Three Differential Variants of the Buchberger-Moeller Algorithm

Stefan Schuster

Department of Informatics and Mathematics
University of Passau

4.3.2010 / Algebraic Oil Project Workshop

Outline

Outline

Transients

- Let us depict the **state** of a production system as the entirety of
  - valve positions
  - status (test vs. production) of wells
  - working points of wells
  - etc.
- Then **transients** take place whenever any of these is changed (startup, test, optimization, shutdown).
- On a larger time scale, the depletion of a reservoir can be regarded as a transient.

Transients - Challenges

- Transients are not fully understood and in general not predictable.
- E.g., after a test phase, it can be impossible to reach the same production state as it has been previously.

Prospects

A better understanding of the dynamic behaviour of transients leading to specific procedures that guide through the testing or optimization procedures.

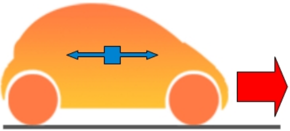
Dynamic vs. Steady-State Models

- In the first phase of the Algebraic Oil Project, among others, means to establish polynomial **steady-state models** for oil production scenarios were developed.
- Changes over time were not explicitly taken into account.
- However, it was always clear that one could (and should) do so.
- In this respect, the approach presented in the following is a **natural extension** of former results to dynamic situations.

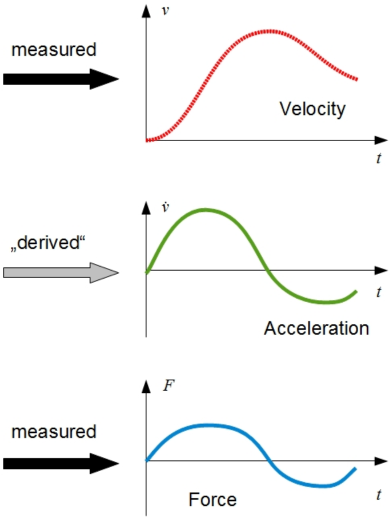
Goals

- **Extend** the AVI / ABM algorithms to the framework of differential algebra.
- **Implement** the developed methods.
- **Evaluate** and explore the methods in the context of transients scenarios coming from oil production.

Outline



$$\dot{v} \sim F$$


measured
v
Velocity
t
„derived“
$\dot{v}$
Acceleration
t
measured
F
Force
t

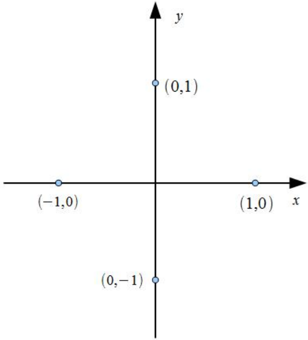
Desired features:

- **"Syntactical"** way to talk about derivatives.
- Values of derivatives do not need to be measured, but can be **deduced**.
- Relations must behave well under **further derivation**.

Differential Algebra provides the mathematical framework to this end.

Outline

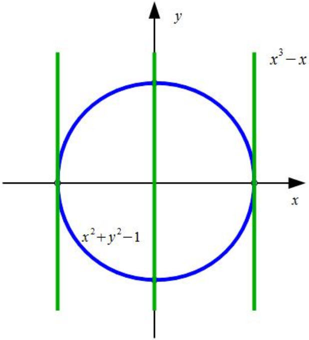
From Points to Polynomials



Goal:

- **Input:** a set of points $S = \{p_1, \dots, p_s\} \subset \mathbb{Q}^n$.
- **Output:** a Groebner basis of the vanishing ideal of S.

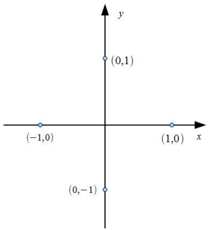
From Points to Polynomials



Goal:

- **Input:** a set of points $S = \{p_1, \ldots, p_s\} \subset \mathbb{Q}^n$.
- **Output:** a Groebner basis of the vanishing ideal of S.

