

Using chatGPT

Thomas Kahle

The babelfish exists!



You

What are Quaternions?



ChatGPT

Quaternions are a number system that extends complex numbers. A quaternion is typically written as $q = a + bi + cj + dk$, where a, b, c , and d are real numbers, and i, j , and k are the fundamental quaternion units. Quaternions are used in computer graphics, robotics, and physics for efficiently handling calculations involving three-dimensional rotations and orientations.

Key properties include:

1. **Non-commutativity:** Unlike real or complex numbers, quaternion multiplication is not commutative. For example, $ij \neq ji$.
2. **Norm:** The norm of a quaternion q is defined as $\sqrt{a^2 + b^2 + c^2 + d^2}$. A quaternion is a unit quaternion if its norm is 1.
3. **Conjugate:** The conjugate of q is $q^* = a - bi - cj - dk$.
4. **Rotation Representation:** In 3D space, rotations can be represented using unit quaternions, providing a more compact and less computationally intensive alternative to rotation matrices.

Quaternion algebra is a fundamental tool in fields requiring three-dimensional geometry computations.

The babelfish exists!



You

What are Quaternions?



ChatGPT

Quaternions are a number system that extends complex numbers. A quaternion is

typically written as $q = a + bi + cj + dk$, where a, b, c

i, j , and k are the fundamental quaternion units. Quaternions are used in

graphics, robotics, and physics for efficiently handling 3D

dimensional rotations and orientations.

Key properties include:

1. **Non-commutativity:** Unlike real or complex numbers, quaternions are not commutative. For example, $ij \neq ji$.
2. **Norm:** The norm of a quaternion q is defined as $\sqrt{a^2 + b^2 + c^2 + d^2}$. A unit quaternion has a norm of 1.
3. **Conjugate:** The conjugate of q is $q^* = a - bi - cj - dk$.
4. **Rotation Representation:** In 3D space, rotations can be represented by quaternions, providing a more compact and less computationally intensive method than rotation matrices.

Quaternion algebra is a fundamental tool in fields requiring 3D geometry computations.



You

Now say it in Chinese

< 2 / 2 >



ChatGPT

四元数是一种扩展了复数的数系。一个四元数通常写作 $q = a + bi + cj + dk$ ，其中

a, b, c , 和 d 是实数， i, j , 和 k 是基本的四元数单位。四元数在计算机图形学、机器人

学和物理学中用于有效处理三维旋转和定向的计算。

关键属性包括：

1. **非交换性：** 与实数或复数不同，四元数乘法不是交换的。例如， $ij \neq ji$ 。
2. **范数：** 四元数 q 的范数定义为 $\sqrt{a^2 + b^2 + c^2 + d^2}$ 。如果四元数的范数为1，则为单位四元数。
3. **共轭：** 四元数 q 的共轭是 $q^* = a - bi - cj - dk$ 。
4. **旋转表示：** 在三维空间中，旋转可以使用单位四元数表示，提供了一种比旋转矩阵更紧凑、计算强度更低的替代方法。

四元数代数是三维几何计算领域的基础工具。

The babelfish exists!



You

What are Quaternions?



ChatGPT

Quaternions are a number system that extends complex numbers. A quaternion is

typically written as $q = a + bi + cj + dk$, where a, b, c, d are real numbers, i, j, k are the fundamental quaternion units. Quaternions are used in computer graphics, robotics, and physics for efficiently handling 3D dimensional rotations and orientations.

Key properties include:

1. **Non-commutativity:** Unlike real or complex numbers, quaternions are not commutative. For example, $ij \neq ji$.
2. **Norm:** The norm of a quaternion q is defined as $\sqrt{a^2 + b^2 + c^2 + d^2}$. A unit quaternion has a norm of 1.
3. **Conjugate:** The conjugate of q is $q^* = a - bi - cj - dk$.
4. **Rotation Representation:** In 3D space, rotations can be represented by quaternions, providing a more compact and less computationally intensive method than rotation matrices.

Quaternion algebra is a fundamental tool in fields requiring 3D geometry computations.



You

Now say it in Chinese

< 2 / 2 >



ChatGPT

四元数是一种扩展了复数的数系。一个四元数通常写作 $q = a + bi + cj + dk$ ，其中 a, b, c, d 是实数， i, j, k 是基本的四元数单位。四元数在计算机科学和物理学中用于有效处理三维旋转和定向的计算。

关键属性包括：

1. **非交换性：** 与实数或复数不同，四元数乘法不是交换的。例如， $ij \neq ji$ 。
2. **范数：** 四元数 q 的范数定义为 $\sqrt{a^2 + b^2 + c^2 + d^2}$ 。如果四元数的范数为 1，则称为单位四元数。
3. **共轭：** 四元数 q 的共轭是 $q^* = a - bi - cj - dk$ 。
4. **旋转表示：** 在三维空间中，旋转可以使用单位四元数表示，提供了一种紧凑、计算强度更低的替代方法。

四元数代数是三维几何计算领域的基础工具。

The babelfish exists!



