}\mathbf{Y})(t_0) = 0, \frac{\mathrm{d}}{\mathrm{d}t}(\mathcal{R}\mathbf{Y})(t_0) = 0, \sigma)$$

$$\mathsf{N}\left(\begin{pmatrix} y_0 \\ f(y_0, t_0) \\ J_y f(y_0, t_0) f(y_0, t_0) + J_t f(y_0, t_0) \\ \star \end{pmatrix}, \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \star \end{pmatrix}\right)$$

($\nu = 3$)

$$p(\mathbf{Y}(t_0) \mid Y(t_0) = y_0, (\mathcal{R}\mathbf{Y})(t_0) = 0, \frac{\mathrm{d}}{\mathrm{d}t}(\mathcal{R}\mathbf{Y})(t_0) = 0, \sigma)$$

$$\mathsf{N}\left(\begin{pmatrix} y_0 \\ f(y_0, t_0) \\ J_y f(y_0, t_0) f(y_0, t_0) + J_t f(y_0, t_0) \\ \star \end{pmatrix}, \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \star \end{pmatrix}\right)$$

Taylor approximation of IVP solutions

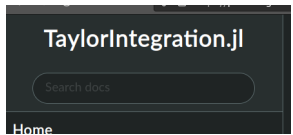
$$
\begin{aligned}
y &= y_0 \\
\frac{\mathrm{d}}{\mathrm{d}t} y &= f \\
\frac{\mathrm{d}^2}{\mathrm{d}t^2} y &= [J_y f](f) + [J_t f](1) \\
\frac{\mathrm{d}^3}{\mathrm{d}t^3} y &= [J_y^2 f](f, f) + [J_y f]([J_y f](f)) + [J_t^2 f](1, 1)
\end{aligned}
$$

$$
[n\text{-th Taylor coefficient of } y] = [(n-1)\text{-th Taylor coefficient of } (f \circ y)]
$$

Taylor arithmetic

$$\mathcal{A} : (f, [n\text{-th order Taylor series of } y]) \longrightarrow [n\text{-th order Taylor series of } f \circ y]$$

$$\left[\frac{\mathrm{d}}{\mathrm{d}t}y(t), ..., \frac{\mathrm{d}^{(n+1)}}{\mathrm{d}t^{(n+1)}}y(t)\right] = \mathcal{A}\left(f, \left[y(t), \frac{\mathrm{d}}{\mathrm{d}t}y(t), ..., \frac{\mathrm{d}^{n}}{\mathrm{d}t^{n}}y(t)\right]\right)$$



e.g. Griewank & Walther

$(\nu = 3)$

$$p(\mathbf{Y}(t_0) \mid Y(t_0) = y_0, (\mathcal{R}\mathbf{Y})(t_0) = 0, \frac{\mathrm{d}}{\mathrm{d}t}(\mathcal{R}\mathbf{Y})(t_0) = 0, \sigma)$$

$$\mathsf{N}\left(\begin{pmatrix} y_0 \\ f(y_0, t_0) \\ J_y f(y_0, t_0) f(y_0, t_0) + J_t f(y_0, t_0) \\ \star \end{pmatrix}, \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \star \end{pmatrix}\right)$$

($\nu = 3$)

$$p(\mathbf{Y}(t_0) \mid Y(t_0) = y_0, (\mathcal{R}\mathbf{Y})(t_0) = 0, \frac{\mathrm{d}}{\mathrm{d}t}(\mathcal{R}\mathbf{Y})(t_0) = 0, \sigma)$$

$$\mathsf{N}\left(\begin{pmatrix} y_0 \\ f(y_0, t_0) \\ J_y f(y_0, t_0) f(y_0, t_0) + J_t f(y_0, t_0) \\ \star \end{pmatrix}, \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \star \end{pmatrix}\right)$$

($\nu = 3$)

$$p(\mathbf{Y}(t_0) \mid Y(t_0) = y_0, (\mathcal{R}\mathbf{Y})(t_0) = 0, \frac{\mathrm{d}}{\mathrm{d}t}(\mathcal{R}\mathbf{Y})(t_0) = 0, \sigma)$$

$$\mathsf{N}\left(\begin{pmatrix} y(t_0) \\ \frac{\mathrm{d}^1}{\mathrm{d}t^1} y(t_0) \\ \frac{\mathrm{d}^2}{\mathrm{d}t^2} y(t_0) \\ \frac{\mathrm{d}^3}{\mathrm{d}t^3} y(t_0) \end{pmatrix}, \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}\right)$$

