Automatic Differentiation With Higher Infinitesimals

or: Computational Smooth Infinitesimal Analysis in Weil Algebra

CASC 2021
Hiromi ISHII (DeepFlow, Inc.)
Slides available at: https://bit.ly/casc-smooth

Repository: github.com/konn/smooth

Today's Topic in short:

To Implement computational
Higher Infinitesimal Analysis,
utilising the connection between
Automatic Differentiation
and
Nilpotent Infinitesimal Analysis,
applying Gröbner basis technique

So what?

- **Automatic Differentiation** is a collection of methods to efficiently compute higher derivatives of the given function exploiting Chain Rule.
- **Nilpotent Infinitesimal Analysis:** an analysis using nilpotent infinitesimal such that $d^2 = 0, d \neq 0$. (Analysis in the era of Newton or Gauß)
    - Although such an infinitesimal real contradicts classically, it can be justified with topos theory and used to develop differential geometry (cf: **Synthetic Differential Geometry** by Lawvere et al.).
    - Such infinitesimals can be algebraically formulated as **Weil algebras.**
- Our Result:Combining Gröbner basis and Automatic Differentiation, we propose an algorithm to achieve infinitesimal analysis with general, higher infinitesimals associated with Weil algebras.
- We first explore the exact connection b/w AD and Nilpotent Analysis.

Outline

1. A Quick Review of Automatic Differentiation (Forward-Mode)
2. Justification of AD in terms of C^∞-rings and Weil algebras
3. Computational Higher-Order Infinitesimal Analysis with Weil Algebras

Outline

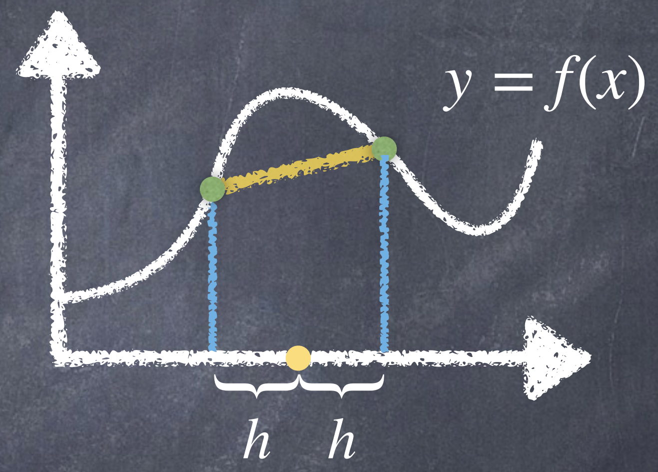
1. A Quick Review of Automatic Differentiation (Forward-Mode)
2. Justification of AD in terms of C^∞-rings and Weil algebras
3. Computational Higher-Order Infinitesimal Analysis with Weil Algebras

A Quick Review of Automatic Differentiation (Forward-Mode)

Three ways of differentiation

- Automatic Differentiation calculates differential coefficients efficiently and precisely, using Chain Rule.
- Numerical Differentiation replaces limits with small reals (e.g. Euler method)
  - Efficient, but only an approximation.
- Symbolic Differentiation expresses functions as symbolic ASTs and then differentiate them symbolically.
  - Exact, but easy to explode exponentially.

$$(f \circ g)'(x) = (f' \circ g)(x) \cdot g'(x)$$



$$f'(x) \approx \frac{f(x+h) - f(x-h)}{2h}$$

Automatic Differentiation (Forward mode)

A coefficient ring $\mathbb{R}$

value

diff. coeff.

```
data AD a = AD { real :: a, inifinitesimal :: a }

AD f f' + AD g g' = AD (f + g) (f' + g')
AD f f' * AD g g' = AD (f * g) (f' * g + f * g')
sin (AD f f') = AD (sin f) (f' * cos f)
```

Linearity

Leibniz Rule

Chain Rule

- Basic Idea: Storing the value and differential coefficient simultaneously.
  - We use a shorthand: $\mathtt{AD}\ x\ y = x + y\mathrm{d}$.
- For $F: \mathbb{R} \to \mathbb{R}$, its AD version $\hat{F} : \mathtt{AD}_{\mathbb{R}} \to \mathtt{AD}_{\mathbb{R}}$ returns $F(f(x)) + \dfrac{\mathrm{d}}{\mathrm{d}x} F(f(x))\mathrm{d}$, regarding inputs as of form $f(x) + f'(x)\mathrm{d}$ for some function f.
- The differential of F at $x = a$ will be the coefficient of d in $\hat{F}(a + \mathrm{d})$.
  - $a + d$ corresp. to the differential at $x = a$ of identity function $f(x) = x$.

DEMO

Why does this work?

We can justify with this
with the concepts of
C^∞-ring and Weil algebra!

