Turing's

Universal

Digital

Computer

Why Universal

Predecessors

Cheapness

The universal machine has the advantage that one design can be applied to any problem. This brings huge economy of scale to manufacturing.

Self reference

Digital

Reality is digital, Turing computability rules.

Why Computer

Pilot Ace Console

Turings idea wa thus something much more fundamental and lasting than that of Church

Paul Cockshott

University of Glasgow

Greg Michaelson

HERIOT WATT UNIVERSITY

Turing's

Universal

Digital

Computer

rsal

Cheapness

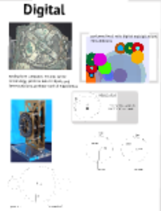
e universal machine has
e advantage that one
sign can be applied to any
oblem. This brings huge
onomy of scale to
anufacturing.

ity
ow

Why Universal

Predecessors

Digital



There were lots of computing machines before Turing put to practical use

Analogue

Cheapness

The universal machine has the advantage that one design can be applied to any problem. This brings huge economy of scale to manufacturing.

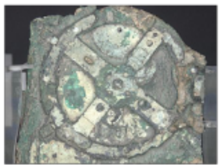
Self reference

Turing needed self reference
in order to address the decision problem.
But this property was the key to the economical adoption of computer technology. It gives Universality.
"This special property of digital computers, that they can mimic any discrete-state machine, is described by saying that they are universal machines."

Predecessors

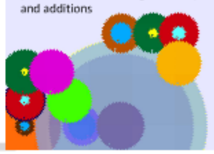
There were lots of computing machines before Turing put to practical use

Digital

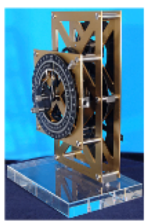


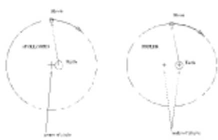
Antikythera computer, ancient Greek technology, predicts lunar eclipses and lunar positions, perhaps work of Appolonius

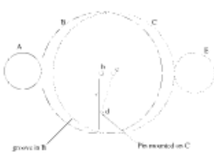
performs fixed ratio digital multiplications and additions



implements Appolonius eccentric orbital model



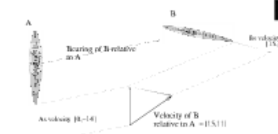




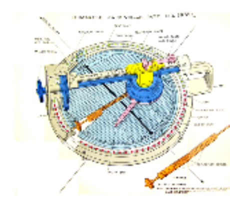
Analogue

100 years ago navies had to solve real-time vector arithmetic problems and regression problems that were way beyond what could be done manually.

Ranges were about 10 miles
relative velocities up to 60 mph
time of flight of the order 20secs



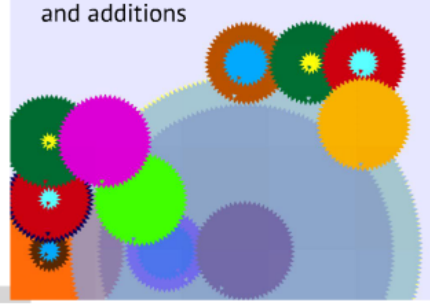
Rear Admiral John Dumaresq commander of Royal Australian Navy described by his colleagues in naval gunnery as a mathematical genius, and unsung pioneer of mechanical computing

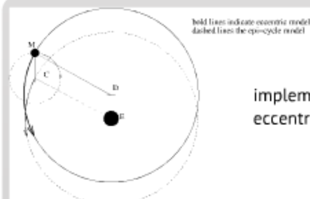


Digital

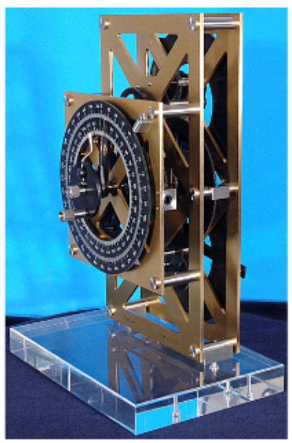
Antikythera computer, ancient Greek technology, predicts lunar eclipses and lunar positions, perhaps work of Appolonius

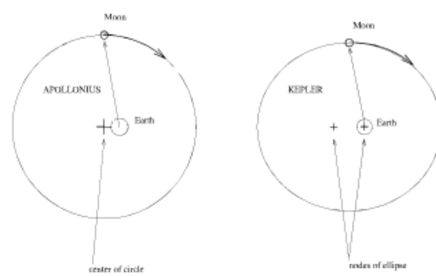
performs fixed ratio digital multiplications and additions

