

# Dual numbers

Miles Gould  
[@pozorvlak@mathstodon.xyz](https://mathstodon.xyz/@pozorvlak)

The screenshot shows a Microsoft Word document window. The title bar includes standard icons for file, edit, and view, along with a search field and a page count of '(1 of 2,147)'. The zoom level is set to '110%' with a minus and plus sign for adjustment. On the right side of the title bar are icons for file, print, and other document functions.

The left sidebar contains a 'Contents' tree view. It starts with a 'Preface' node, followed by two main sections: 'A Armv8-M Architecture Introduction and Overview' and 'B Armv8-M Architecture Rules'. The 'B' section is expanded, showing seven sub-sections: 'B1 Resets', 'B2 Power Management', 'B3 Programmers' Model', 'B4 Floating-point Support', 'B5 Vector Extension', 'B6 Pointer authentication and branch target identification Extension', and 'B7 Memory Model'.

The main content area displays the title 'Arm®v8-M Architecture Reference Manual' in a large, bold, black font.

The screenshot shows a PDF viewer interface with the following elements:

- Top Bar:** Includes icons for file operations (New, Open, Save, Print, Copy, Paste, Find, etc.), zoom controls (- | + 110%), and a search icon.
- Page Number:** A red circle highlights the page number "(1 of 2,147)" in the top center.
- Table of Contents:** On the left, a tree view of the manual's structure:
  - Contents
  - Preface
  - A Armv8-M Architecture Introduction and Overview
  - B Armv8-M Architecture Rules
    - B1 Resets
    - B2 Power Management
    - B3 Programmers' Model
    - B4 Floating-point Support
    - B5 Vector Extension
    - B6 Pointer authentication and branch target identification Extension
    - B7 Memory Model
- Title:** In the center, the title "Arm®v8-M Architecture Reference Manual" is displayed in a large, bold, black font.

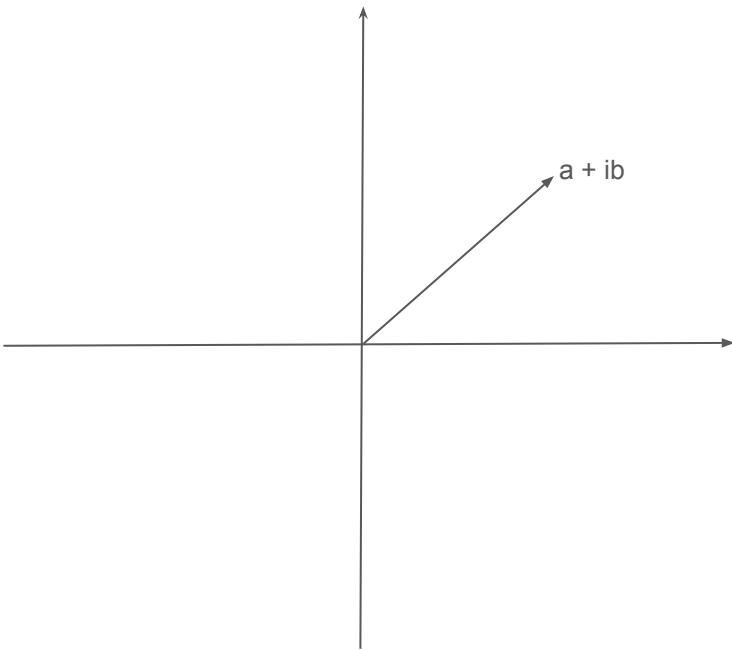
## **C2.4.381 VMLADAV Vector Multiply Add Dual Accumulate Across Vector.**

The elements of the vector registers are handled in pairs. In the base variant, corresponding elements from the two source registers are multiplied together, whereas the exchange variant swaps the values in each pair of values read from the first source register, before multiplying them with the values from the second source register. The results of the pairs of multiply operations are combined by adding them together. At the end of each beat these results are accumulated and the lower 32 bits written back to the general-purpose destination register. The initial value of the general-purpose destination register can optionally be added to the result.