Outline

1. A Quick Review of Automatic Differentiation (Forward-Mode)
2. Justification of AD in terms of C^∞-rings and Weil algebras
3. Computational Higher-Order Infinitesimal Analysis with Weil Algebras

Outline

1. A Quick Review of Automatic Differentiation (Forward-Mode)
2. Justification of AD in terms of C^∞-rings and Weil algebras
3. Computational Higher-Order Infinitesimal Analysis with Weil Algebras

2. Justification of AD in terms of -rings and Weil algebras

Justification

- Forward Mode AD corresponds to the dual number ring $\mathbb{R}[d] = \mathbb{R}[X]/(X^2)$, which has the canonical C^∞-ring structure
  - The law $d^2 = 0$ characterises the AD!

Def. (Lawvere)

A C^∞-ring is a commutative ring where the product-preserving map $A(f): A^n \to A$ is defined for any C^∞-function $f: \mathbb{R}^n \to \mathbb{R}$.

"Smooth multivariate function naturally lifts to A"

➡ More Rigorous Def

- Example: the ring $C^\infty(M)$ of real-valued smooth func. on a manifold M.
- Specifically, $\mathbb{R}[d]$ is an instance of Weil algebras, which is the speciall class of C^∞-rings with a nilpotent infinitesimal structure!

Power serieses form a C^∞-ring

Thm 2 (Lawvere)

The ring of multivariate formal power series $\mathbb{R}[\![\mathbb{X}]\!]$ has the structure of C^∞-ring via Taylor expansion.

Idea

For $f : \mathbb{R}^n \xrightarrow{C^\infty} \mathbb{R}$, $g_1, \ldots, g_n \in \mathbb{R}[\![\mathbb{X}]\!]$, we let:

$$\mathbb{R}[\![\mathbb{X}]\!](f)(\overrightarrow{g}) := \sum_{\boldsymbol{\alpha} \in \mathbb{N}^n} \frac{\boldsymbol{X}^{\boldsymbol{\alpha}}}{\alpha!} \mathrm{D}^{\boldsymbol{\alpha}}(f \circ \langle g_1, \ldots, g_n \rangle)(\mathbf{0}).$$

Weil algebra

Def.

$\mathbb{R}$-vector space W is called a Weil algebra if there exists an ideal $I \subseteq \mathbb{R}[X_1, \ldots, X_n]$ such that

$$W \cong \mathbb{R}[\mathbb{X}]/I, \text{ and } (X_1, \ldots, X_n)^k \subseteq I \text{ for some k.}$$

- Intuition: a real line augmented with nilpotent infinitesimals.
- Each variable X_i corresponds to an infinitesimal.
  - X_i is nilpotent in the quotient ring $\mathbb{R}[\mathbb{X}]/I$.
  - Specifically, it is finite dimensional as a vector-space; hence I is zero-dimensional ideal!

Weil ring is a C^∞-ring

Thm (Lawvere)

Any Weil algebra $W = \mathbb{R}[\mathbb{X}]/I$ has the canonical C^∞-structure.

This follows from the following lemma:

Lem 1

For any ring-theoretic ideal I on a C^∞-ring A, A/I has the canonical C^∞-structure induced by the quotient map:

$$(A/I)(f)([x_1]_I, \ldots, [x_m]_I) := \left[A(f)(x_1, \ldots, x_m)\right]_I$$

- Since I is zero-dimensional, $\mathbb{R}[[X]]/I \cong \mathbb{R}[X]/I$ is a C^∞-ring.
- In particular, dual number ring $\mathbb{R}[d] \cong \mathbb{R}[X]/(X^2)$ is also a C^∞-ring!

The C^∞-structure of $\mathbb{R}[d]$

- We let $f(a + b\mathrm{d}) = f(a) + bf'(a)\mathrm{d}$ for the univariate case.
- What about multivariate case $f : \mathbb{R}^n \xrightarrow{C^\infty} \mathbb{R}$?
  - This can be calculated from the C^∞ -structure of the univariate formal power series ring by cutting-off the terms at the degree 1 and replacing X with d.

The C^∞-structure of $\mathbb{R}[d]$

Letting $g_i = a_i + b_i X\ (i = 1,\ldots,n)$,

$$\begin{aligned}
&\mathbb{R}[\![X]\!](f)(\overrightarrow{g}) \\
&= f(\overrightarrow{g}(0)) + \frac{\mathrm{d}}{\mathrm{d}x}(f \circ \langle g_1, \ldots, g_n\rangle)(0)X + \ldots \\
&= f(a_1, \ldots, a_n) + (g_1'(0) + \ldots g_n'(0)) \cdot f'(g_1(0), \ldots, g_n(0))\ X + \ldots \\
&= f(a_1, \ldots, a_n) + (b_1 + \ldots + b_n) \cdot f'(a_1, \ldots, a_n)\ X + \ldots
\end{aligned}$$

Hence, by quotienting by $I = (X^2)$, we have

$$\mathbb{R}[d](f)(a_1 + b_1 d, \ldots, a_n + b_n d) = f(\overrightarrow{a}) + (b_1 + \ldots + b_n) f'(\overrightarrow{a}) \cdot d\,.$$

This coincides with the definition of AD when $n = 1$!