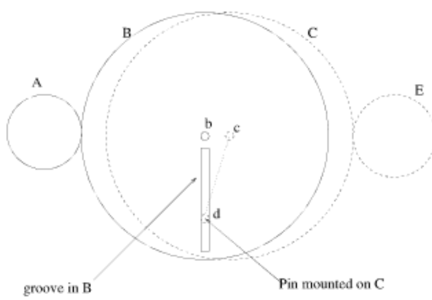




implements Appolonius eccentric orbital model



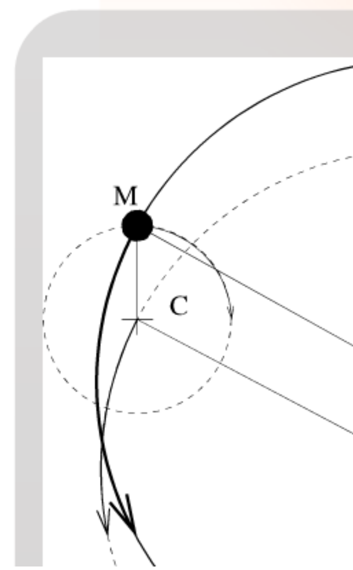




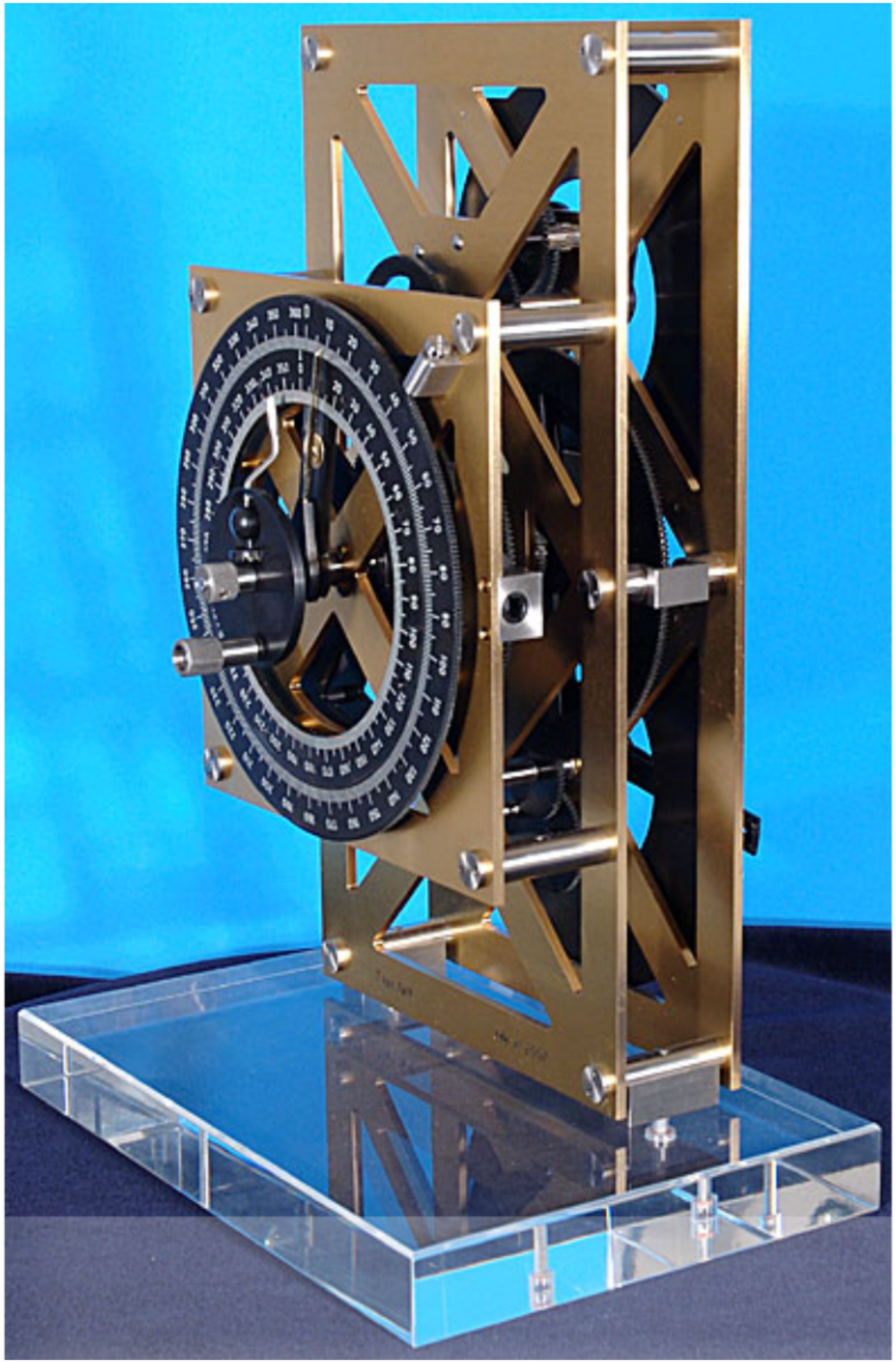