## **C2.4.381 VMLADAV Vector Multiply Add Dual Accumulate Across Vector.**

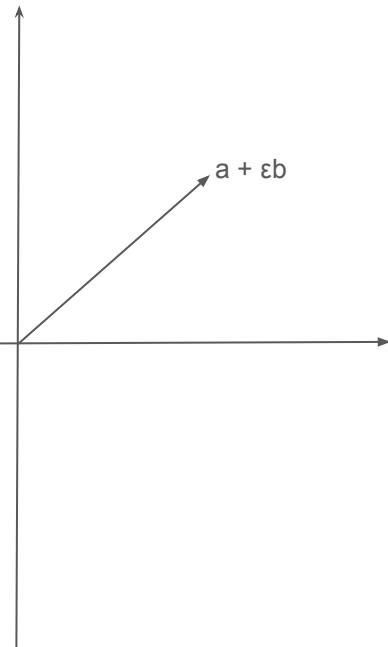
The elements of the vector registers are handled in pairs. In the base variant, corresponding elements from the two source registers are multiplied together, whereas the exchange variant swaps the values in each pair of values read from the first source register, before multiplying them with the values from the second source register. The results of the pairs of multiply operations are combined by adding them together. At the end of each beat these results are accumulated and the lower 32 bits written back to the general-purpose destination register. The initial value of the general-purpose destination register can optionally be added to the result.

# Complex numbers



- 2d vectors
- Write “ $a + ib$ ” for  $(a, b)$
- “Real” and “imaginary” parts
- $(a + ib) + (c + id) = (a + c) + i(b + d)$
- $i^2 = -1$
- $(a + ib)(c + id) = (ac - bd) + i(bc + ad)$

# Dual numbers



- 2d vectors
- Write “ $a + \varepsilon b$ ” for  $(a, b)$
- “Body” and “soul”
- $(a + \varepsilon b) + (c + \varepsilon d) = (a + c) + \varepsilon(b + d)$
- $\varepsilon^2 = 0$
- $(a + \varepsilon b)(c + \varepsilon d) = (ac + \cancel{bd}) + \varepsilon(bc + ad)$

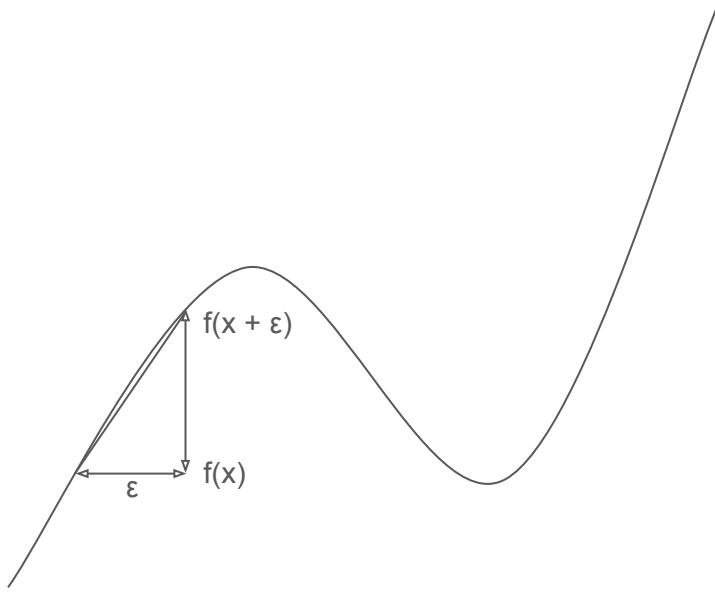
# You have seen these formulae before!

$$(a + \varepsilon b) + (c + \varepsilon d) = (a + c) + \varepsilon(b + d) \quad (f(x) + g(x))' = f'(x) + g'(x)$$

$$(a + \varepsilon b)(c + \varepsilon d) = ac + \varepsilon(bc + ad) \quad (f(x)g(x))' = f'(x)g(x) + f(x)g'(x)$$

