An Introduction to Differential Algebra

Alexander Wittig[1], P. Di Lizia, R. Armellin, et al.[2]

[1] *ESA Advanced Concepts Team (TEC-SF)*
[2] *Dinamica SRL, Milan*

Outline

Overview

Differential Algebra

- A numerical technique based on algebraic manipulation of polynomials
- Its computer implementation
- Algorithms using this numerical technique with applications in physics, math and engineering.

Various aspects of what we call *Differential Algebra* are known under other names:

- Truncated Polynomial Series Algebra (TPSA)
- Automatic forward differentiation
- Jet Transport

History of Differential Algebra

(*Incomplete*) History of Differential Algebra and similar Techniques

- Introduced in Beam Physics (Berz, 1987)
  Computation of transfer maps in particle optics
- Extended to Verified Numerics (Berz and Makino, 1996)
  Rigorous numerical treatment including truncation and round-off errors for computer assisted proofs
- Taylor Integrator (Jorba et al., 2005)
  Numerical integration scheme based on arbitrary order expansions
- Applications to Celestial Mechanics (Di Lizia, Armellin, 2007)
  Uncertainty propagation, Two-Point Boundary Value Problem, Optimal Control, Invariant Manifolds...
- Jet Transport (Gomez, Masdemont, et al., 2009)
  Uncertainty Propagation, Invariant Manifolds, Dynamical Structure

Differential Algebra

A numerical technique to automatically compute high order Taylor expansions of functions

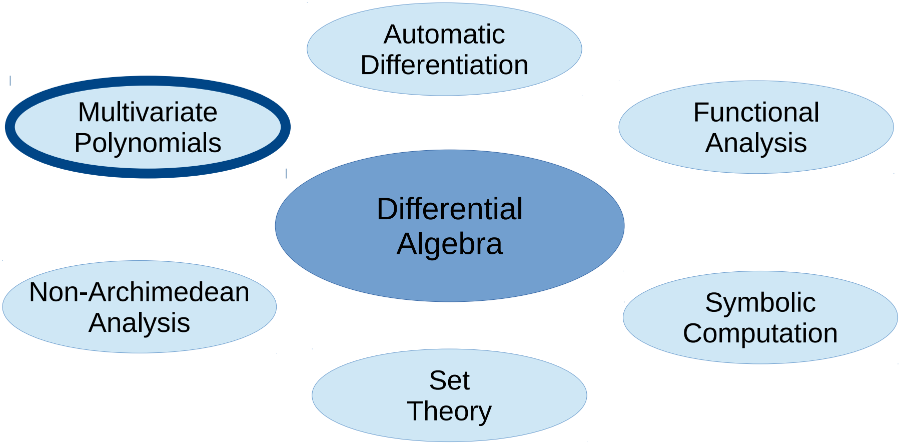
$$f(\vec{x_0} + \vec{\delta x}) \approx f(\vec{x_0}) + f'(\vec{x_0}) \cdot \vec{\delta x} + \cdots + \frac{1}{n!} f^{(n)}(\vec{x_0}) \cdot \vec{\delta x}^{\,n}$$

and algorithms to manipulate these expansions.

Can be conceptualized in various ways from different view points:

- Multivariate Polynomials
- Functional Analysis
- Set Theory
- Automatic Differentiation
- Non-Archimedean Analysis
- (Symbolic Computation)

Multivariate Polynomials



Multivariate Polynomials: Motivation

Motivation: What does an expression like this mean?

$$r = \frac{x \cdot y + 1}{\sqrt{1 + x^2}}$$

- Instructions of basic operations to be performed in a certain order:

1. $r_1 \leftarrow x \cdot y$
2. $r_1 \leftarrow r_1 + 1$
3. $r_2 \leftarrow x \cdot x$
4. $r_2 \leftarrow 1 + r_2$
5. $r_2 \leftarrow \sqrt{r_2}$
6. $r \leftarrow r_1 / r_2$

Multivariate Polynomials: Motivation

x, y can be anything for which basic operations used (e.g. $+, -, \cdot, /, \sqrt{}, \dots$) are defined in "useful" way:

- Abstract mathematical entities:
  - real number ($\mathbb{R}$)
  - matrices ($\mathbb{R}^{n\times n}$)
  - complex number ($\mathbb{C}$)
  - functions (e.g. C^r)
- Computer representations of mathematical entities
  - floating point numbers
  - DA objects

Key idea of DA

Replace all algebraic operations between numbers by ones that act on (a suitably chosen subset of) polynomials instead.