Key Ideas of the BM Algorithm

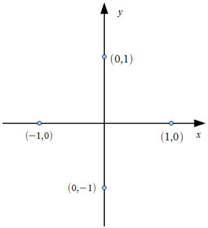


- **Order** the terms t_i.
- Consider the **evaluation vectors** $t_i(p_j)$ as columns.
- **Add** one column $(t_i(p_j))_j$ after the other,
- and find **linear dependencies** among the columns.
- "Stop when done".

	1	x	y	x^2	xy	y^2	$\dots$
$p_1 = (1,0)$							
$p_2 = (0,1)$							
$p_3 = (-1,0)$							
$p_4 = (0,-1)$							

$$V(S) = \{\}$$

Key Ideas of the BM Algorithm

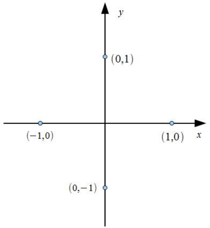


- **Order** the terms t_i.
- Consider the **evaluation vectors** $t_i(p_j)$ as columns.
- **Add** one column $(t_i(p_j))_j$ after the other,
- and find **linear dependencies** among the columns.
- "Stop when done".

	1	x	y	x^2	xy	y^2	$\dots$
$p_1 = (1,0)$	1	1					
$p_2 = (0,1)$	1	0					
$p_3 = (-1,0)$	1	-1					
$p_4 = (0,-1)$	1	0					

$$V(S) = \{\}$$

Key Ideas of the BM Algorithm

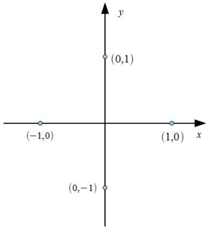


- **Order** the terms t_i.
- Consider the **evaluation vectors** $t_i(p_j)$ as columns.
- **Add** one column $(t_i(p_j))_j$ after the other,
- and find **linear dependencies** among the columns.
- "Stop when done".

	1	x	y	x^2	xy	y^2	$\dots$
$p_1 = (1,0)$	1	1	0				
$p_2 = (0,1)$	1	0	1				
$p_3 = (-1,0)$	1	−1	0				
$p_4 = (0,-1)$	1	0	−1				

$$V(S) = \{\}$$

Key Ideas of the BM Algorithm

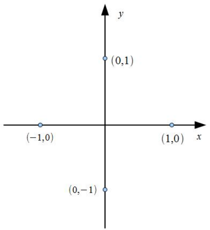


- **Order** the terms t_i.
- Consider the **evaluation vectors** $t_i(p_j)$ as columns.
- **Add** one column $(t_i(p_j))_j$ after the other,
- and find **linear dependencies** among the columns.
- "Stop when done".

	1	x	y	x^2	xy	y^2	...
$p_1 = (1,0)$	1	1	0	1			
$p_2 = (0,1)$	1	0	1	0			
$p_3 = (-1,0)$	1	-1	0	1			
$p_4 = (0,-1)$	1	0	-1	0			

$$V(S) = \{\}$$

Key Ideas of the BM Algorithm

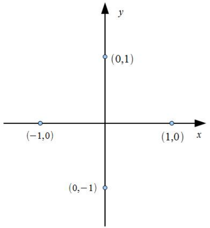


- **Order** the terms t_i.
- Consider the **evaluation vectors** $t_i(p_j)$ as columns.
- **Add** one column $(t_i(p_j))_j$ after the other,
- and find **linear dependencies** among the columns.
- "Stop when done".

	1	x	y	x^2	xy	y^2	...
$p_1 = (1,0)$	1	1	0	1	0		
$p_2 = (0,1)$	1	0	1	0	0		
$p_3 = (-1,0)$	1	-1	0	1	0		
$p_4 = (0,-1)$	1	0	-1	0	0		

$$V(S) = \{xy\}$$

Key Ideas of the BM Algorithm

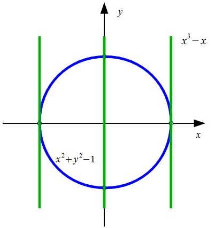


- **Order** the terms t_i.
- Consider the **evaluation vectors** $t_i(p_j)$ as columns.
- **Add** one column $(t_i(p_j))_j$ after the other,
- and find **linear dependencies** among the columns.
- "Stop when done".

