

# Rethinking compilers with live coding

Tomas Petricek, Alan Turing Institute + fsharpWorks  
[@tomaspetricek](#) | [tomasp.net](#) | [fsharpworks.com](#)

29

NOT #INF OR THIS?

Welcome to the post-fact world

REGISTER TO VOTE ON THE EU REFERENDUM NOW!

WWW.  
WEAREEUROPE.  
ORG.UK



Can the result be reproduced?

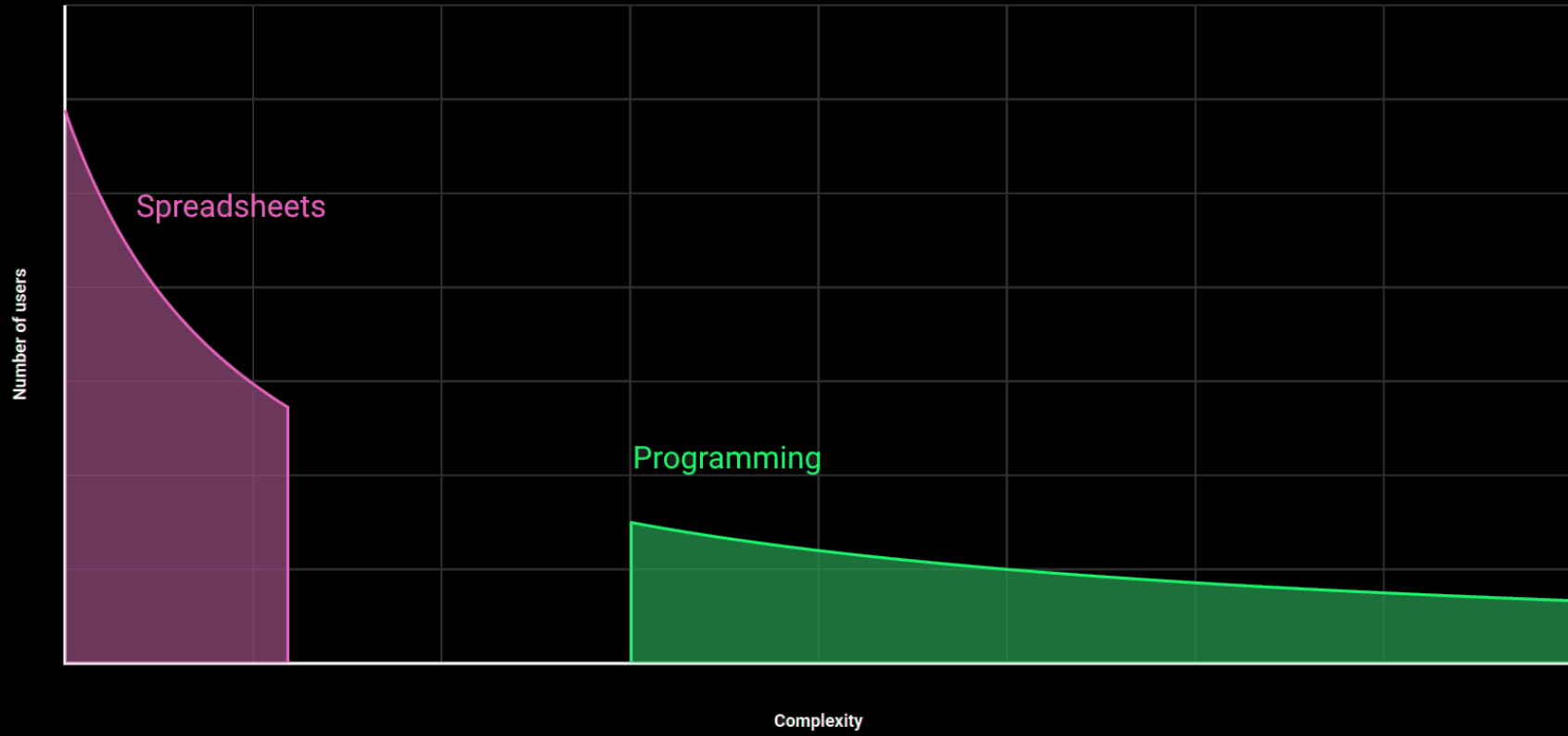
Is the visualization misleading?