Multivariate Polynomials: Definition

1. Ring of polynomials $p(x) = \sum_{i=0} a_i x^i$
   - Natural *Addition*, *Subtraction*, *Multiplication* of polynomials
   - Problems:
     - order of polynomials not limited, grows under multiplication
     - infinite dimensional
     - not well suited for computations

Multivariate Polynomials: Definition

1. Ring of polynomials $p(x) = \sum_{i=0} a_i x^i$
   - Natural *Addition*, *Subtraction*, *Multiplication* of polynomials
   - Problems:
     - order of polynomials not limited, grows under multiplication
     - infinite dimensional
     - not well suited for computations
2. Algebra of truncated polynomials $p(x) = \sum_{i=0}^{n} a_i x^i$
   - Truncate all results to a fixed order n
   - Finite dimensional space, hence computable
   - Space ${}_nD_v$ of polynomials of order up to n in v variables has $\frac{(n+v)!}{n!v!}$ dimensions
   - Not a ring or field: e.g. many nil-potent elements (zero divisors): e.g. $x, x - x^2, \ldots$

Multivariate Polynomials: Intrinsics

Can also introduce *division* and *intrinsic functions* on ${}_nD_v$.

- Division: $\frac{1}{P} \in {}_nD_v$ such that $P \cdot \frac{1}{P} = 1$
  - Does not always exist: $P(x) = x$ has no multiplicative inverse
  - Exists for all polynomials with non-zero constant part
  - Does not exist in ring of polynomials!
- Other *intrinsic functions* (e.g. $\sqrt{\ }$, sin, cos, exp, ...)
  - can be defined appropriately on ${}_nD_v$
  - often with similar restrictions as division, or stricter (e.g. $\sqrt{\ }$)

Multivariate Polynomials: Intrinsics

Can also introduce *division* and *intrinsic functions* on ${}_nD_v$.

- Division: $\frac{1}{P} \in {}_nD_v$ such that $P \cdot \frac{1}{P} = 1$
  - Does not always exist: $P(x) = x$ has no multiplicative inverse
  - Exists for all polynomials with non-zero constant part
  - Does not exist in ring of polynomials!
- Other *intrinsic functions* (e.g. $\sqrt{\ }$, sin, cos, exp, ...)
  - can be defined appropriately on ${}_nD_v$
  - often with similar restrictions as division, or stricter (e.g. $\sqrt{\ }$)

Result

Now we can evaluate expressions such as

$$r = \frac{x \cdot y + 1}{\sqrt{1 + x^2}}$$

in DA arithmetic using polynomials.

Multivariate Polynomials: Differential Structure

Last thing: Add derivation ∂ and inverse derivation ∂^{-1} operators to obtain *differential* algebra.

- ∂_x: Simple polynomial derivation w.r.t. independent variable x
- ∂_x^{-1}: Simple polynomial integration w.r.t. independent variable x

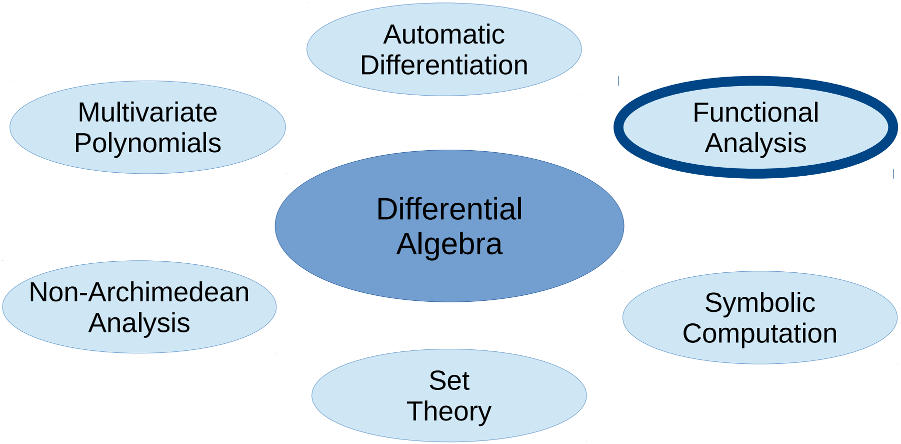
Now we can evaluate even complicated operators *directly in DA*:

$$g(x, y, z) = \int\int \frac{d}{dy} \exp\left(\frac{\sin(x)\cdot\cos(y)+1}{\sqrt{1+x^2+y^2+z^2}}\right) dx\, dz$$

Result: Differential Algebra

Together with the *right* definitions for all these operations, we extended the basic polynomial algebra to the Differential Algebra (DA).

Differential Algebra



Floating Point Arithmetic

Taking a step back

$\mathbb{R}$ is infinite. How does arithmetic on $\mathbb{R}$ get into the computer?

Answer: floating point numbers ($\mathbb{F}$)

$$x = \pm m \cdot 2^e$$

- Mantissa *m* and exponent *e* are integers in some range
- Approximate representation of real numbers $\mathbb{R}$
  - mantissa represents the “most significant digits”
  - exponent represents the “magnitude”
- All algebraic operations on $\mathbb{F}$ defined to “approximate” the corresponding real operation on $\mathbb{R}$

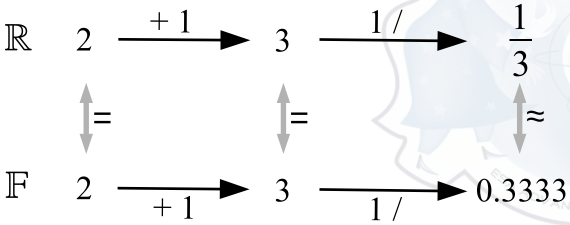
Floating Point Arithmetic: Example

In $\mathbb{R}$ and hypothetical $\mathbb{F}$ with 4 significant decimal digits, evaluate

$$\frac{1}{x+1} \quad \text{for } x = 2.$$

List of operations:

- start with $x = 2$
- perform $+1$ operation
- perform $1/$ operation



Functional Analysis on Computers

Idea:

Use Taylor Expansions around 0 as approximate computer representations of functions in $C^r(0)$ function space.

- Each function $f \in C^r(0)$ is represented by a Taylor Expansion $\mathcal{T}_f$ of order r.
- $\mathcal{T}_f$ approximates f just like floating point numbers $\mathbb{F}$ approximate real numbers $\mathbb{R}$

Functional Analysis on Computers

Idea:

Use Taylor Expansions around 0 as approximate computer representations of functions in $C^r(0)$ function space.

- Each function $f \in C^r(0)$ is represented by a Taylor Expansion $\mathcal{T}_f$ of order r.
- $\mathcal{T}_f$ approximates f just like floating point numbers $\mathbb{F}$ approximate real numbers $\mathbb{R}$

Example for $n = 3$

$f(x) = 1 + x + \frac{x^2}{2} + \frac{x^3}{6}$ $\quad g(x) = \exp(x)$ $\quad h(x) = \exp(x) + x^3 \cdot \sin(x)$

All three functions f, g, and h are represented by $f(x)$.

Functional Analysis on Computers

Just as for floating point numbers, DA operations are defined to “approximate” operations in C^r:

- Binary operators $\times$ and DA equivalent $\otimes$ (e.g. $+, -, \cdot, /$)

$$\mathcal{T}_f(x) \otimes \mathcal{T}_g(x) = \mathcal{T}_{f \times g}(x)$$

- Intrinsic functions $g(x)$ and DA equivalent $G(x)$ (e.g. sin, cos)

$$G(\mathcal{T}_f(x)) = \mathcal{T}_{g(f(x))}.$$

where $\mathcal{T}_f(x)$ is Taylor expansion of f around 0 up to fixed order n.