	1	x	y	x^2	xy	y^2	...
$p_1 = (1,0)$	1	1	0	1	0	0	
$p_2 = (0,1)$	1	0	1	0	0	1	
$p_3 = (-1,0)$	1	-1	0	1	0	0	
$p_4 = (0,-1)$	1	0	-1	0	0	1	

$$V(S) = \{xy, y^2 + x^2 - 1\}$$

Key Ideas of the BM Algorithm

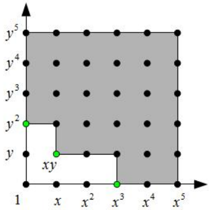


- **Order** the terms t_i.
- Consider the **evaluation vectors** $t_i(p_j)$ as columns.
- **Add** one column $(t_i(p_j))_j$ after the other,
- and find **linear dependencies** among the columns.
- "Stop when done".

	1	x	y	x^2	xy	y^2	$\dots$
$p_1 = (1,0)$	1	1	0	1	0	0	$\dots$
$p_2 = (0,1)$	1	0	1	0	0	1	$\dots$
$p_3 = (-1,0)$	1	-1	0	1	0	0	$\dots$
$p_4 = (0,-1)$	1	0	-1	0	0	1	$\dots$

$$V(S) = \{x^3 - x, xy, y^2 + x^2 - 1\}$$

Key Ideas of the BM Algorithm



- **Order** the terms t_i.
- Consider the **evaluation vectors** $t_i(p_j)$ as columns.
- **Add** one column $(t_i(p_j))_j$ after the other,
- and find **linear dependencies** among the columns.
- “Stop when done”.

	1	x	y	x^2	xy	y^2	$\dots$
$p_1 = (1,0)$	1	1	0	1	0	0	$\dots$
$p_2 = (0,1)$	1	0	1	0	0	1	$\dots$
$p_3 = (-1,0)$	1	-1	0	1	0	0	$\dots$
$p_4 = (0,-1)$	1	0	-1	0	0	1	$\dots$

$$V(S) = \{x^3 - x, xy, y^2 + x^2 - 1\}$$

Outline

Introducing "Differential"

- Original polynomial ring: $\mathbb{Q}[y_1, \ldots, y_n]$.
- We want to speak about **derivatives** of the y_i:

$$\dot{y}_2 = -y_1$$

- So we add them

$$D = \mathbb{Q}[y_1, \ldots, y_n, \dot{y}_1, \ldots, \dot{y}_n, \ddot{y}_1, \ldots, \ddot{y}_n, \ldots] = \mathbb{Q}\{y_1, \ldots, y_n\}$$

  to obtain the **differential polynomial ring** in the indeterminates y_i.
- Besides the **degree**, d.p.'s also have an **order**.

Introducing "Differential"

- Besides addition and multiplication, introduce a new operation, **differentiation**

$$\partial : D \longrightarrow D : f \mapsto \partial f$$

- E.g., the 1st and 2nd derivations of $f = xy \in \mathbb{Q}\{x, y\}$ are

$$\partial f = \dot{x}y + x\dot{y}$$

$$\partial^2 f = \ddot{x}y + 2\dot{x}\dot{y} + x\ddot{y}$$

- Differential ideals I are those ideals of D which are **closed** under taking derivatives:

$$f \in I \Longrightarrow \partial^k f \in I \text{ for all } k \in \mathbb{N}$$

.

Intordocing "Differential"

- Differential algebra is the algebraic treatment of differential equations.
- Differential ideals are the algebraic way to describe solutions of differential equations.