You

Now say it in python



ChatGPT

Finished analyzing

python

Copy code

Explaining quaternions in Python code format

class Quaternion:

def __init__(self, a, b, c, d):

self.a = a # Real part

self.b = b # i component

self.c = c # j component

self.d = d # k component

def __repr__(self):

return f"{self.a} + {self.b}i + {self.c}j + {self.d}k"

def norm(self):

return (self.a**2 + self.b**2 + self.c**2 + self.d**2)**0.5

def conjugate(self):

return Quaternion(self.a, -self.b, -self.c, -self.d)

def multiply(self, other):

a1, b1, c1, d1 = self.a, self.b, self.c, self.d

a2, b2, c2, d2 = other.a, other.b, other.c, other.d

return Quaternion(a1*a2 - b1*b2 - c1*c2 - d1*d2,

a1*b2 + b1*a2 + c1*d2 - d1*c2,

a1*c2 - b1*d2 + c1*a2 + d1*b2,

a1*d2 + b1*c2 - c1*b2 + d1*a2)

Something to English

Something to English



You

Translate to English:

```
X :: Ord a => [a] -> [a]
X [] = []
X [x] = [x]
X (x:y:xs) = if x > y then y : X (x:xs) else x : X (y:xs)
```

```
Y :: Ord a => [a] -> Bool
Y [] = True
Y [x] = True
Y (x:y:xs) = if x > y then False else Y (y:xs)
```

```
XMain :: Ord a => [a] -> [a]
XMain list = if Y list then list else XMain (X list)
```



ChatGPT

The Haskell functions you've provided are for a sorting algorithm. Here's the English translation of the code:

- Function ``X`` is a helper function that performs a single pass of a bubble sort:
 - If the list is empty or has only one element, it returns the list as is.
 - If the list has more than one element, it compares the first two elements (``x`` and ``y``). If ``x`` is greater than ``y``, it swaps them and continues the process with the rest of the list. Otherwise, it keeps ``x`` in place and continues the process with ``y`` and the rest of the list.
- Function ``Y`` checks if a list is sorted:
 - If the list is empty or has only one element, it returns ``True`` (a sorted list).
 - If the list has more than one element, it checks if the first element is greater than the second. If so, it returns ``False`` (not sorted). Otherwise, it continues checking the rest of the list.
- Function ``XMain`` is the main sorting function that repeatedly applies ``X`` until the list is sorted (as determined by ``Y``):

Idea to Picture



You

Can you make tikz code that visualizes a complex number?



ChatGPT

Certainly! To visualize a complex number using TikZ in LaTeX, you'll need to draw a 2D coordinate system and then plot the complex number as a point or vector in this system. A complex number $z = a + bi$ can be represented as a point (or vector) in the 2D plane with coordinates (a, b) .