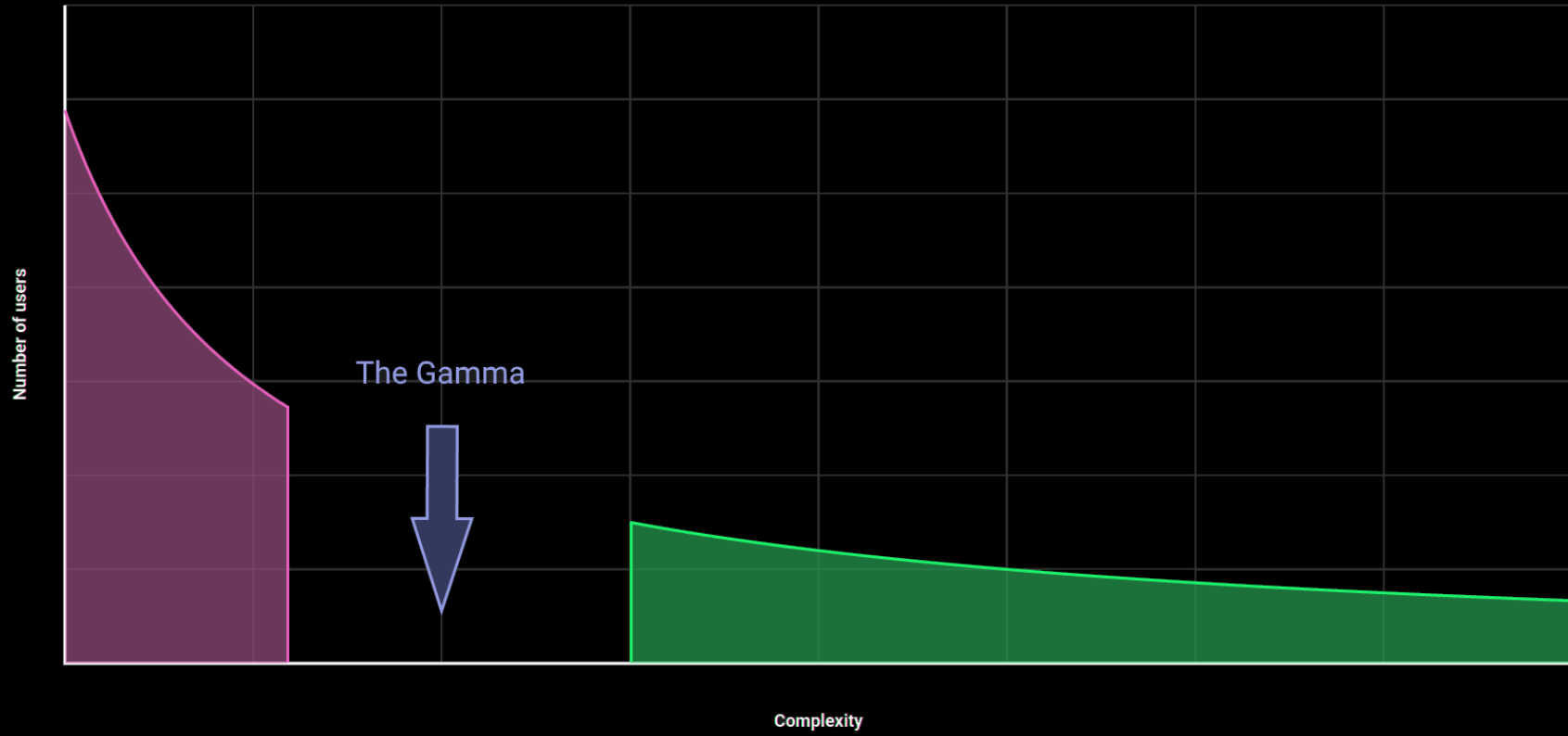
Can the reader explore further?



The Gamma

Making data science easier







Book1 - Excel Tomas Petricek

File Home Insert Page Layout Formulas Data Review View Add-ins Team Tell me Share

D4

	A	B	C	D	E	F	G	H	I
1									
2									
3		<b>Cost</b>	<b>Rate</b>	<b>Converted</b>					
4		149	0.63						
5		99	1.23						
6		14	0.63						
7		99	1						
8		249	1.23						
9									
10									
11									
12									
13									

Sheet1

Ready 120%

Book1 - Excel Tomas Petricek

File Home Insert Page Layout Formulas Data Review View Add-ins Team Tell me Share

D4

	A	B	C	D	E	F	G	H	I
1									
2									
3		<b>Cost</b>	<b>Rate</b>	<b>Converted</b>					
4		149	0.63	+					
5		99	1.23						
6		14	0.63						
7		99	1						
8		249	1.23						
9									
10									
11									
12									
13									

Sheet1

Ready 120%

DEMO

Live previews in TheGamma

# TEXTBOOK COMPILERS

Textbook compiler is batch based

Textbook compiler builds & transforms trees

Textbook compiler rejects bad programs

# Example - live image processing

```
let pope = image.load("pope.png")
let shadow = image.load("shadow.png")

shadow.greyscale().blur(5)
    .combine(pope, 50)
```

# SKETCH

Transforming abstract syntax trees

# DEMO

Writing a batch evaluator



# LIVE COMPILERS

Live compiler caches earlier results

Live compiler builds dependency graphs

Live compiler accepts bad programs

# SKETCH

Building a dependency graph

# DEMO

Writing a graph-based evaluator

# SUMMARY

Check out **TheGamma** project

And help make facts great again!

[www.thegamma.net](http://www.thegamma.net)

[gamma.turing.ac.uk](http://gamma.turing.ac.uk)

# F# language

[www.fsharp.org](http://www.fsharp.org)

Pragmatic functional-  
first programming

# Fable compiler

[www.fable.io](http://www.fable.io)

F# to JavaScript compiler  
for the 21st century

Dependency graphs over syntax trees

Reusing nodes enables caching

Good luck writing a decent parser :-)



Now you know how to write  
your own live coding environment!

Questions?

Tomas Petricek, Alan Turing Institute + fsharpWorks  
[@tomaspetricek](#) | [tomasp.net](#) | [fsharpworks.com](#)