Remarks about Differential Ideals

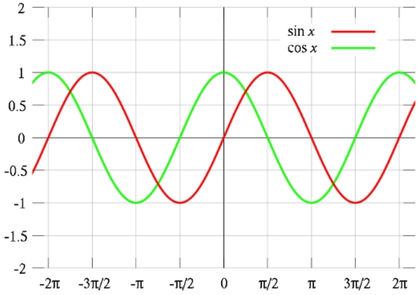
- D is not Noetherian.
- There are differential ideals that are not finitely generated.
- There are differential ideals that are not even recursive.

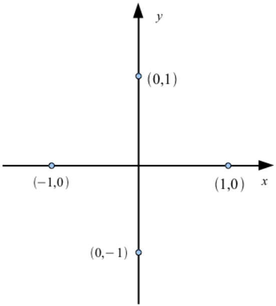
Differential Points

- In order to evaluate differential polynomials we need “values” for all differential indeterminates $y_1, \ldots, y_n, \dot{y}_1, \ldots, \dot{y}_n, \ldots$.
- We therefore require a possibly **infinite** amount of information.

Let us try to understand this at the previous example.

Differential Points





Differential Points

- Evaluate x as $\cos(p)$, $p \in \{0, \frac{\pi}{2}, \pi, \frac{3\pi}{2}\}$
- and y as $\sin(p)$.
- Now we extend the table downwards, taking derivatives of cos and sin into account.

	1	x	y	x^2	xy	y^2	$\dots$
φ_{p_1}	1	1	0	1	0	0	$\dots$
φ_{p_2}	1	0	1	0	0	1	$\dots$
φ_{p_3}	1	-1	0	1	0	0	$\dots$
φ_{p_4}	1	0	-1	0	0	1	$\dots$

Differential Points

	1	x	y	x^2	xy	y^2	$\dots$
φ_{p_1}	1	1	0	1	0	0	$\dots$
φ_{p_2}	1	0	1	0	0	1	$\dots$
φ_{p_3}	1	-1	0	1	0	0	$\dots$
φ_{p_4}	1	0	-1	0	0	1	$\dots$
∂	0	$\dot{x}$	$\dot{y}$	$2\dot{x}x$	$\dot{x}y+x\dot{y}$	$2\dot{y}y$	$\dots$
φ_{p_1}	0	0	1	0	1	0	$\dots$
φ_{p_2}	0	-1	0	0	-1	0	$\dots$
φ_{p_3}	0	0	-1	0	1	0	$\dots$
φ_{p_4}	0	1	0	0	-1	0	$\dots$
∂^2	0	$\ddot{x}$	$\ddot{y}$	$\dots$			
φ_{p_1}	$\dots$						
φ_{p_2}	$\dots$						
φ_{p_3}	$\dots$						
φ_{p_4}	$\dots$						

Differential Points

	1	x	y	x^2	xy	y^2	$\dots$	$\dot{x}$	$\dot{y}$	$\dots$
φ_{p_1}	1	1	0	1	0	0	$\dots$			
φ_{p_2}	1	0	1	0	0	1	$\dots$			
φ_{p_3}	1	-1	0	1	0	0	$\dots$			
φ_{p_4}	1	0	-1	0	0	1	$\dots$			
∂	1	$\dot{x}$	$\dot{y}$	$2\dot{x}x$	$\dot{x}y + x\dot{y}$	$2\dot{y}y$	$\dots$			
φ_{p_1}	0	0	1	0	1	0	$\dots$			
φ_{p_2}	0	-1	0	0	-1	0	$\dots$			
φ_{p_3}	0	0	-1	0	1	0	$\dots$			
φ_{p_4}	0	1	0	0	-1	0	$\dots$			
∂^2	1	$\ddot{x}$	$\ddot{y}$	$\dots$						
φ_{p_1}	$\dots$									
φ_{p_2}	$\dots$									
φ_{p_3}	$\dots$									
φ_{p_4}	$\dots$									

Differential Points

- How can we **control** the **downward growth**, i.e., how can we handle higher order derivatives?
- How can we **control** and possibly limit the **growth to the right**, i.e., higher degree differential polynomials?
- After all, is this really **useful** for practical applications?