Summary

- (Forward Mode) Automatic Differentiation is a method to compute differentials of composite functions of smooth maps efficiently and precisely by storing the direct value and differential coefficient at some point simultaneously.
- This can be viewed as utilising the structure of C^∞-ring of the dual number ring $\mathbb{R}[d] = \mathbb{R}[X]/(X^2)$.
- The dual number ring is an instance of Weil algebra which axiomatises the real line with nilpotent infinitesimal.

Outline

1. A Quick Review of Automatic Differentiation (Forward-Mode)
2. Justification of AD in terms of C^∞-rings and Weil algebras
3. Computational Higher-Order Infinitesimal Analysis with Weil Algebras

Outline

1. A Quick Review of Automatic Differentiation (Forward-Mode)
2. Justification of AD in terms of C^∞-rings and Weil algebras
3. Computational Higher-Order Infinitesimal Analysis with Weil Algebras

3. Computational Higher-Order Infinitesimal Analysis with Weil Algebras

3. Computational Higher-Order Infinitesimal Analysis with Weil Algebras

Why Weil algebras?

Q. Why we have referred to Weil algebra and C^∞-rings, instead of just saying "Forward-mode AD can be justified by Taylor expansion of smooth maps"?

A. To generalise Forward Mode AD to multivariate / higher-order differentials and higher-order nilpotent infinitesimals.

Higher-order diff. with Weil algebras

- A tensor of the dual number rings can be used to compute higher derivatives. Let:

$$\mathbb{R}[d_1, \ldots, d_n] = \mathbb{R}[X_1, \ldots, X_n]/(X_i^2 \ \mid\ i \leq n) \cong \underbrace{\mathbb{R}[d] \otimes_{\mathbb{R}} \ldots \otimes_{\mathbb{R}} \mathbb{R}[d]}_{n\text{-copies}}$$

- Then, by an easy induction, we have:

$$\mathbb{R}[\overrightarrow{d}](f)(x + d_1 + \ldots + d_n) = \sum_{0 \leq i \leq n} f^{(i)}(x)\sigma_n^i(\overrightarrow{d}),$$

  where σ_k^i is the k-variate elementary symmetric expr. of degree i.

- In particular, we can get k-th derivative of f by looking at the coefficient of $d_1 \ldots d_k$ of $\mathbb{R}[\overrightarrow{d}](f)(x + d_1 + \ldots + d_n)$!
  - It has all the information of derivatives up to nth
- Likewise, higher derivatives of multivariate functions can be computed by a finite tensor products of the dual number rings.

DEMO

Our Contribution

Further Generalisation

- The structure of a Weil algebra $W = \mathbb{R}[\mathbb{X}]/I$ is determined by an ideal I with $(X_1, \ldots, X_n) \subseteq \sqrt{I}$.
- W is a finite-dimensional vector space; i.e, I is zero-dimensional.
  - We can employ Gröbner basis algorithms to compute their $\mathbb{R}$-algebra structure.
- For C^∞-structures of infinitesimal, we can first calculate in the formal power series ring, cutoff at the appropriate degree and take the quotient map.
  - In this way, we can do the calculation in the space with general nilpotent infinitesimals on computers!

Weil algebra Test

Algorithm 1 (WeilTest)

- Input: a polynomial ideal $I \subseteq \mathbb{R}[\mathbf{X}]$
- Is $I \subseteq \mathbb{R}[\mathbb{X}]$ a zerodimensional ideal? No → Not a Weil algebra
- Computer the Gröbner basis of $\sqrt{I}$ with Gröbner algorithms
- $X_1, \ldots, X_n \overset{?}{\in} \sqrt{I}$
    - No → Not a Weil algebra
    - Yes → Returns monomial basis, an upper bound of multidegree, and the multiplication table ([6])

- With these information, algebraic operations on Weil algebra W is easy.
- In our paper, we also returns the expression of nontrivial monomials as the linear combination of basis for the sake of efficiency.

C^∞-structure of Weil algebra

- Lem 1: the C^∞-structu.re of a Weil algebra is given by the quotienting that of $\mathbb{R}[\![X]\!]$.
- By Thm. 2, the C^∞-structure of $\mathbb{R}[\![X]\!]$ is given by:

$$\mathbb{R}[\![\mathbf{X}]\!](f)(g_1, \ldots, g_m) = \sum_{\alpha \in \mathbb{N}^n} \frac{\mathbf{X}^\alpha}{\alpha!} D^\alpha \left(f \circ \langle g_1, \ldots, g_m \rangle\right)(\mathbf{0}).$$

- We need an arbitrary partial derivatives of composite functions.
  - In general, AD calculating higher-order derivatives of multivariate functions is called Tower-mode AD.
  - Tower-mode AD and $\mathbb{R}[\![\mathbf{X}]\!]$ coincides modulo the factor of $\alpha! = \alpha_1! \cdot \cdots \cdot \alpha_m!$.
- Here, we think about C^∞-maps, derivatives are commutative.
  - We proposed, in [9], a variant of Tower AD optimised for this case (Appendix)

Computing C^∞-structure of Weil algebra

Algorithm 2 (LiftWeil)

- Input: $f\colon \mathbb{R}^n \xrightarrow{C^\infty} \mathbb{R}$ a smooth map admitting Tower AD,
  Information of Weil algebra $W = \mathbb{R}[X_1, \ldots, X_n]/I$ given by Alg. 1 (Especially, monomial basis $\{X^{\beta_1}, \ldots, X^{\beta_k}\}$ and multidegree upper bound α of W), and an element $\overrightarrow{w} = (w_1, \ldots, w_n) \in W^n$.
- Output: $W(f)(\overrightarrow{w}) \in W$: given by C^∞-structure of W.