Extrapolation

$$p(\mathbf{Y}(t+\Delta t) \mid \mathbf{Y}(t), \sigma) = \mathsf{N}((\Phi(\Delta t) \otimes I)\mathbf{Y}(t), \Sigma(\Delta t, \sigma) \otimes I),$$

with

e.g. Schober et al., Kersting et al.

$$\Phi(\Delta t) = \begin{pmatrix} 1 & \Delta t & (\Delta t)^2/2 \\ 0 & 1 & \Delta t \\ 0 & 0 & 1 \end{pmatrix},$$

$$\Sigma(\Delta t, \sigma) = \begin{pmatrix} (\Delta t)^4/20 & (\Delta t)^3/8 & (\Delta t)^2/6 \\ (\Delta t)^3/8 & (\Delta t)^2/6 & \Delta t/2 \\ (\Delta t)^2/6 & \Delta t/2 & 1 \end{pmatrix} \cdot \sigma \cdot \Delta t$$

Extrapolation

$$p(\mathbf{Y}(t+\Delta t) \mid \mathbf{Y}(t), \sigma) = \mathsf{N}((\Phi(\Delta t) \otimes I)\mathbf{Y}(t), \Sigma(\Delta t, \sigma) \otimes I),$$

with

e.g. Schober et al., Kersting et al.

$$\Phi(\Delta t) = \begin{pmatrix} 1 & \Delta t & (\Delta t)^2/2 \\ 0 & 1 & \Delta t \\ 0 & 0 & 1 \end{pmatrix},$$

$$\Sigma(\Delta t, \sigma) = \begin{pmatrix} (\Delta t)^4/20 & (\Delta t)^3/8 & (\Delta t)^2/6 \\ (\Delta t)^3/8 & (\Delta t)^2/6 & \Delta t/2 \\ (\Delta t)^2/6 & \Delta t/2 & 1 \end{pmatrix} \cdot \sigma \cdot \Delta t$$

Extrapolation

$$C(t + \Delta t) \approx 0 + \Sigma(\Delta t, \sigma)$$

with

$$\Phi(\Delta t) = \begin{pmatrix} 1 & \Delta t & (\Delta t)^2/2 \\ 0 & 1 & \Delta t \\ 0 & 0 & 1 \end{pmatrix},$$

$$\Sigma(\Delta t, \sigma) = \begin{pmatrix} (\Delta t)^4/20 & (\Delta t)^3/8 & (\Delta t)^2/6 \\ (\Delta t)^3/8 & (\Delta t)^2/6 & \Delta t/2 \\ (\Delta t)^2/6 & \Delta t/2 & 1 \end{pmatrix} \cdot \sigma \cdot \Delta t$$

Extrapolation

$$C(t + \Delta t) = \Phi(\Delta t) C(t) \Phi(\Delta t)^\top + \Sigma(\Delta t, \sigma)$$

with

$$\Phi(\Delta t) = \begin{pmatrix} 1 & \Delta t & (\Delta t)^2/2 \\ 0 & 1 & \Delta t \\ 0 & 0 & 1 \end{pmatrix},$$

$$\Sigma(\Delta t, \sigma) = \begin{pmatrix} (\Delta t)^4/20 & (\Delta t)^3/8 & (\Delta t)^2/6 \\ (\Delta t)^3/8 & (\Delta t)^2/6 & \Delta t/2 \\ (\Delta t)^2/6 & \Delta t/2 & 1 \end{pmatrix} \cdot \sigma \cdot \Delta t$$

Extrapolation

$$C(t+\Delta t) = T(\Delta t)\left[\Phi_0 T(\Delta t)^{-1} C(t) T(\Delta t)^{-\top} \Phi_0^\top + \Sigma_0\right] T(\Delta t)^\top$$

with

$$T(\Delta t) = \sqrt{\Delta t}\begin{pmatrix} \frac{(\Delta t)^2}{2} & 0 & 0 \\ 0 & \Delta t & 0 \\ 0 & 0 & 1 \end{pmatrix}, \qquad \Phi_0 = \begin{pmatrix} 1 & 2 & 1 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{pmatrix}$$

$$T(\Delta t)^{-1} = \frac{1}{\sqrt{\Delta t}}\begin{pmatrix} 2/(\Delta t)^2 & 0 & 0 \\ 0 & \frac{1}{\Delta t} & 0 \\ 0 & 0 & 1 \end{pmatrix}, \qquad \Sigma_0 = \begin{pmatrix} 1/5 & 1/4 & 1/3 \\ 1/4 & 1/3 & 1/2 \\ 1/3 & 1/2 & 1 \end{pmatrix}$$