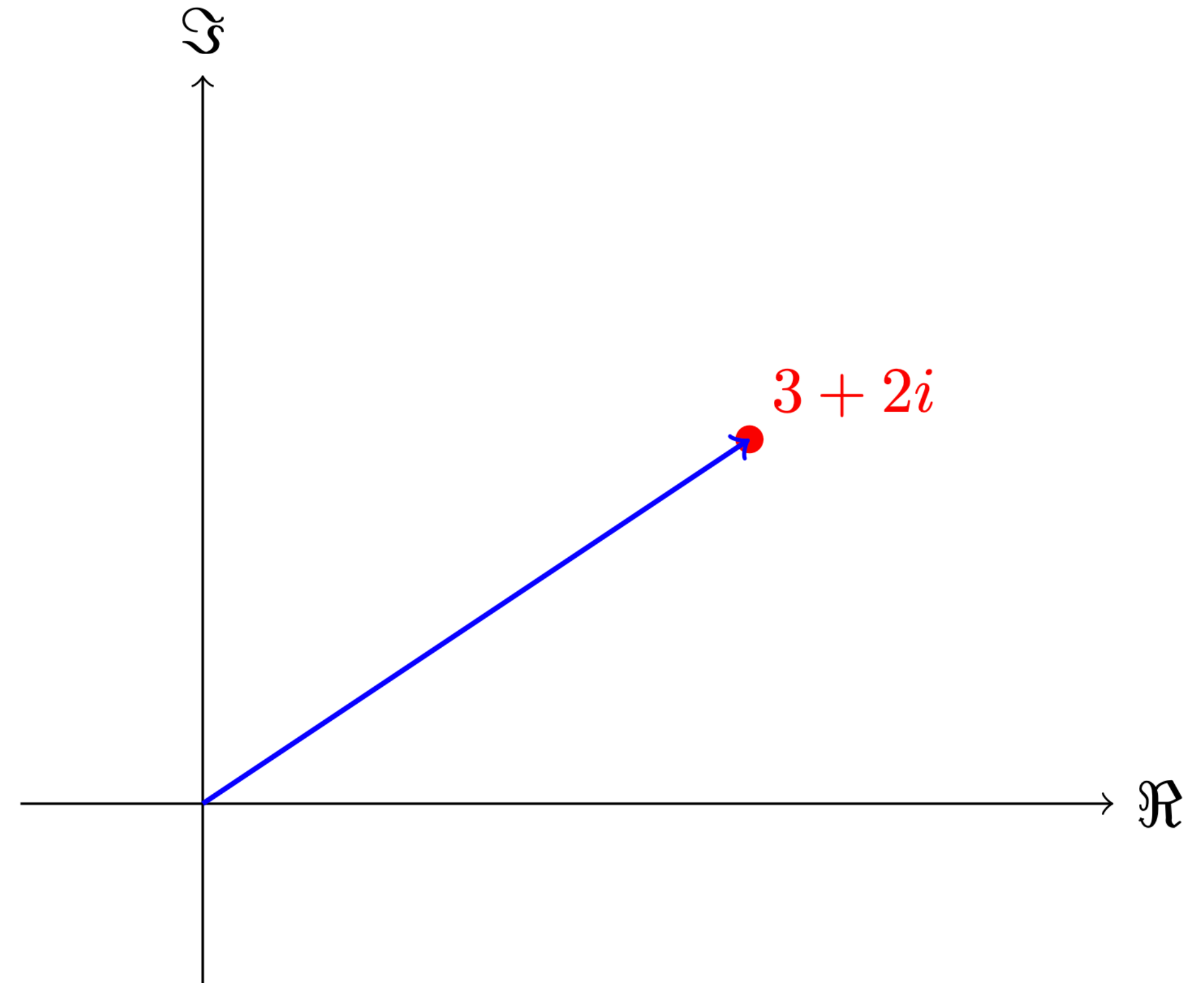
Here's a basic example of TikZ code to visualize the complex number $z = 3 + 2i$:

latex

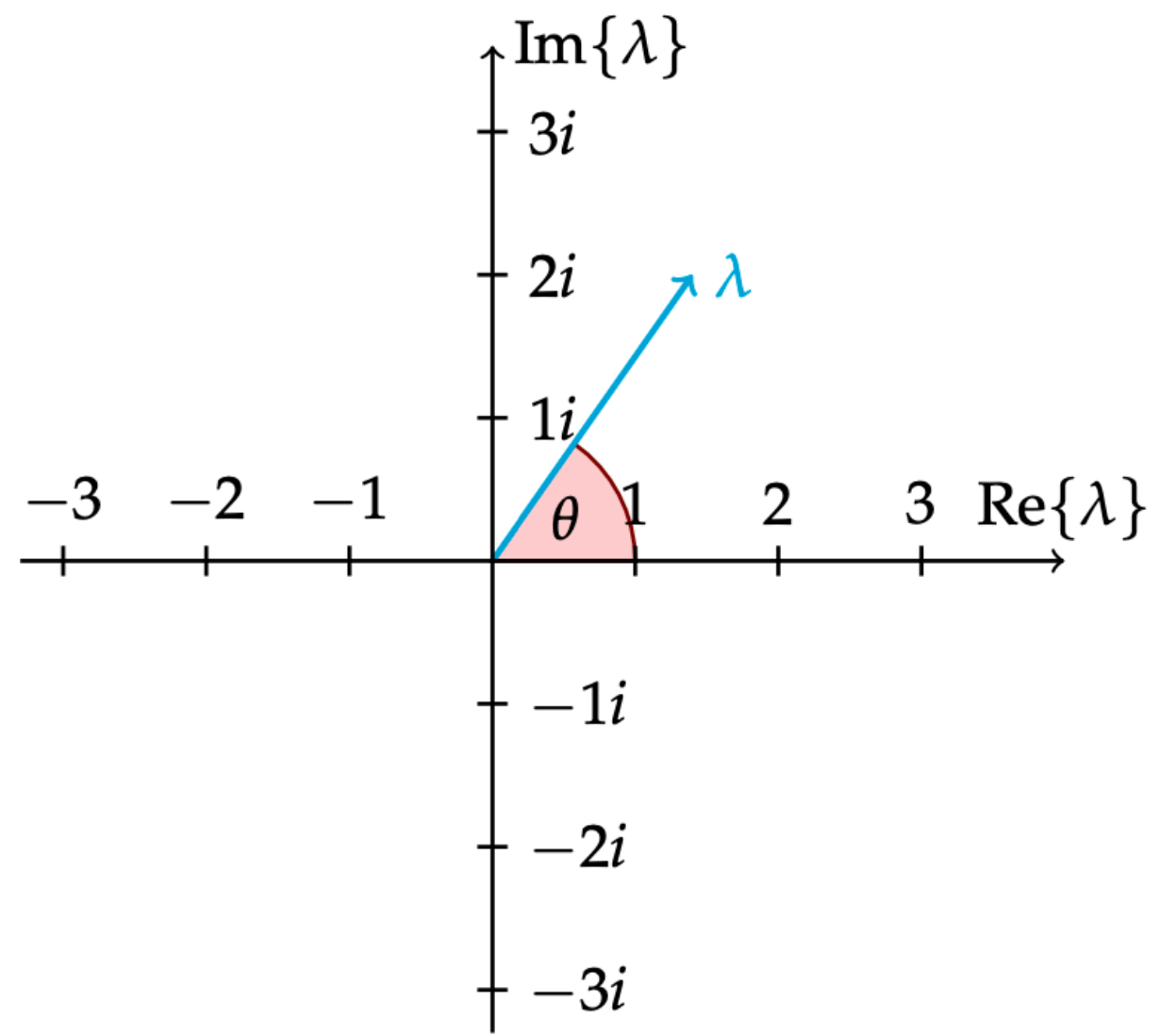
Copy code

```
\documentclass[tikz,border=10pt]{standalone}
\usepackage{tikz}
\usepackage{amsmath}

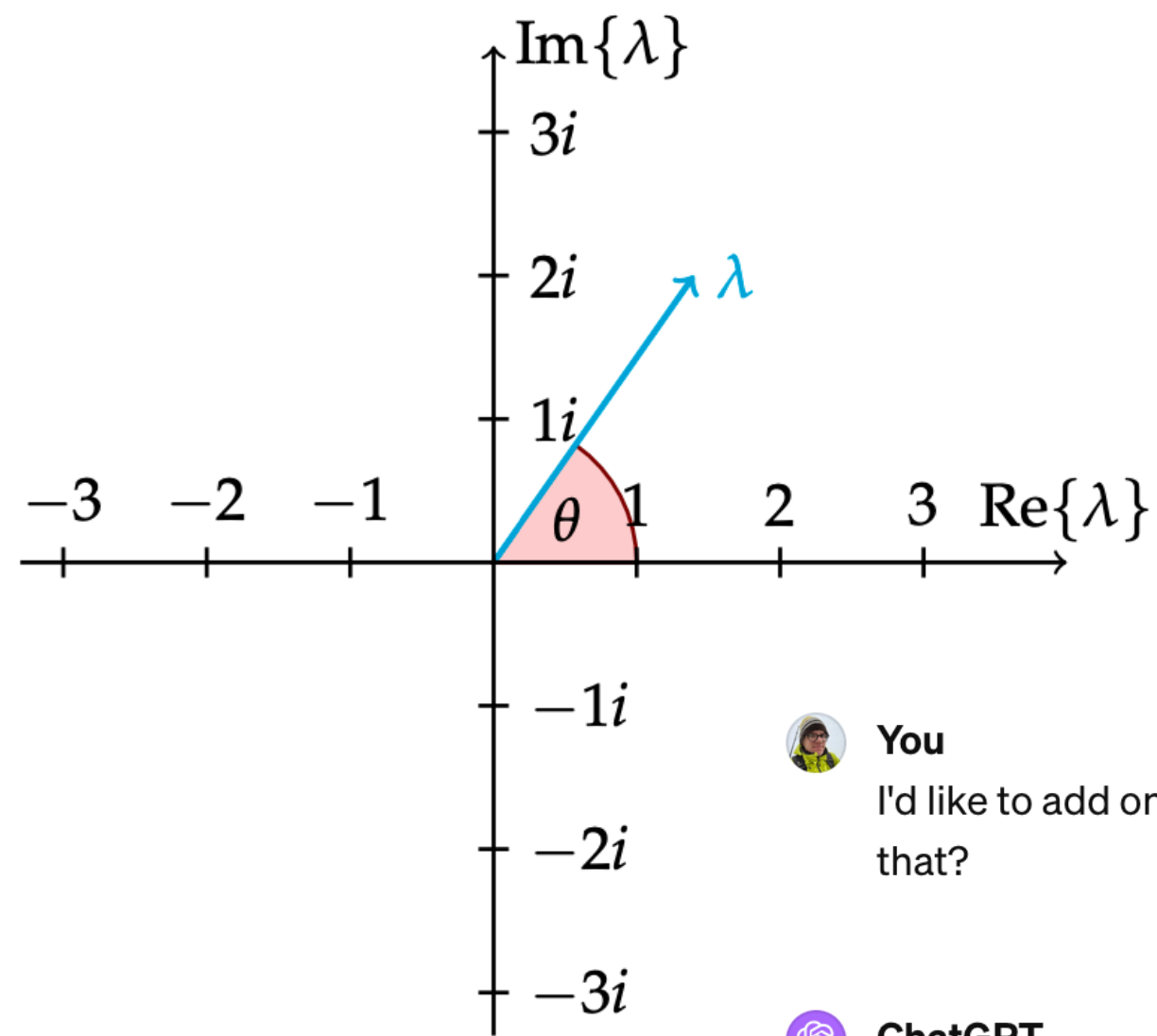
\begin{document}
```