machin

put to p

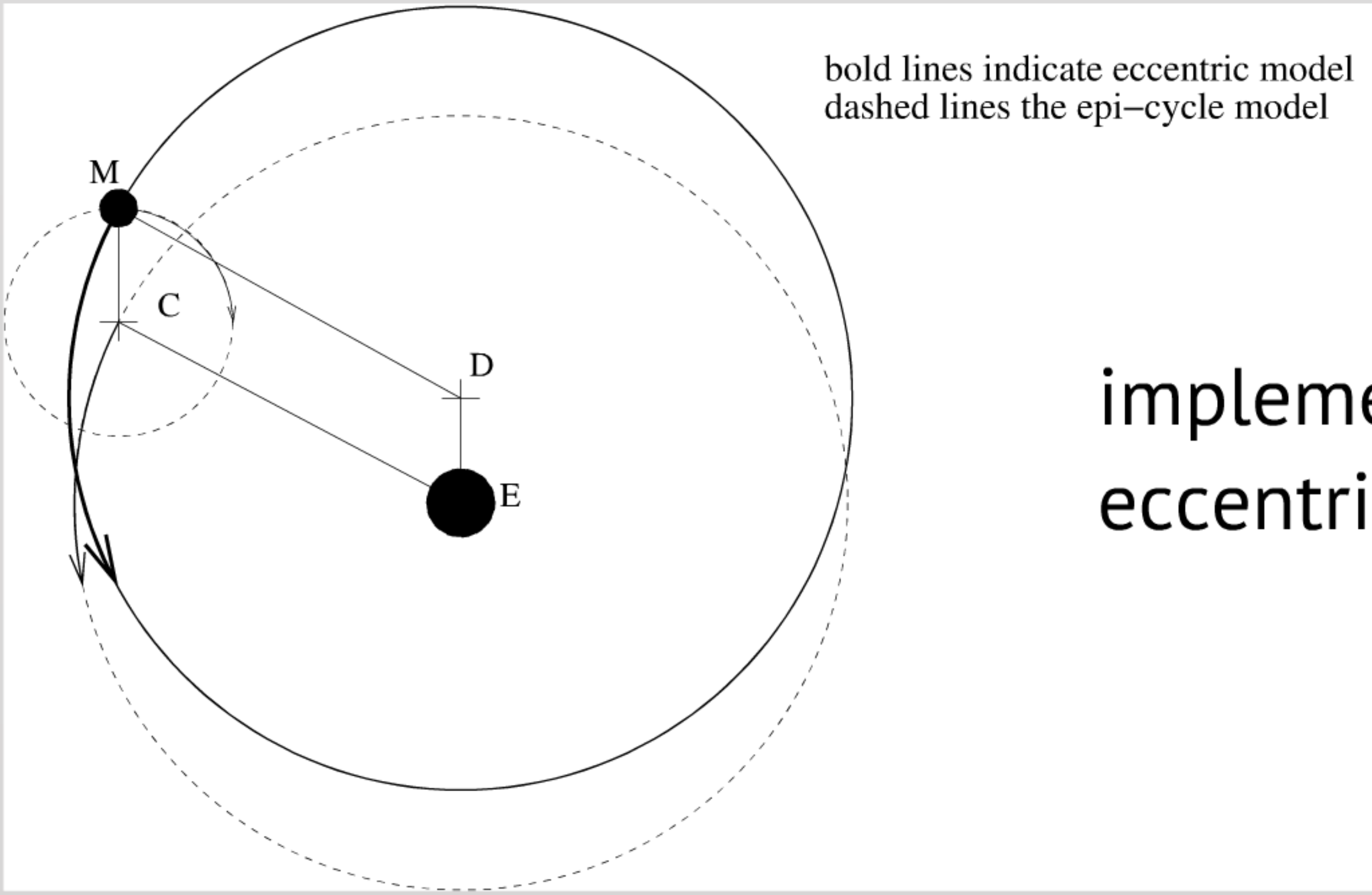
Antikythera computer, ancient Greek technology, predicts lunar eclipses and lunar positions, perhaps work of Appolonius