Functional Analysis on Computers

Just as for floating point numbers, DA operations are defined to "approximate" operations in C^r:

- Binary operators $\times$ and DA equivalent $\otimes$ (e.g. $+, -, \cdot, /$)

$$\mathcal{T}_f(x) \otimes \mathcal{T}_g(x) = \mathcal{T}_{f \times g}(x)$$

- Intrinsic functions $g(x)$ and DA equivalent $G(x)$ (e.g. sin, cos)

$$G(\mathcal{T}_f(x)) = \mathcal{T}_{g(f(x))}.$$

where $\mathcal{T}_f(x)$ is Taylor expansion of f around 0 up to fixed order n.

How is it done?

Don't worry about the *how* this is implemented, just accept it *is* implemented correctly for you by someone (=us)!
You also accepted that there exist algorithms to compute $1/3 \approx 0.3333$.

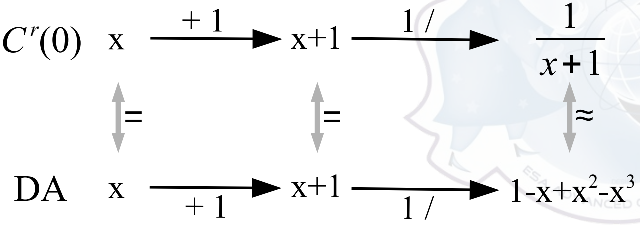
Functional Analysis: Example

In $C^3(0)$ and 3rd order DA, evaluate

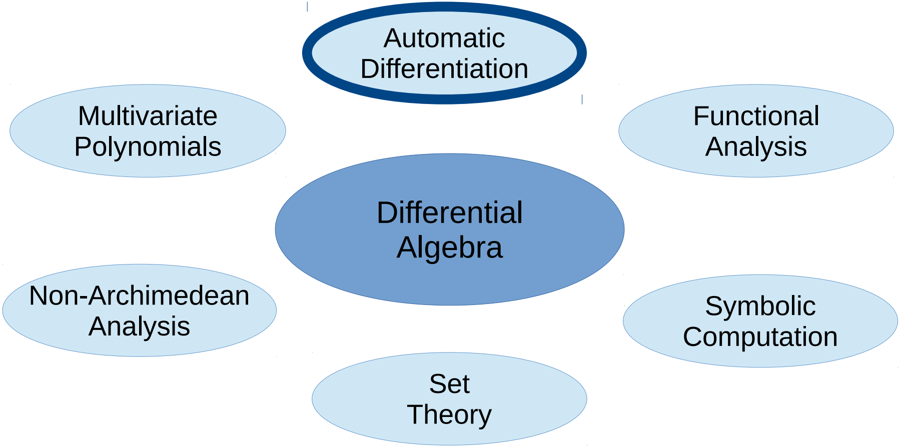
$$\frac{1}{x+1}.$$

List of operations:

- start with the identity $x = x$
- perform $+1$ operation
- perform $1/$ operation



Automatic Differentiation



Automatic Differentiation

Let $\vec{x}_0 \in \mathbb{R}^m$, $f : \mathbb{R}^m \to \mathbb{R}$ and compute the derivatives

$$\left.\frac{d^k f}{d\vec{x}^k}\right|_{\vec{x}_0}$$

for any given order k at the point $\vec{x}_0$.

Automatic differentiation:

- Concerned with accurate computation of arbitrary derivatives of a function f at given point x_0
- Algorithms that are much faster and more accurate than e.g. divided differences
- DA is a specific instance of a *forward differentiation* method

DA as Automatic Differentiation

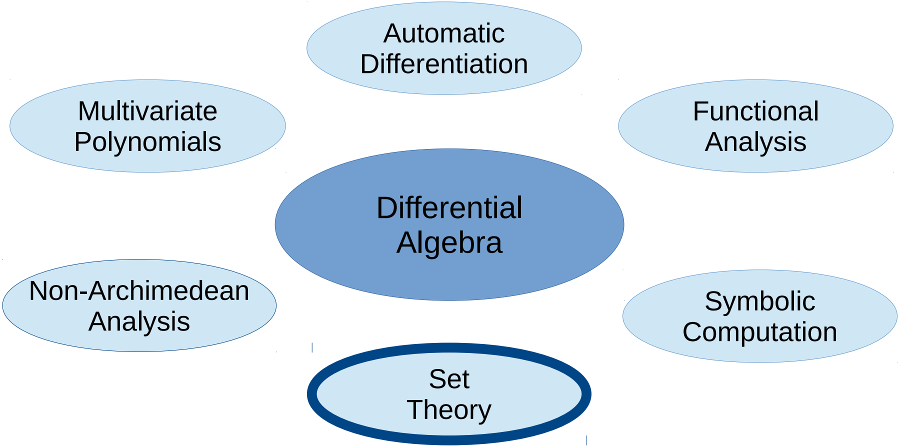
Given $\vec{x}_0 \in \mathbb{R}^m$, $f : \mathbb{R}^m \to \mathbb{R}$, compute