# Automatic differentiation

$$f(x + \varepsilon) = f(x) + \varepsilon f'(x)$$



# Does it work?

$$\text{Let } f(x) = x$$

$$f(x + \varepsilon) = x + \varepsilon$$

# Does it work?

$$\text{Let } f(x) = x^2$$

$$f(x + \varepsilon) = (x + \varepsilon)^2$$

$$= x^2 + 2x\varepsilon + \varepsilon^2$$

$$= x^2 + 2x\varepsilon$$

# Does it work?

$$\text{Let } f(x) = x^3$$

$$f(x + \varepsilon) = (x + \varepsilon)^3$$

$$= x^3 + 3x^2\varepsilon + 3x\varepsilon^2 + \varepsilon^3$$

$$= x^3 + 3x^2\varepsilon$$

# Does it work?

$$\text{Let } f(x) = x^n$$

$$f(x + \varepsilon) = (x + \varepsilon)^n$$

$$= x^n + nx^{n-1}\varepsilon + \dots$$

$$= x^n + nx^{n-1}\varepsilon$$

# Yes!

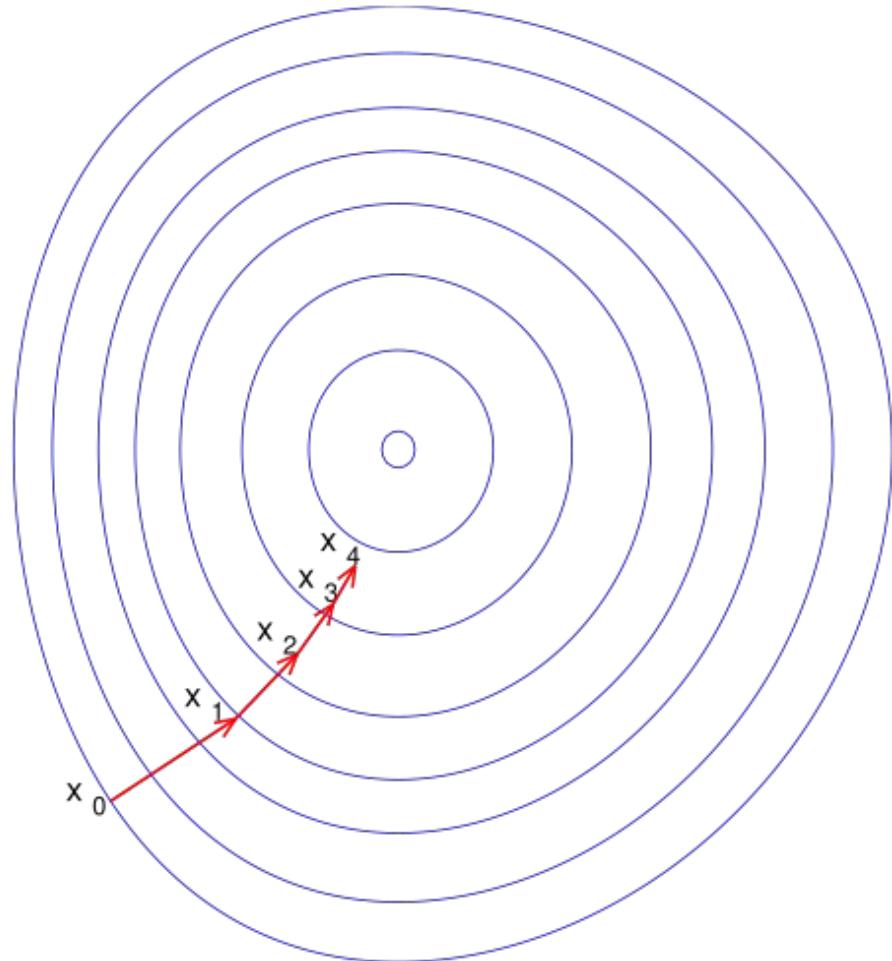
- Powers of  $n$  work
- Sums and products work
- So polynomial functions work
- So approximations to power series work
- But in fact, *arbitrary code* works
- We can extend this to higher or partial derivatives

# Why do we care?

Physics simulations!

Computer graphics!

Gradient descent!



# Historical note

- I first learned about these from Dan Piponi
- He used them in 2005 to do special effects for the *Matrix* sequels
- At the time they weren't widely-known in the graphics community
- But apparently Yoshua Bengio's deep-learning group already knew about them
  - “Source: rooftop beers” – Paul Khuong