Krämer & Hennig

see also: Nordsieck methods, balanced representations

Linearisation

- Taylor-linearise

$$f(y,t) \approx f(y_0,t) + J_y f(y_0,t_0)(y-y_0)$$

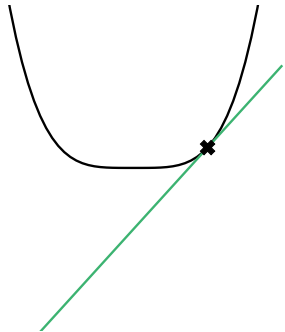
- Statistical linear regression

$$f(y,t) \approx Hy + b, \quad b \sim \mathrm{N}(\lambda, \Lambda)$$

- Approximate *Bosch et al.*

$$\mathcal{R}_1\mathbf{Y}(t) \approx \dot{Y}(t) - HY - b$$
$$\mathcal{R}_2\mathbf{Y}(t) \approx \ddot{Y}(t) - H_1 Y - H_2 \dot{Y} - b$$
$$\mathcal{R}_M\mathbf{Y}(t) \approx M\dot{Y}(t) - HY - b$$

- Plug into solver *Tronarp et al.*
- Iterate? *e.g. Tronarp et al., Krämer & Hennig*



Zeroth-order Linearisation

- ◇ Taylor-linearise *Schober et al., Kersting et al.*

$$f(y, t) \approx f(y_0, t)$$

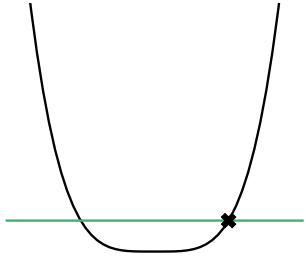
- ◇ Statistical linear regression *Kersting & Hennig*

$$f(y, t) \approx b, \quad b \sim \mathrm{N}(\lambda, \Lambda)$$

- ◇ Approximate *Bosch et al.*

$$\mathcal{R}_1 \mathbf{Y}(t) \approx \dot{Y}(t) - b$$
$$\mathcal{R}_2 \mathbf{Y}(t) \approx \ddot{Y}(t) - b$$
$$\mathcal{R}_M \mathbf{Y}(t) \approx M\dot{Y}(t) - b$$

- ◇ Plug into solver *e.g. Schober et al.*
- ◇ Iterate? *Arvanitidis et al.*



Why does linearisation matter?

- ◇ Jacobians → stability *c.f. Rosenbrock methods*
- ◇ First-order: A-stable *Tronarp et al.*
- ◇ First-order: for stiff problems
- ◇ Zeroth-order: lightning fast *Krämer et al.*

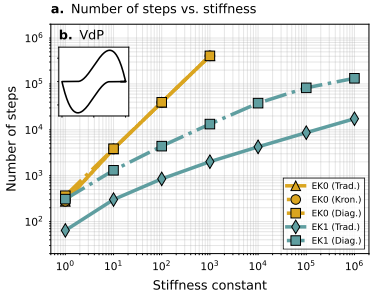
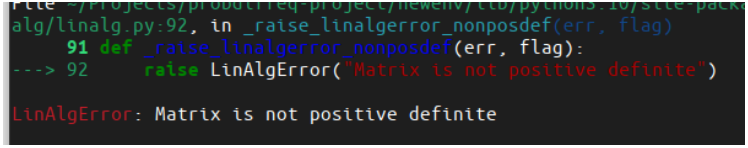