1. Take the representative polynomials $\overrightarrow{g}$ of $\overrightarrow{w}$.
2. Calculate $h \in \mathbb{R}[X]$ given by cutting off $\mathbb{R}[\![\mathbf{X}]\!](f)(\overrightarrow{g})$ at the upper bound α (Using Tower AD and multiply by factorial).
3. Calculates $\bar{h}^G = \sum_{i \leq k} c_k X^{\beta_k}$ and return $(c_1, \ldots, c_k)$.

Application 1: More efficient higher derivatives

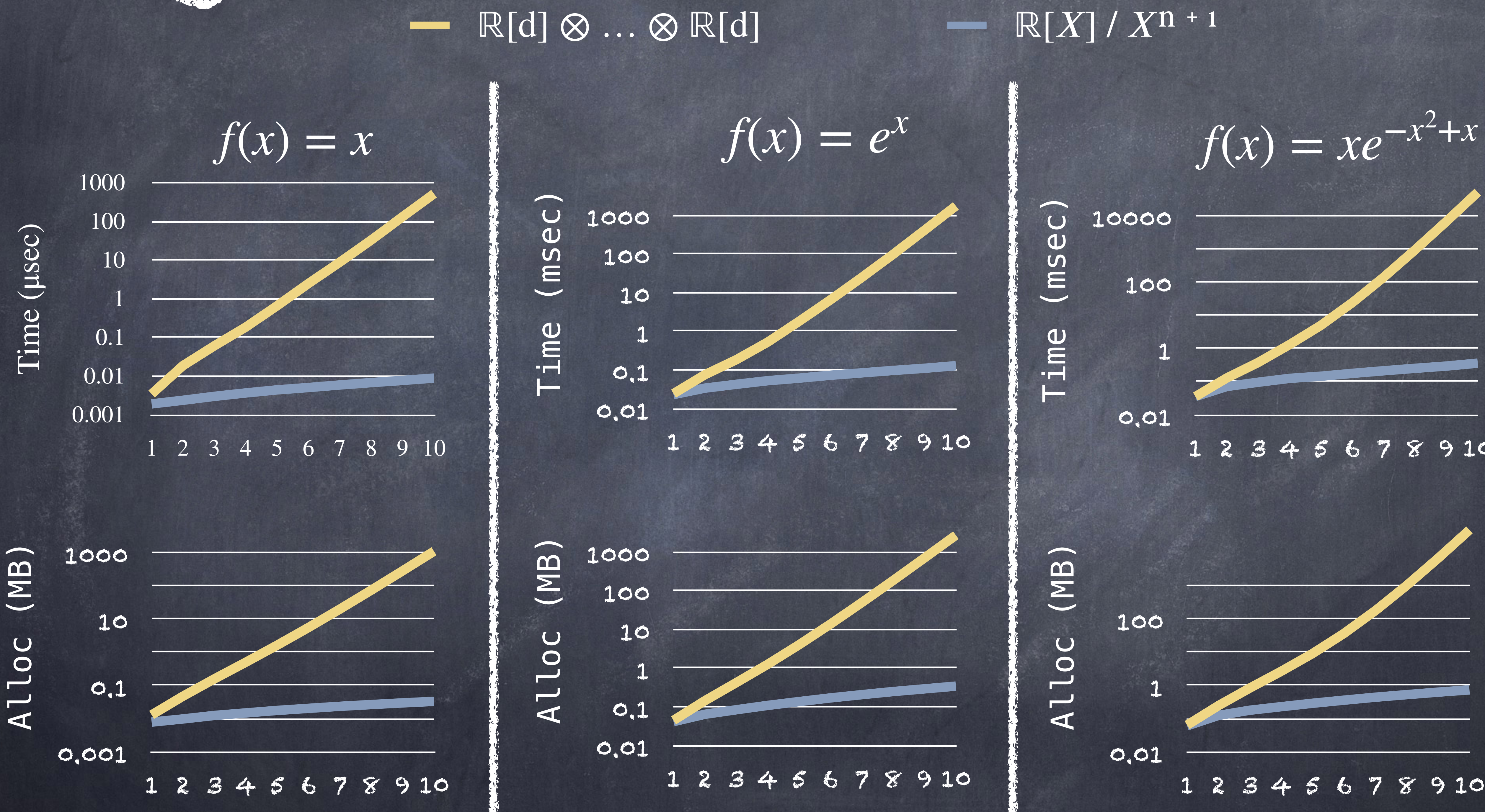
- "Tensor of dual numbers" approach needs n-many variables to calculate nth derivative.
  - The result much duplicate and redundant informations.
  - In particular, # of terms explodes to 2^n to calculate up to nth!
- We can save space and time using the following lemma:

Lem

For $\varepsilon = (X) \in \mathbb{R}[X]/X^n$ and smooth $f : \mathbb{R} \to \mathbb{R}$, we have:

$$\mathbb{R}[\varepsilon](f)(x + \varepsilon) = \sum_{0 \leq i < n} \frac{1}{i!} f^{(i)}(x) \varepsilon^i$$

Higher derivatives: bench



Environment: virtual Linux environment on GitHub Actions (Standard_DS2_v2 Azure instance) with two Intel Xeon Platinum 8171M virtual CPUs (2.60GHz) and 7 GiB of RAM.

Application 2: Tensors

- The tensor $W_1 \otimes_{\mathbb{R}} W_2$ of Weil algebras W_1, W_2 is again a Weil (Lem 4):

$$\mathbb{R}[\![X]\!]/I \otimes_{\mathbb{R}} \mathbb{R}[\![Y]\!]/J \cong \mathbb{R}[\![X, Y]\!]/\langle I, J \rangle$$

- The structure of tensors can be calculated with convolution (Alg. 4).
  - Using tensors we can combine arbitrary, multiple Weil algebras.
  - In particular, we can "package" higher-order ADs as Weil algebras, and "compose" them with tensors!

Ex.

Letting $\mathbb{R}[\varepsilon] = \mathbb{R}[x]/(x^{n+1}), \mathbb{R}[\delta] = \mathbb{R}[y]/(y^{m+1})$,

$$(\mathbb{R}[\varepsilon] \otimes_{\mathbb{R}} \mathbb{R}[\delta])(f)(x+\varepsilon, y+\delta) = \sum_{\substack{0 \leq i \leq n \\ 0 \leq j \leq m}} \frac{1}{i!j!} \partial x^i \partial y^j f(x, y).$$

DEMO

Summary

- Forward Mode Automatic Differentiation corresponds to the nilpotent infinitesimal analysis of second-order.
  - Justified by the theory of C^∞-rings by Lawvere et al.
  - In particular, Forward AD is an instance of Weil algebra which has nilpotent infinitesimals.
  - These concepts came from Synthetic Differential Geometry.
- Our Result: Combining AD and algorithms for zero-dimensional ideals, we achieved computational infinitesimal analysis with higher-order nilpotent infinitesimals on computers.
- Future Works: Optimisation, Synthetic treatment of Differential Geometry on Computers

Existing Works

- Nishimura and Osoekawa (2007) applies zero-dimensional algorithms to calculate the fundamental relation of generators of limit of Weil algebras.
    - They focus on the utility application to develop the theory of Weil algebra in SDG and not interested in computing differentials of smooth functions.
- Our interest is other direction - our central concern is computational infinitesimal analysis and smooth-ring computation.
    - Our contribution clarifies the connection of ADs and SIAs, which has been already pointed out but not discussed thoroughly, and implement it on the computer.

Future Works

- Develop Synthetic Differential Geometry (by Lawvere et al.) on computers using higher infinitesimals.
  - In SDG, general Weil algebras such as $\mathbb{R}[d_1, \ldots, d_n] = \mathbb{R}[x_1, \ldots, x_n]/(d_i d_j \mid i, j \leq n)$ are used to develop theory.
  - This opens a door to synthetic treatment of differential-geometric objects on computer.
    - Challenge: although $C^\infty(M)$ is finitely-presented as a C^∞-ring but, not so as an $\mathbb{R}$-algebra!
- Investigate the connection with other flavours of ADs, such as Reverse-Mode.

References

1. Moerdijk, I. and Reyes, G. E. *Models for Synthetic Differential Geometry*. Springer-Veriag New York, lnc.,1991.
2. Kmett, E., *The ad package*. `https://hackage.haskell.org/package/ad`
3. 稲葉一浩「自動微分 «フォワード・モード»」`http://www.kmonos.net/wlog/123.html#_2257111201`
4. Elliott. C, Beautiful Differentiation. ICFP 2009.
5. Cox, D., Little, J. and O'Shea, D., *Ideals, Varieties, and Algorithms*. 3rd ed. Springer Science+Business Media, 2007.
6. Cox, D., Little, J. and O'Shea, D., Using Algebraic Geometry. Springer Science+Business Media, 2005.
7. Nishimura, H. and Osoekawa, T., General Jacobi Identity Revisited Again. 2007.
8. I., *Automatic Differentiation With Higher Infinitesimals, or Computational Smooth Infinitesimal Analysis in Weil Algebra.* To appear in CASC 2021.
9. I., *A Succinct Multivariate Lazy Multivariate Tower AD for Weil Algebra Computation*. In: Fujimura, M. (ed.) Computer Algebra. RIMS Kôkyûroku, pp. 104–112. Research Institute for Mathematical Sciences, Kyoto University, Kyoto, Japan (2021). `arXiv:2103.11615`

Thank you!

Any Questions?