Refining in conversation



Refining in conversation



You

I'd like to add one of these curly braces that signifies length of the vector. How do I do that?



ChatGPT

To add a curly brace in TikZ to signify the length of the vector, you can use the ``decorations.pathreplacing`` library and the ``decorate`` command. Here's how you can modify your existing TikZ code to include a curly brace that represents the length of the vector λ :

latex

Copy code

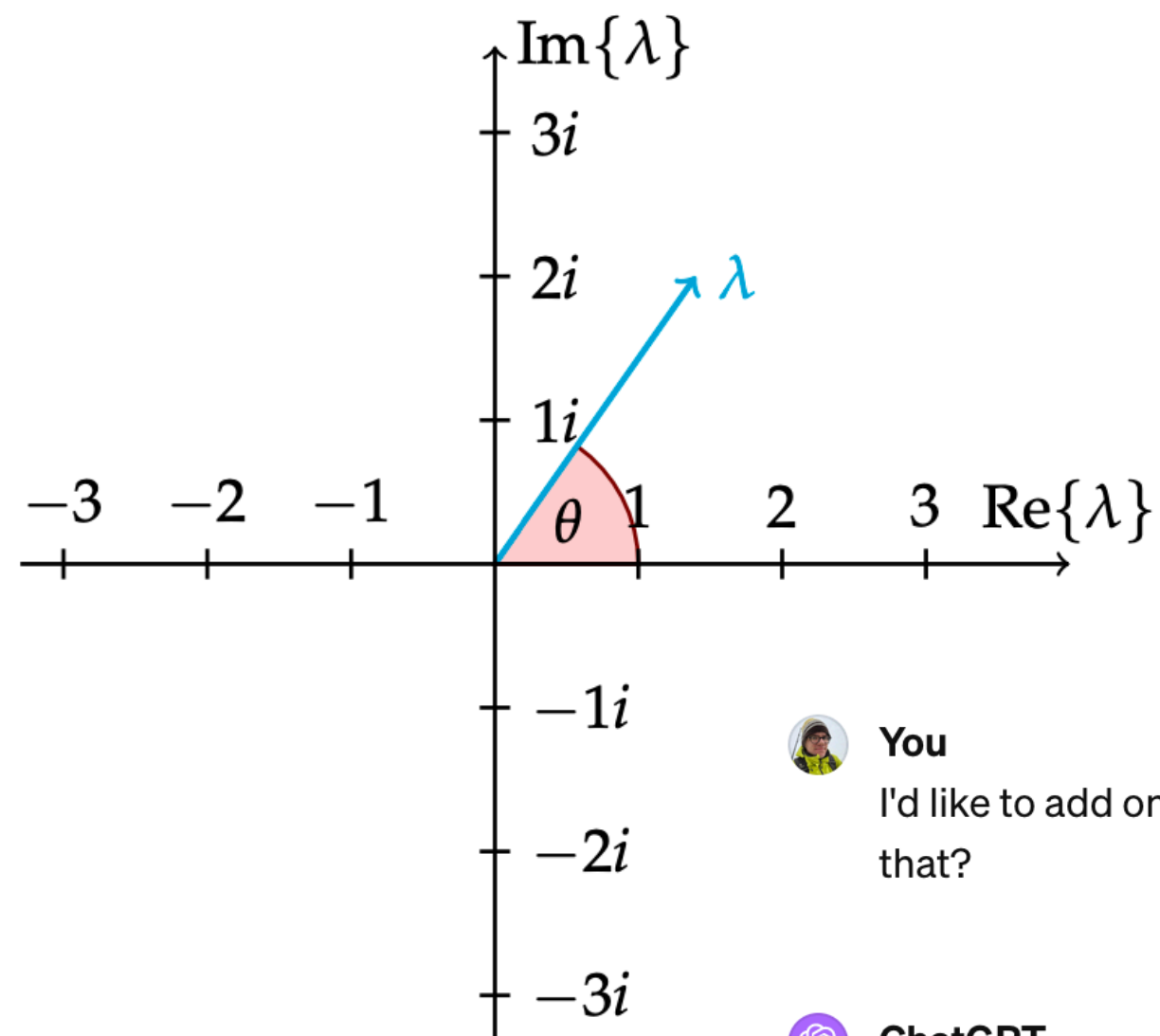
```
\begin{tikzpicture}
  \begin{scope}[thick,font=\normalsize]
    % ... [your existing code] ...

    % Vector Representation of lambda:
    \draw[->, very thick, color=cyan] (0,0) -- (1.4,2) node [right]

    % Curly brace to indicate the length of vector lambda
    \draw[decorate, decoration={brace, amplitude=10pt, mirror}] (0
      node[midway, xshift=2em, yshift=1em] {\lambda};

    % ... [rest of your existing code] ...
  \end{scope}
\end{tikzpicture}
```