Figure: Krämer, Bosch, Schmidt, Hennig

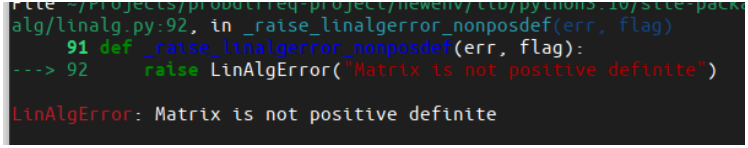


Square-root arithmetic

- ◇ Sometimes, covariance + covariance $\neq$ covariance
- ◇ Factorise $C = \sqrt{C}\sqrt{C}^\top$ (e.g. Cholesky)
- ◇ Then, QR-decomposition $QR = (\sqrt{C_1}^\top, \sqrt{C_2}^\top)$ yields *e.g. Grewal & Andrews*

$$R^\top R = (\sqrt{C_1}, \sqrt{C_2})(\sqrt{C_1}, \sqrt{C_2})^\top = C_1 + C_2$$

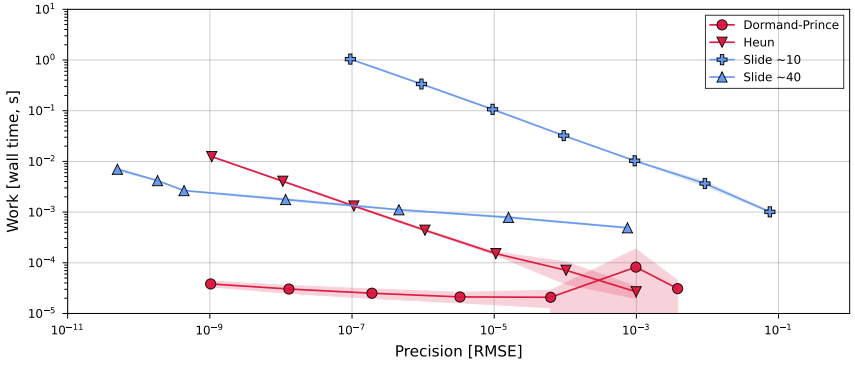
- ◇ Extrapolate covariances in "square-root arithmetic"
- ◇ Conditional covariances: same idea, different QR decomp. ($\rightarrow$ Wednesday)



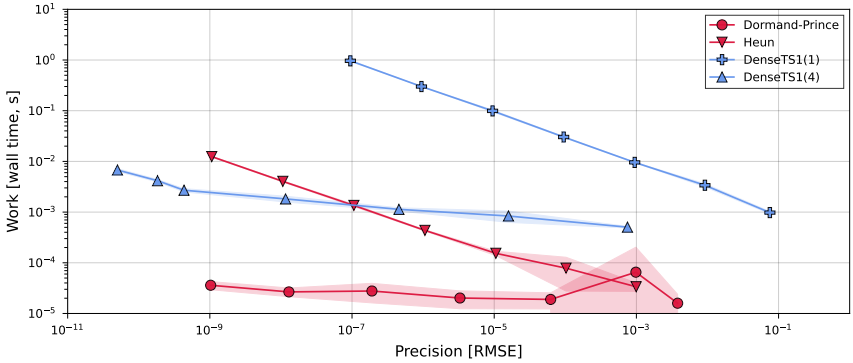
Algorithm: Sequential IVP solver

1. Initialise boundary conditions: $p(\mathbf{Y}(t_0) \mid Y(t_0) = y_0, \sigma)$
2. Initialise first grid point: $p(\mathbf{Y}(t_0) \mid Y(t_0) = y_0, (\mathcal{R}\mathbf{Y})(t_0) = 0, \sigma)$
3. While $t_n + \Delta t_n \leq 1$ (= terminal time):
   - 3.1 Extrapolate: Compute
     - $p(\mathbf{Y}(t_{n+1}) \mid \mathcal{R}\mathbf{Y}(t_{0:n}) = 0, Y(t_0) = y_0, \sigma)$
     - $p(\mathbf{Y}(t_n) \mid \mathbf{Y}(t_{n+1}), \mathcal{R}\mathbf{Y}(t_{0:n}) = 0, Y(t_0) = y_0, \sigma)$
   - 3.2 Linearise: $\mathcal{R}\mathbf{Y}(t_{n+1}) \approx H\mathbf{Y}(t_{n+1}) + b$
   - 3.3 Correct: $\approx p(\mathbf{Y}(t_{n+1}) \mid \mathcal{R}\mathbf{Y}(t_{0:n+1}) = 0, Y(t_0) = y_0, \sigma)$
   - 3.4 Estimate error
   - 3.5 Accept or reject step
   - 3.6 Propose "optimal" Δt_{n+1} and repeat
4. Do something fun with the terminal-value solution and (optionally) the backward transitions

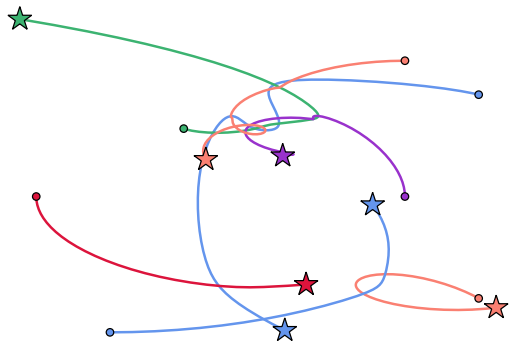
Where are we now?