- Forward Mode Automatic Differentiation corresponds to the nilpotent infinitesimal analysis of second-order.
  - Justified by the theory of C^∞-rings by Lawvere et al.
  - In particular, Forward AD is an instance of Weil algebra which has nilpotent infinitesimals.
  - These concepts came from Synthetic Differential Geometry.
- Our Result: Combining AD and algorithms for zero-dimensional ideals, we achieved computational infinitesimal analysis with higher-order nilpotent infinitesimals on computers.
- Future Works: Optimisation, Synthetic treatment of Differential Geometry on Computers

Appendices

Categorically Rigorous

Definition of smooth rings

Def. (Lawvere)

Let $\mathbf{CartSp}$ be the category of all finite-dimensional Euclidean spaces and C^∞-maps between them.

A C^∞-ring is a product-preserving functor $A : \mathbf{CartSp} \to \mathbf{Sets}$.

Succinct Tower AD

$$
\begin{array}{ccc}
 & f(\boldsymbol{a}) & \\
f_x(\boldsymbol{a}) & f_y(\boldsymbol{a}) & f_z(\boldsymbol{a}) \\
f_{x^2}(\boldsymbol{a})\ f_{xy}(\boldsymbol{a})\ f_{xz}(\boldsymbol{a}) & f_{y^2}(\boldsymbol{a})\ f_{yz}(\boldsymbol{a}) & f_{z^2}(ba) \\
\vdots\ \ \vdots\ \ \vdots\ \vdots\ \ \vdots\ \ \vdots & \vdots\ \ \vdots\ \ \vdots & \vdots
\end{array}
$$

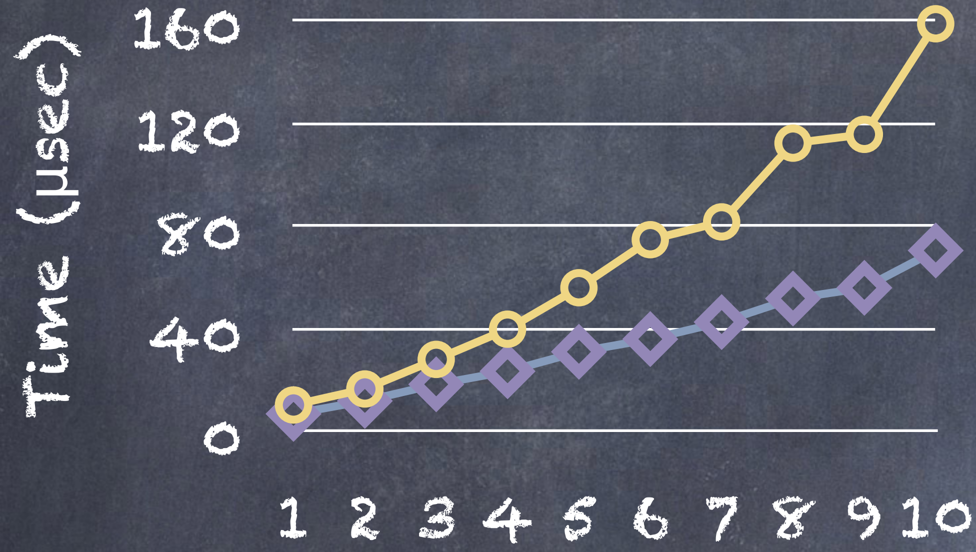
- Stores differentials as an infinite tree with decaying branching.
- Goes right if no further differentiation will be taken for preceding variables.