Differential Points

- How can we **control** the **downward growth**, i.e., how can we handle higher order derivatives?
- How can we **control** and possibly limit the **growth to the right**, i.e., higher degree differential polynomials?
- After all, is this really **useful** for practical applications?

Differential Points

- How can we **control** the **downward growth**, i.e., how can we handle higher order derivatives?
- How can we **control** and possibly limit the **growth to the right**, i.e., higher degree differential polynomials?
- After all, is this really **useful** for practical applications?

Outline

Overview

How can we control the downward growth, i.e., how can we handle higher order derivatives?

Variant 1

Restrict the class of possible "points".

How can we control and possibly limit the growth to the right, i.e., how can we handle higher degree differential polynomials?

Variant 2

Restrict the order of the derivatives.

After all, is this really useful for practical applications?

Variant 3

Restrict the order of the derivatives and add approximate computations to handle noisy data.

Overview

How can we control the downward growth, i.e., how can we handle higher order derivatives?

Variant 1

Restrict the class of possible "points".

How can we control and possibly limit the growth to the right, i.e., how can we handle higher degree differential polynomials?

Variant 2

Restrict the order of the derivatives.

After all, is this really useful for practical applications?

Variant 3

Restrict the order of the derivatives and add approximate computations to handle noisy data.

Overview

How can we control the downward growth, i.e., how can we handle higher order derivatives?

Variant 1

Restrict the class of possible "points".

How can we control and possibly limit the growth to the right, i.e., how can we handle higher degree differential polynomials?

Variant 2

Restrict the order of the derivatives.

After all, is this really useful for practical applications?

Variant 3

Restrict the order of the derivatives and add approximate computations to handle noisy data.

Overview

How can we control the downward growth, i.e., how can we handle higher order derivatives?

Variant 1

Restrict the class of possible "points".

How can we control and possibly limit the growth to the right, i.e., how can we handle higher degree differential polynomials?

Variant 2

Restrict the order of the derivatives.

After all, is this really useful for practical applications?

Variant 3

Restrict the order of the derivatives and add approximate computations to handle noisy data.

Overview

How can we control the downward growth, i.e., how can we handle higher order derivatives?

Variant 1

Restrict the class of possible “points”.

How can we control and possibly limit the growth to the right, i.e., how can we handle higher degree differential polynomials?

Variant 2

Restrict the order of the derivatives.

After all, is this really useful for practical applications?

Variant 3

Restrict the order of the derivatives and add approximate computations to handle noisy data.

Overview

How can we control the downward growth, i.e., how can we handle higher order derivatives?

Variant 1

Restrict the class of possible “points”.

How can we control and possibly limit the growth to the right, i.e., how can we handle higher degree differential polynomials?

Variant 2

Restrict the order of the derivatives.

After all, is this really useful for practical applications?

Variant 3

Restrict the order of the derivatives and add approximate computations to handle noisy data.

Variant 1

Restrict the class of possible “points”.

- Idea: Assume the evaluation of the differential indeterminates is given by **recursive definitions**.
- Then it can be shown that the corresponding vanishing ideal must be finitely generated.
- If we consider only polynomials up to a certain degree, the order of the derivatives can be bounded.
- We therefore arrive at a procedure that eventually will give us a finite Groebner basis of the differential vanishing ideal.
- However, we currently do not know when to stop.