$$P(\vec{x}) = f(\vec{x}_0 + \vec{x})$$

to some order in DA arithmetic.

- Then $P(\vec{x})$ is Taylor Expansion of f around $\vec{x}_0$ by the way we defined each operation.
- Contains *exact* derivatives $\left.\frac{d^k f}{d\vec{x}^k}\right|_{\vec{x}_0}$ in coefficients (up to floating point error, typically $\sim 10^{-15}$).
- Coefficients can be extracted by repeated application of differentiation operator ∂_i followed by extraction of constant part.

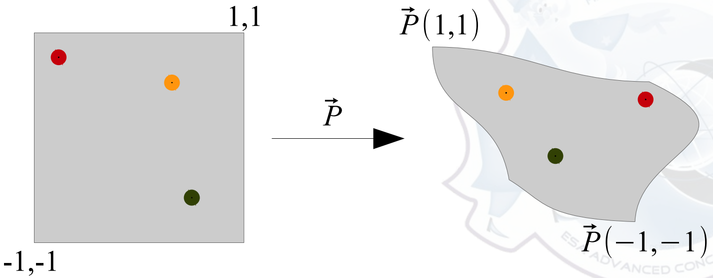
Sets and Manifolds



Sets and Manifolds

DA objects can be considered as representation of very general sets:

- Consider DA as a (structured) set by looking at image of domain $[-1, 1]^n$ under a polynomial map
- Can approximate very complicated sets very well
- Much better approximation of set valued functions than Interval Arithmetic



Sets and Manifolds

Set theoretical view of DA allows:

- easy representation of and computation on complicated sets
- fast propagation of sets of points (by one single function evaluation)
- accurate bounding of resulting sets

DA representation of sets has structure $\Rightarrow$ Manifolds

- Instead of one single map, consider many maps each covering a small part of the manifold ($\Rightarrow$ Domain Splitting)
- Natural computer representation of the mathematical concept of a manifold by representing the charts of the atlas as DA objects
- Calculations on a manifold straight forward

Taylor Models

- Extension of DA techniques to automatically compute rigorous bounds of truncation errors
  - DA:
    $f(\vec{x_0} + \vec{\delta x}) \approx f(\vec{x_0}) + f'(\vec{x_0}) \cdot \vec{\delta x} + \cdots + \frac{1}{n!} f^{(n)}(\vec{x_0}) \cdot \vec{\delta x}^{\,n}$
  - TM:
    $f(\vec{x_0} + \vec{\delta x}) \in f(\vec{x_0}) + f'(\vec{x_0}) \cdot \vec{\delta x} + \cdots + \frac{1}{n!} f^{(n)}(\vec{x_0}) \cdot \vec{\delta x}^{\,n} + [-\varepsilon, \varepsilon]$
- Combined with *polynomial bounders* provides highly accurate, rigorous bounds for range of f over given domains.

Applications in *verified numerics*:

- Global Optimization
- Global Fixed Point Finder
- Verified Integration
- Manifold Enclosures

$\Longrightarrow$ Computer Assisted Proofs

ODE flow expansion

Several methods to compute flow expansion $\varphi(\vec{x}_0, t)$:

- Arbitrary order time expansion by DA Picard iteration
- DA evaluation of classical numerical schemes
  - Runge Kutta (e.g. RK45, DP78)
  - Adams-Bashforth
- *Never:* variational equations!