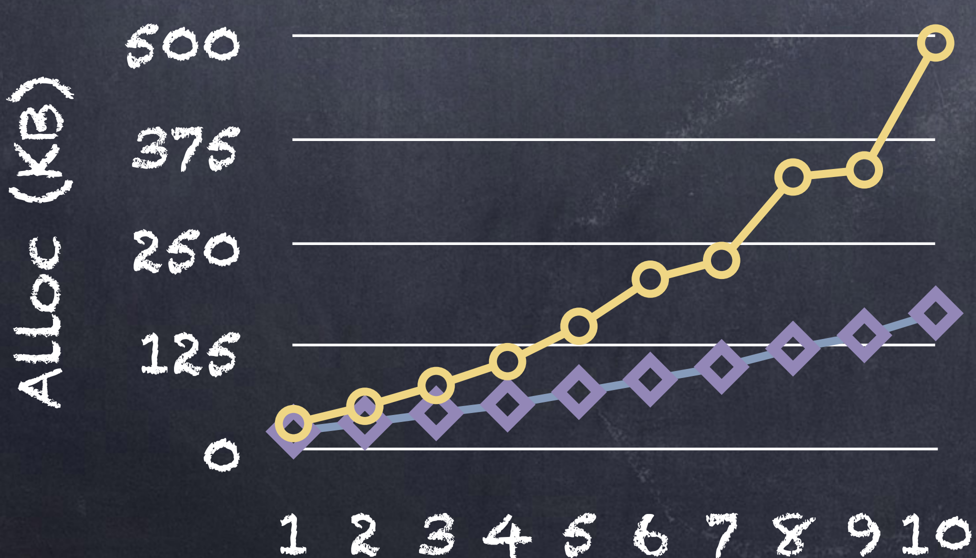
Tower bench: Univariate

Tower bench: Multivariate

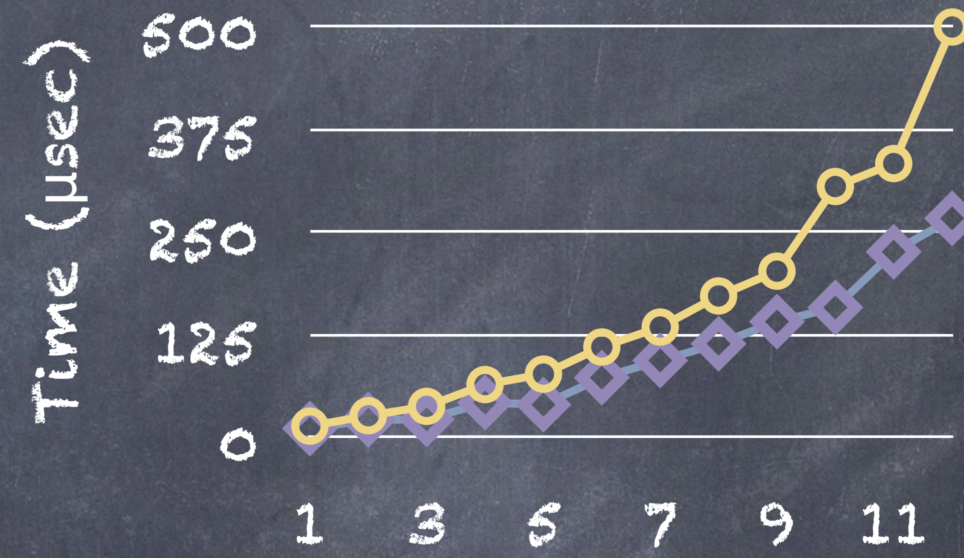
○ Sparse ◇ Our Method

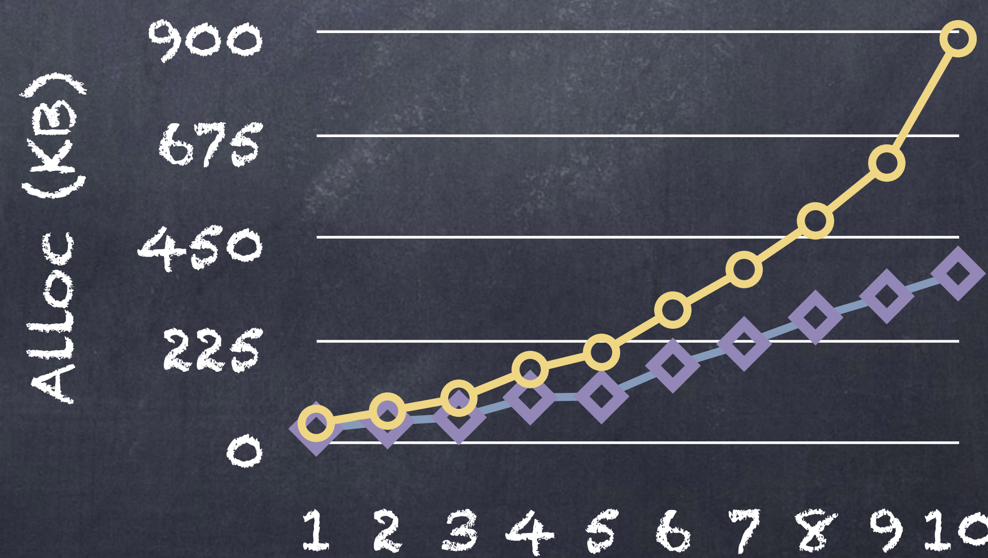
$$f(x, y) = \sin x \cdot e^{y^2}$$





$$f(x, y, z) = \sin x \cdot e^{y^2+z}$$