Variant 1 - The Example Continued

- For the cosine/sine example we find from the recursive definition of the evaluation, that the differential polynomials

$$\ddot{x}+x, \quad \ddot{y}+y$$

must be contained in the vanishing ideal.
- We can carry out all computations in $P = \mathbb{Q}[x, \dot{x}, y, \dot{y}]$.
- ∂ is replaced by a homogeneous $\mathbb{Q}$-linear mapping $\Delta : P \to P$.
- If we consider $P_{\leq 10}$, we compute the following set of polynomials:

$$V_{\partial}(S)_{\leq 10} = \{-y+\dot{x},\ \dot{y}+x,\ \dot{x}^2+x^2-1\}$$

Variant 1 - The Example Continued

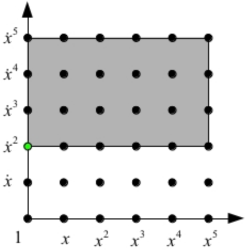
- For the cosine/sine example we find from the recursive definition of the evaluation, that the differential polynomials

$$\ddot{x}+x, \qquad \ddot{y}+y$$

  must be contained in the vanishing ideal.
- We can carry out all computations in $P=\mathbb{Q}[x,\dot{x},y,\dot{y}]$.
- ∂ is replaced by a homogeneous $\mathbb{Q}$-linear mapping $\Delta : P \longrightarrow P$.
- If we consider $P_{\leq 10}$, we compute the following set of polynomials:

$$V_\partial(S)_{\leq 10}=\{-y+\dot{x},\ \dot{y}+x,\ \dot{x}^2+x^2-1\}$$

Variant 1 - Termination



- The corresponding vanishing ideals are not 0-dimensional.
- The Groebner basis must be finite.
- If we can find a degree bound is still an open question.

Variant 1 - Further Remarks

The recursive definitions of the evaluation maps is directly connected to

- particular solutions of linear differential equations with constant coefficients,
- the Taylor series of the solutions,
- certain classes of meromorphic functions.

Variant 2 - Exact but Limited Order

Restrict the order of the derivatives.

- Consider only derivatives up to order N.
- Choose an order $0 \leq M \leq N$ up to which you are interested in.
- Even though the orders are restricted, we can guarantee that with f also $\partial f, \ldots, \partial^{N-M} f$ are contained in the resulting ideal $V_{N,M}(S)$.
- Polynomials that do not survive this requirement are filtered out.

$$V_\partial(S) \cap D_M \subseteq V_{N,M}(S) \subseteq V(S) \cap D_M$$

Variant 2 - The Example Continued

	1	x	y	xy	$\dot{x}$
φ_{p_1}	1	1	0	0	0
φ_{p_2}	1	0	1	0	-1
φ_{p_3}	1	-1	0	0	0
φ_{p_4}	1	0	-1	0	1

	1	$\dot{x}$	$\dot{y}$	$\dot{x}y + x\dot{y}$	$\ddot{x}$
φ_{p_1}	0	0	1	1	0
φ_{p_2}	0	-1	0	-1	-1
φ_{p_3}	0	0	-1	1	0
φ_{p_4}	0	1	0	-1	1

- $N = 2$, $M = 1$
- xy is filtered out.
- $\dot{x} - y$ survives.

$$V_{2,1} = \{x^4 - x^2, x^2 + y^2 - 1, \dot{x} - y, \dot{y} + x\}$$

Variant 2 - The Example Continued

	1	x	y	xy	$\dot{x}$
φ_{p_1}	1	1	0	0	0
φ_{p_2}	1	0	1	0	-1
φ_{p_3}	1	-1	0	0	0
φ_{p_4}	1	0	-1	0	1

	1	$\dot{x}$	$\dot{y}$	$\dot{x}y + x\dot{y}$	$\ddot{x}$
φ_{p_1}	0	0	1	1	0
φ_{p_2}	0	-1	0	-1	-1
φ_{p_3}	0	0	-1	1	0
φ_{p_4}	0	1	0	-1	1

- $N = 2$, $M = 1$
- xy is filtered out.
- $\dot{x} - y$ survives.

$$V_{2,1} = \{x^4 - x^2, x^2 + y^2 - 1, \dot{x} - y, \dot{y} + x\}$$

Variant 2 - The Example Continued

	1	x	y	xy	$\dot{x}$
φ_{p_1}	1	1	0	0	0
φ_{p_2}	1	0	1	0	-1
φ_{p_3}	1	-1	0	0	0
φ_{p_4}	1	0	-1	0	1