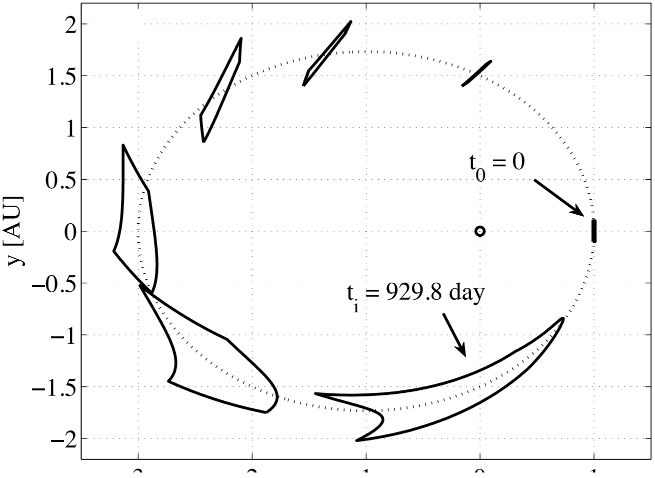
Result of each method:

$$P(\delta\vec{x}, \delta t) = \varphi(\vec{x}_0 + \delta\vec{x}, t + \delta t)$$

- First order of P corresponds to state transition matrix
- Extremely useful to propagate entire sets of initial conditions

ODE flow expansion

Propagation of set of initial conditions in Kepler dynamics (set view):



ODE flow expansion

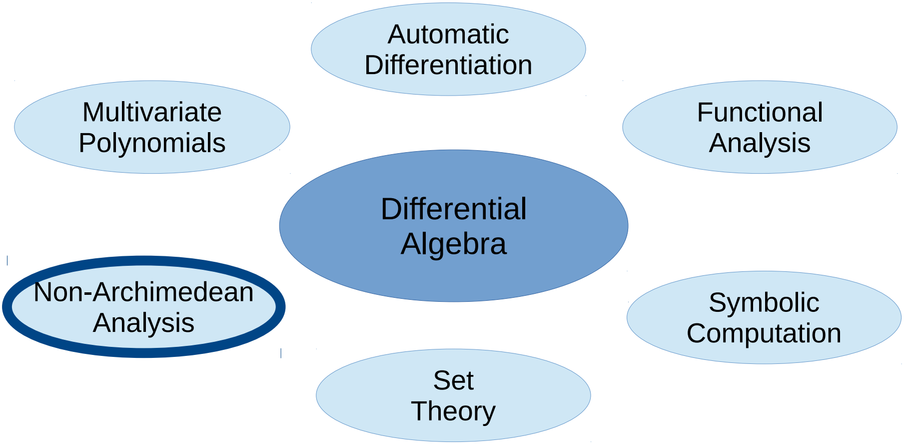
Advanced set propagation techniques:

Domain Splitting: nonlinear dynamics cause sets to grow. Automatically decompose polynomial into smaller polynomials covering subsets to ensure convergence.

Taylor integrator: arbitrary order integrator using the Taylor flow expansion. Instead of numerical scheme, use Taylor flow expansion and compute and evaluate at each time step.

Verified integration: Taylor integrator extended with verified Taylor Models. Computes verified enclosure of flow including truncation and round-off errors in each step. Yields verified enclosure of set (computer assisted proof).

Non-Archimedean Analysis



Non-Archimedean Analysis

Axiom of Archimedes

$$\forall \varepsilon > 0 \ \exists n \in \mathbb{N} \text{ s.t. } 1/n < \varepsilon$$

"every positive ε can be multiplied by some n such that $\varepsilon \cdot n$ is larger than 1"

Non-Archimedean Analysis: Drop axiom to allow *infinitesimals*
$\Rightarrow$ The Levi-Civita Field

- Rigorous mathematical treatment of algebra with infinitesimals
- Provides rigorous theoretical underpinning for DA
- Useful to develop fast implementations of the computation of basic DA algorithms
  - *contracting operators*: number of correct orders increases by 1
  - *super-convergent operators*: number of correct orders doubles

An Introduction to Differential Algebra

Alexander Wittig[1], P. Di Lizia, R. Armellin, et al.[2]

[1] *ESA Advanced Concepts Team (TEC-SF)*
[2] *Dinamica SRL, Milan*