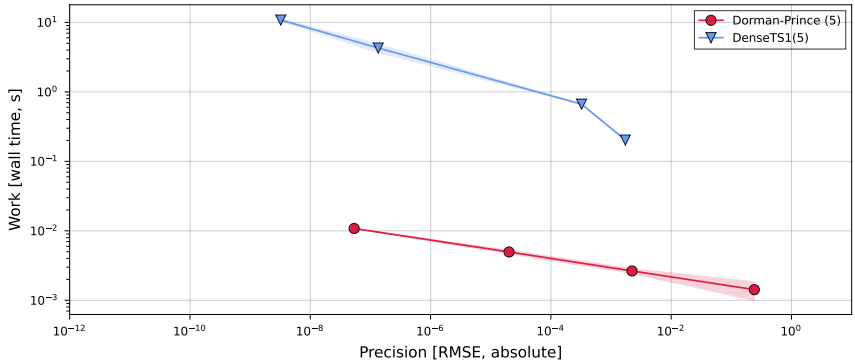
Where are we now?



A medium-dimensional problem



A medium-dimensional problem



$$O([\text{no. time-steps}][\text{no. derivatives}]^3[\text{ODE dimension}]^{\mathbf{3}})$$

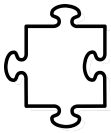


Image source: Pixabay (via Google Images)

Trading correlation for efficiency

Proposition I

If $\Sigma(\Delta t, \sigma)$ is block-diagonal, and if linearisation preserves block-diagonality, all posterior covariances are block-diagonal.

Proposition II

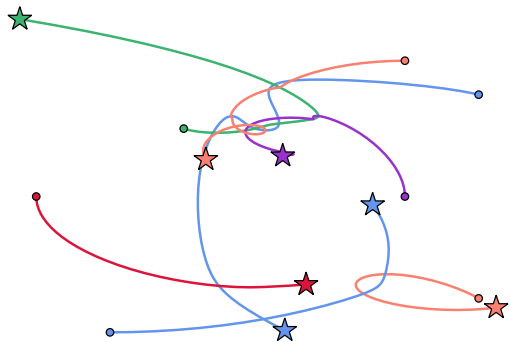
If $\Sigma(\Delta t, \sigma)$ has Kronecker structure, and if linearisation preserves Kronecker structure, all posterior covariances have Kronecker structure.

Corollary

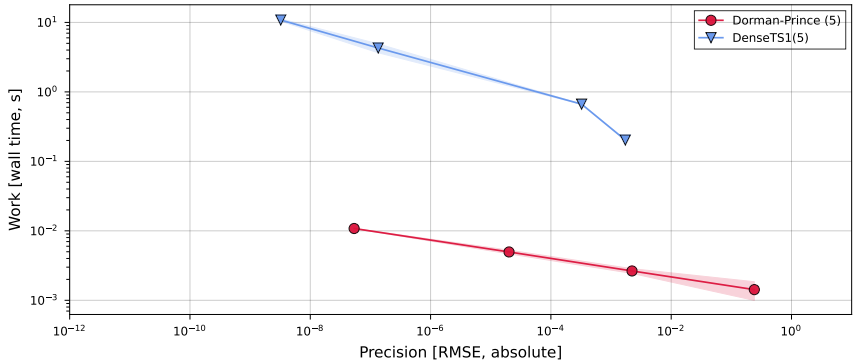
Kronecker-models can be implemented in $O(N(\nu^3 + d\nu^2))$, and block-diagonal models in $O(Nd\nu^3)$.