	1	$\dot{x}$	$\dot{y}$	$\dot{x}y + x\dot{y}$	$\ddot{x}$
φ_{p_1}	0	0	1	1	0
φ_{p_2}	0	-1	0	-1	-1
φ_{p_3}	0	0	-1	1	0
φ_{p_4}	0	1	0	-1	1

- $N = 2$, $M = 1$
- xy is filtered out.
- $\dot{x} - y$ survives.

$$V_{2,1} = \{x^4 - x^2, x^2 + y^2 - 1, \dot{x} - y, \dot{y} + x\}$$

Variant 2 - The Example Continued

	1	x	y	xy	$\dot{x}$
φ_{p_1}	1	1	0	0	0
φ_{p_2}	1	0	1	0	-1
φ_{p_3}	1	-1	0	0	0
φ_{p_4}	1	0	-1	0	1

	1	$\dot{x}$	$\dot{y}$	$\dot{x}y + x\dot{y}$	$\ddot{x}$
φ_{p_1}	0	0	1	1	0
φ_{p_2}	0	-1	0	-1	-1
φ_{p_3}	0	0	-1	1	0
φ_{p_4}	0	1	0	-1	1

- $N = 2$, $M = 1$
- xy is filtered out.
- $\dot{x} - y$ survives.

$$V_{2,1} = \{x^4 - x^2, x^2 + y^2 - 1, \dot{x} - y, \dot{y} + x\}$$

Variant 3 - The Approximate Case

Restrict the order of the derivatives and add approximate computations to handle noisy data.

- Use the ideas from variant 2.
- Use the methods and techniques known from the AVI / ABM methods.

Let us see how it works. . .

Variant 3 - The Approximate Case

Restrict the order of the derivatives and add approximate computations to handle noisy data.

- Use the ideas from variant 2.
- Use the methods and techniques known from the AVI / ABM methods.

Let us see how it works. . .

Variant 3 - The Approximate Case

Restrict the order of the derivatives and add approximate computations to handle noisy data.

- Use the ideas from variant 2.
- Use the methods and techniques known from the AVI / ABM methods.

Let us see how it works...

How to Derive Derivatives

- Goal: Given noisy, one-dimensional time series of data, provide higher order derivatives.
- Established interpolation and curve fitting methods are available.
- Careful selection and preparation will be necessary.

How to Derive Derivatives

- Goal: Given noisy, one-dimensional time series of data, provide higher order derivatives.
- Established interpolation and curve fitting methods are available.
- Careful selection and preparation will be necessary.

How to Derive Derivatives

- Goal: Given noisy, one-dimensional time series of data, provide higher order derivatives.
- Established interpolation and curve fitting methods are available.
- Careful selection and preparation will be necessary.

Conclusion

- We demonstrated differential version of AVI suitable for differential modeling.
- The methods have been implemented in CoCoAL.
- Efficient implementation in C++ is under construction.

Conclusion

- We demonstrated differential version of AVI suitable for differential modeling.
- The methods have been implemented in CoCoAL.
- Efficient implementation in C++ is under construction.

Conclusion

- We demonstrated differential version of AVI suitable for differential modeling.
- The methods have been implemented in CoCoAL.
- Efficient implementation in C++ is under construction.

Outlook - Possible Next Steps

- Evaluate the methods with real world data.
- Adapt the methods according to the output of the evaluation.
- Compare the differential models to other approaches, e.g., the inflow models used in reservoir engineering.

Outlook - Possible Next Steps

- Evaluate the methods with real world data.
- Adapt the methods according to the output of the evaluation.
- Compare the differential models to other approaches, e.g., the inflow models used in reservoir engineering.

Outlook - Possible Next Steps

- Evaluate the methods with real world data.
- Adapt the methods according to the output of the evaluation.
- Compare the differential models to other approaches, e.g., the inflow models used in reservoir engineering.

Thank you for your attention!

Any questions?