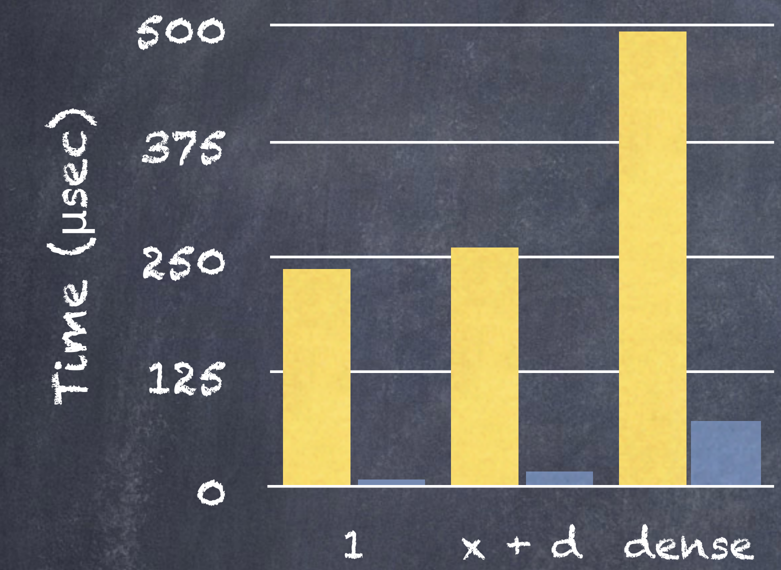


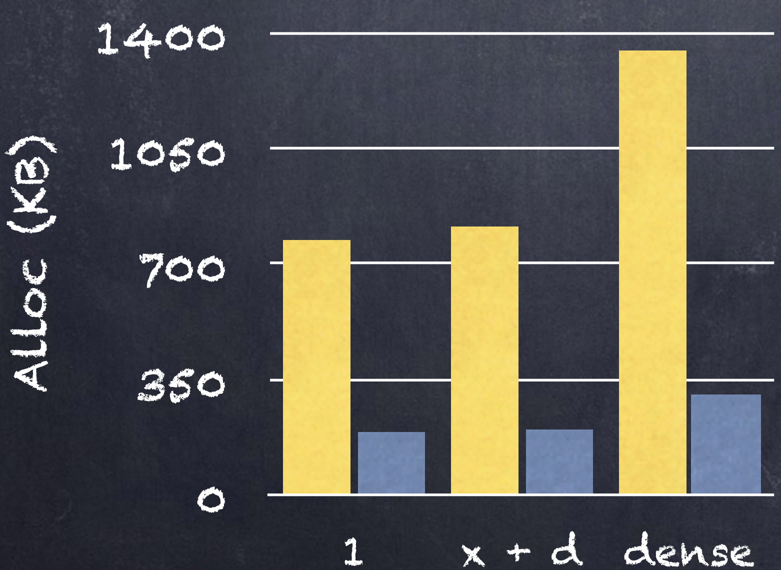


Tower bench: $\mathbb{R}[x, y]/(x^3 - y^2, y^3)$

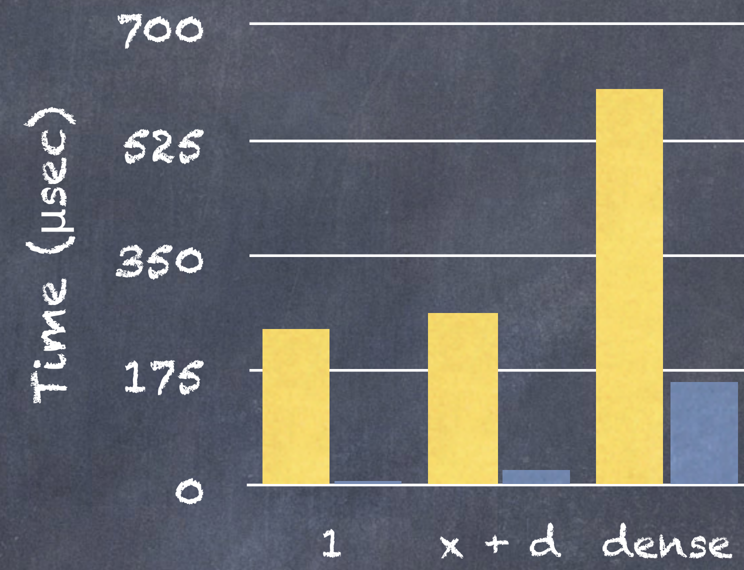
Sparse Our Method

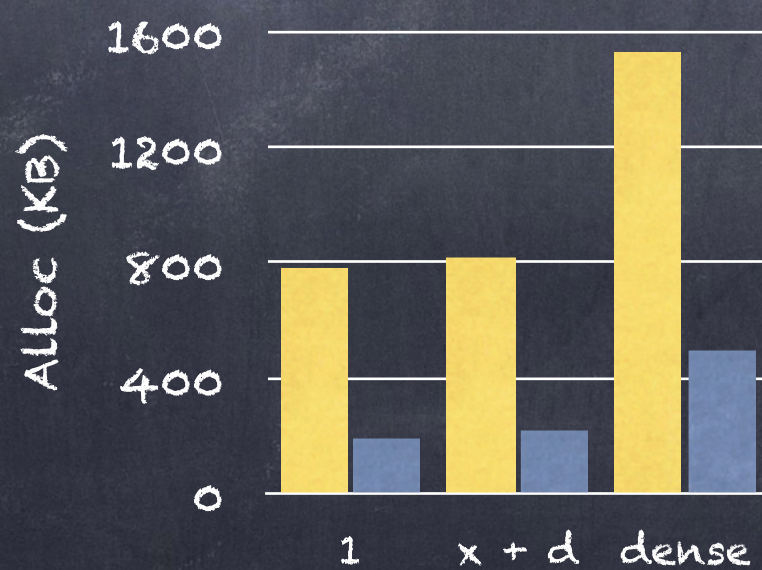
$f(x) = x$





$f(x) = e^x$





$f(x, y, z) = \sin x \cdot e^{x^2+x}$