Krämer, Bosch, Schmidt, Hennig

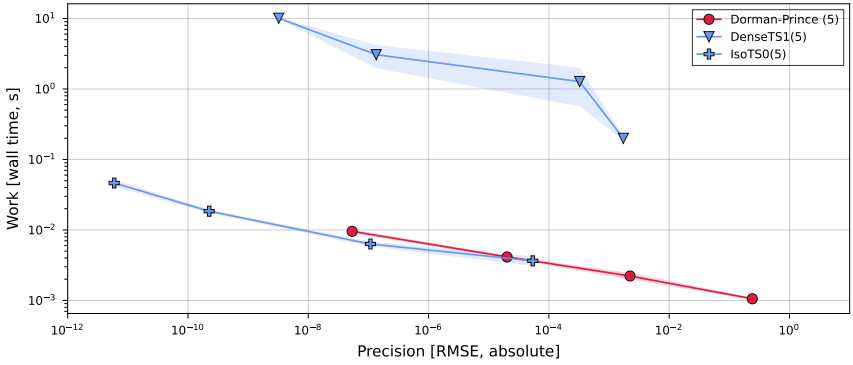
A medium-dimensional problem



A medium-dimensional problem



A medium-dimensional problem



And now the good stuff.

Parameter estimation: Neural ODEs

$$\frac{\mathrm{d}}{\mathrm{d}t}y(t) = \mathrm{NN}_\theta(y(t)), \quad \theta = ???$$

Observations: $\forall n: \; q_n = y(t_n) + \epsilon, \quad \epsilon \sim N(0, \rho^2)$

Parameter estimation

Posterior distribution:

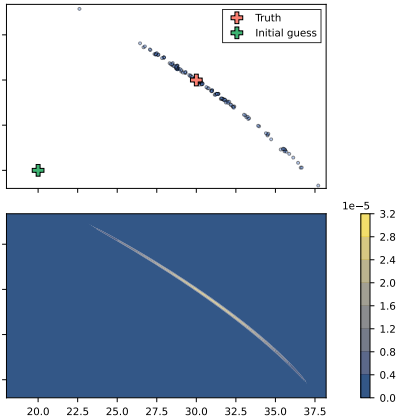
$$\mu(Y \mid \theta) := p(Y \mid \{\dot{Y}(t_n) = f_\theta(Y(t_n), t_n)\}_{n=0}^{N}, Y(t_0) = y_0, \sigma, \theta)$$

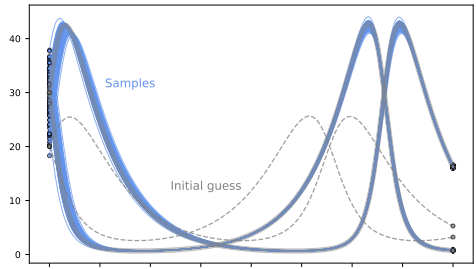
Average likelihood of observations over all IVP solutions:

$$M(\theta) = p(\{q_n\}_n \mid \theta) \approx \int p(q_n \mid Y(t_n))\mu(Y \mid \theta) dY$$

e.g. Kersting et al., Tronarp et al., Cockayne et al.

Probabilistic solvers & MCMC





High resolution images:

https://pnkraemer.github.io/probdiffeq/

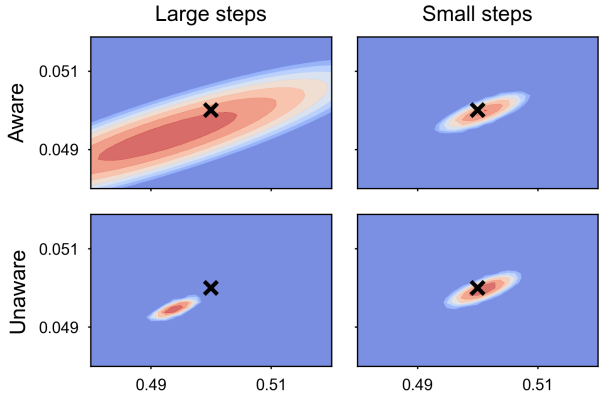


Figure: Kersting, et al.