Refining in conversation



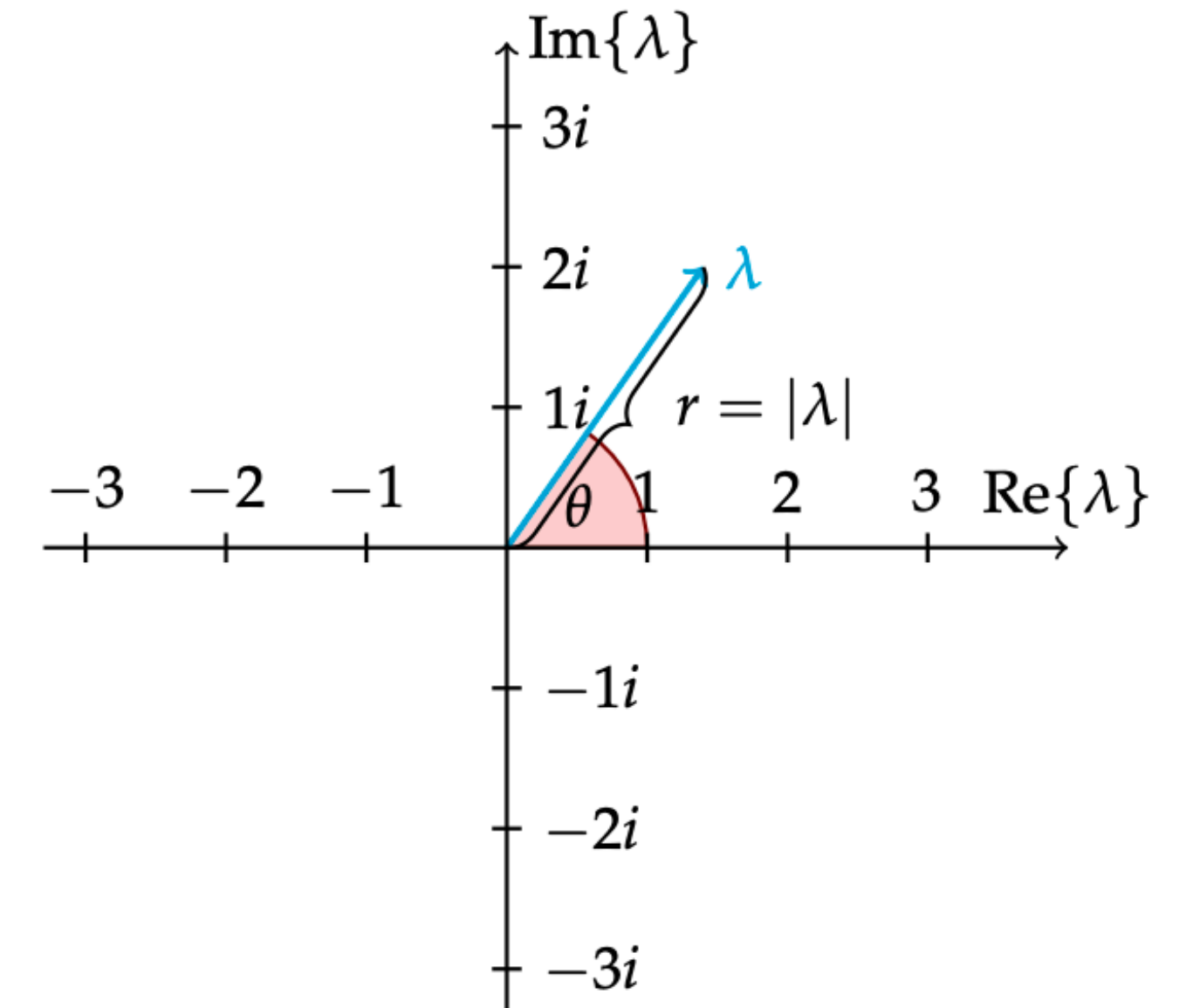
You

I'd like to add one of these curly braces that signifies length of the vector. How do I do that?