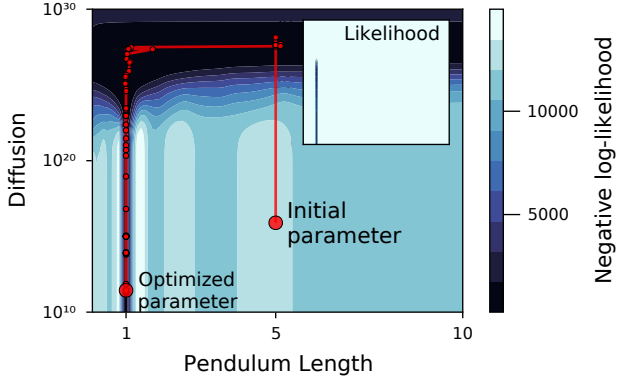


Figure: Tronarp, Bosch, Hennig

Conclusion:

- ◇ Integrated Wiener process priors: estimate IVP solution sequentially
- ◇ Discretise in time. But do it adaptively.
- ◇ Take Taylor series seriously, but not for linearisation.
- ◇ Use only probabilistic solvers for parameter estimation

Future quests:

- ◇ Deep dive into partial differential equations *e.g. Cockayne et al., Krämer et al.*
- ◇ Latent force estimation *Schmidt et al., see work by Särkkä*
- ◇ Other statistical perspectives on numerical differential equation solvers

  see works by Garegnani, Chkrebtii, Wang, Teymur, Pentland, Sullivan, Owhadi, Girolami, Oates, Pförtner, and others

The software landscape


```
pip install probnum
```

```
pip install probdiffeq
```


```
]add ProbNumDiffEq
]add Fenrir
```

References

- ◇ Tronarp, F., Kersting, H., Särkkä, S., & Hennig, P. (2019). Probabilistic solutions to ordinary differential equations as nonlinear Bayesian filtering: a new perspective. Statistics and Computing.
- ◇ Cockayne, J., Oates, C., Sullivan, T., & Girolami, M. (2017,). Probabilistic numerical methods for PDE-constrained Bayesian inverse problems. AIP Conference Proceedings
- ◇ Krämer, N., & Hennig, P. (2021). Linear-time probabilistic solution of boundary value problems. Neurips.
- ◇ Krämer, N., Schmidt, J., & Hennig, P. (2022). Probabilistic numerical method of lines for time-dependent partial differential equations. AISTATS.
- ◇ Bosch, N., Tronarp, F., & Hennig, P. (2022). Pick-and-mix information operators for probabilistic ODE solvers. AISTATS.
- ◇ Bosch, N., Hennig, P., & Tronarp, F. (2021). Calibrated adaptive probabilistic ODE solvers. AISTATS.

References

- ◇ Schober, M., Särkkä, S., & Hennig, P. (2019). A probabilistic model for the numerical solution of initial value problems. Statistics and Computing.
- ◇ Krämer, N., & Hennig, P. (2020). Stable implementation of probabilistic ODE solvers.
- ◇ Hartikainen, J., & Särkkä, S. (2010). Kalman filtering and smoothing solutions to temporal Gaussian process regression models. IEEE MLSP.
- ◇ Tronarp, F., Särkkä, S., & Hennig, P. (2021). Bayesian ODE solvers: the maximum a posteriori estimate. Statistics and Computing.
- ◇ Arvanitidis, G., Hauberg, S., Hennig, P., & Schober, M. (2019). Fast and robust shortest paths on manifolds learned from data. AISTATS.
- ◇ Kersting, H., Sullivan, T. J., & Hennig, P. (2020). Convergence rates of Gaussian ODE filters. Statistics and computing.
- ◇ Kersting, H., & Hennig, P. (2016). Active uncertainty calibration in Bayesian ODE solvers. UAI.
- ◇ Tronarp, F., Bosch, N., & Hennig, P. (2022). Fenrir: Physics-Enhanced Regression for Initial Value Problems. ICML.

References

- ◇ Kersting, H., Krämer, N., Schiegg, M., Daniel, C., Tiemann, M., & Hennig, P. (2020). Differentiable likelihoods for fast inversion of 'likelihood-free' dynamical systems. ICML.
- ◇ Hairer, E., Nørsett, S. P., & Wanner, G. (1993). Solving Ordinary Differential Equations: I, Nonstiff problems. Springer.
- ◇ Griewank, A., & Walther, A. (2008). Evaluating derivatives: principles and techniques of algorithmic differentiation. SIAM.
- ◇ Krämer, N., Bosch, N., Schmidt, J., & Hennig, P. (2022). Probabilistic ODE solutions in millions of dimensions. ICML.
- ◇ Grewal, M. S., & Andrews, A. P. (2014). Kalman filtering: Theory and Practice with MATLAB. John Wiley & Sons.
- ◇ Schmidt, J., Krämer, N., & Hennig, P. (2021). A probabilistic state space model for joint inference from differential equations and data. Neurips.