ChatGPT

To add a curly brace in TikZ to signify the length of the vector, you can use the `'decorations.pathreplacing'` library and the `'decorate'` command. Here's how you can modify your existing TikZ code to include a curly brace that represents the length of the vector λ :



latex

Copy code

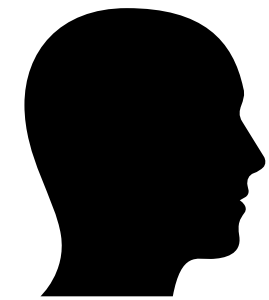
```
\begin{tikzpicture}
  \begin{scope}[thick,font=\normalsize]
    % ... [your existing code] ...

    % Vector Representation of  $\lambda$ :
    \draw[->, very thick, color=cyan] (0,0) -- (1.4,2) node [right]

    % Curly brace to indicate the length of vector  $\lambda$ 
    \draw[decorate, decoration={brace, amplitude=10pt, mirror}] (0
      node[midway, xshift=2em, yshift=1em] {\math|\lambda|};

    % ... [rest of your existing code] ...
  \end{scope}
\end{tikzpicture}
```

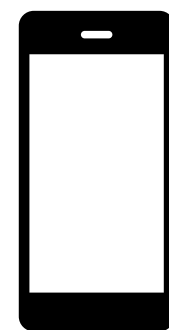
Finally Context !!



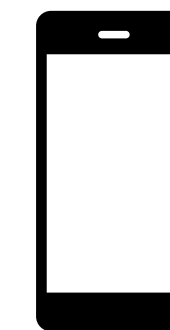
Hey Siri,
play blinding lights
by the Weeknd



Not that! The original
version please!



♪ Acoustic version ♪



I'm sorry, I've not
understood your query. Here
are some web search results
for "the original version":



ChatGPT

Hmm...something seems to have gone wrong.



ChatGPT

Hmm...something seems to have gone wrong.



You

Can you give an explicit example?



ChatGPT

Certainly! Let's consider a concrete example:

Take the rings \mathbb{Z} (the ring of integers) and \mathbb{Z}_6 (the ring of integers modulo 6). Both of these are rings with unity, where the unity in \mathbb{Z} is 1 and in \mathbb{Z}_6 it's $[1]_6$ (the equivalence class of 1 modulo 6).

Now, define a function $f : \mathbb{Z} \rightarrow \mathbb{Z}_6$ by $f(x) = [2x]_6$. This function is a ring homomorphism because it preserves addition and multiplication:

1. **Preservation of Addition:** For any $a, b \in \mathbb{Z}$,
$$f(a + b) = [2(a + b)]_6 = [2a + 2b]_6 = [2a]_6 + [2b]_6 = f(a) + f(b).$$
2. **Preservation of Multiplication:** For any $a, b \in \mathbb{Z}$,
$$f(a \cdot b) = [2(a \cdot b)]_6 = [2a \cdot 2b]_6 = [2a]_6 \cdot [2b]_6 = f(a) \cdot f(b).$$

However, this function is not a homomorphism of rings with unity because it does not map the unity of \mathbb{Z} to the unity of \mathbb{Z}_6 :

- The unity in \mathbb{Z} is 1, and $f(1) = [2 \cdot 1]_6 = [2]_6$, which is not the unity in \mathbb{Z}_6 (which would be $[1]_6$).

Thus, f is a ring homomorphism but not a homomorphism of rings with unity.





ChatGPT

Hmm...something seems to have gone wrong.



You

Can you give an explicit example?



ChatGPT

Certainly! Let's consider a concrete example:

Take the rings \mathbb{Z} (the ring of integers) and \mathbb{Z}_6 (the ring of integers modulo 6). Both of these are rings with unity, where the unity in \mathbb{Z} is 1 and in \mathbb{Z}_6 it's $[1]_6$ (the equivalence class of 1 modulo 6).

Now, define a function $f : \mathbb{Z} \rightarrow \mathbb{Z}_6$ by $f(x) = [2x]_6$. This function is a ring homomorphism because it preserves addition and multiplication:

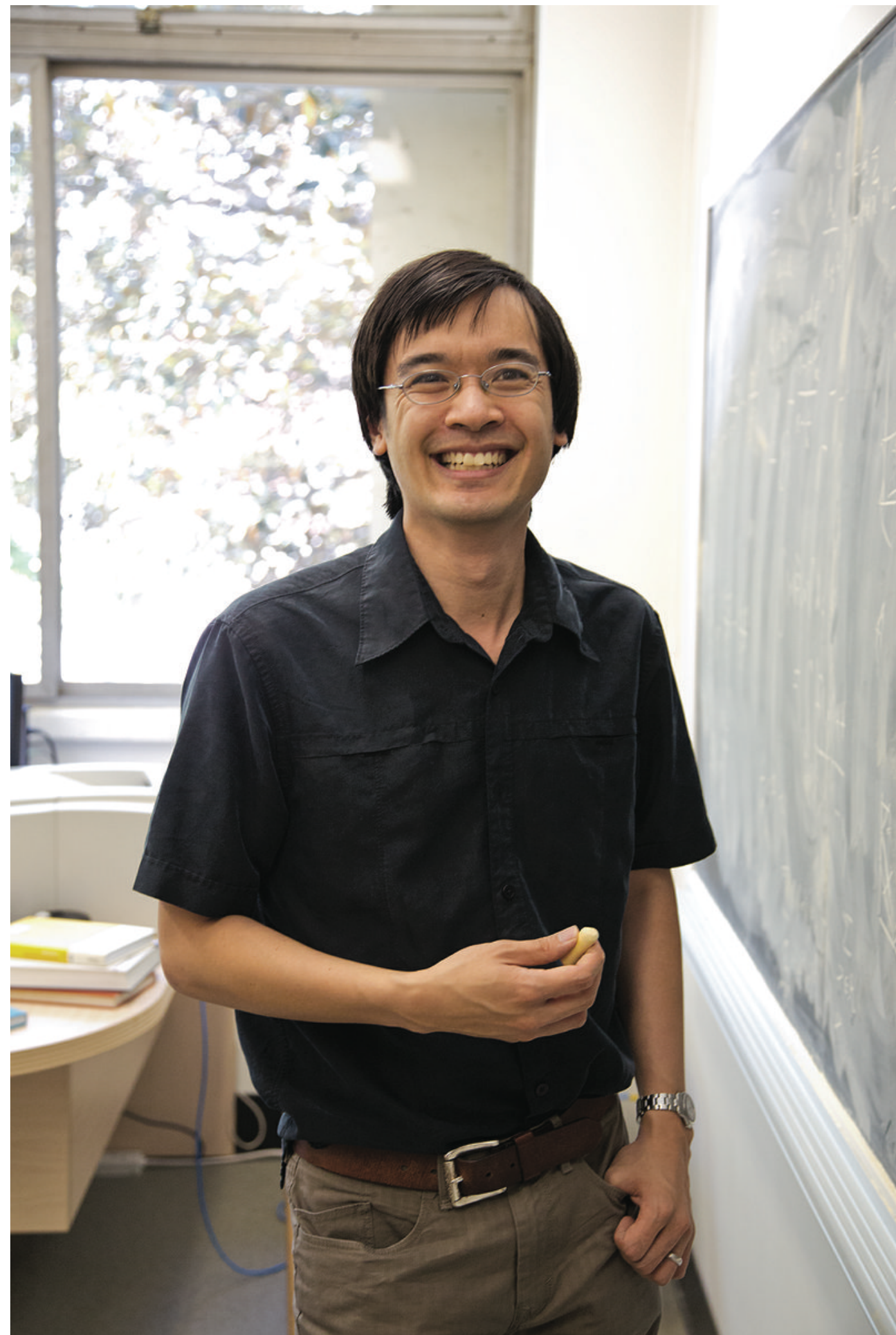
1. **Preservation of Addition:** For any $a, b \in \mathbb{Z}$,
$$f(a + b) = [2(a + b)]_6 = [2a + 2b]_6 = [2a]_6 + [2b]_6 = f(a) + f(b).$$
2. **Preservation of Multiplication:** For any $a, b \in \mathbb{Z}$,
$$f(a \cdot b) = [2(a \cdot b)]_6 = [2a \cdot 2b]_6 = [2a]_6 \cdot [2b]_6 = f(a) \cdot f(b).$$

However, this function is not a homomorphism of rings with unity because it does not map the unity of \mathbb{Z} to the unity of \mathbb{Z}_6 :

- The unity in \mathbb{Z} is 1, and $f(1) = [2 \cdot 1]_6 = [2]_6$, which is not the unity in \mathbb{Z}_6 (which would be $[1]_6$).

Thus, f is a ring homomorphism but not a homomorphism of rings with unity.





The stylistic signals that I traditionally rely on to “smell out” a hopelessly incorrect math argument are of little use with LLM-generated mathematics. Only line-by-line reading can discern if there is any substance.

Strangely, even nonsensical LLM-generated math often references relevant concepts. With effort, human experts can modify ideas that do not work as presented into a correct and original argument.

Talking points

- Assessments based on “producing language” need to be reconsidered
- Correlation between language quality and content quality is broken
- We need (to teach) proficiency in using LLMs
- We need (to teach) to deal with a world in which LLMs are abundant
- “Made with AI” stickers are pointless, or will be soon enough