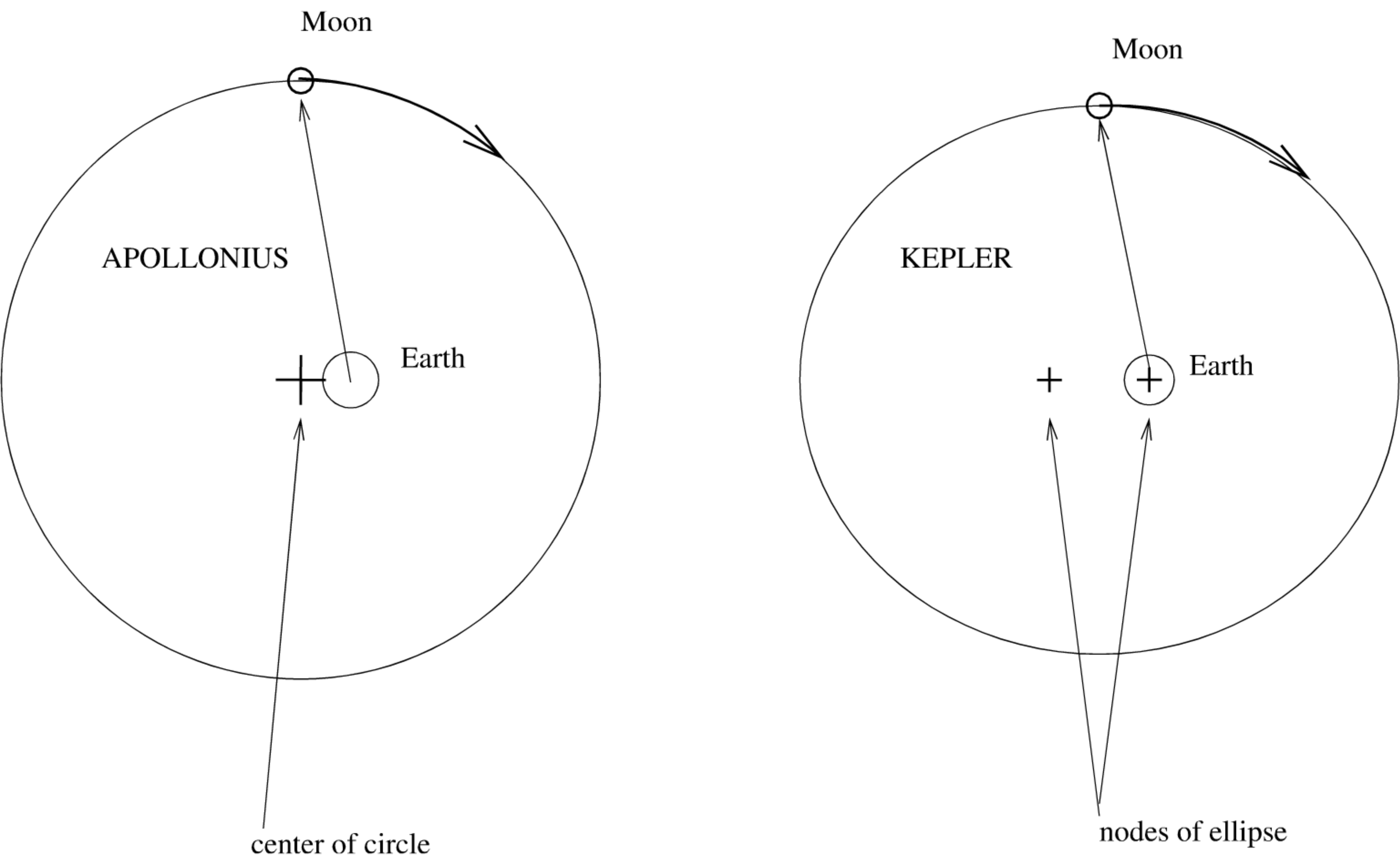
performs fixed ratio digital multiplications and additions

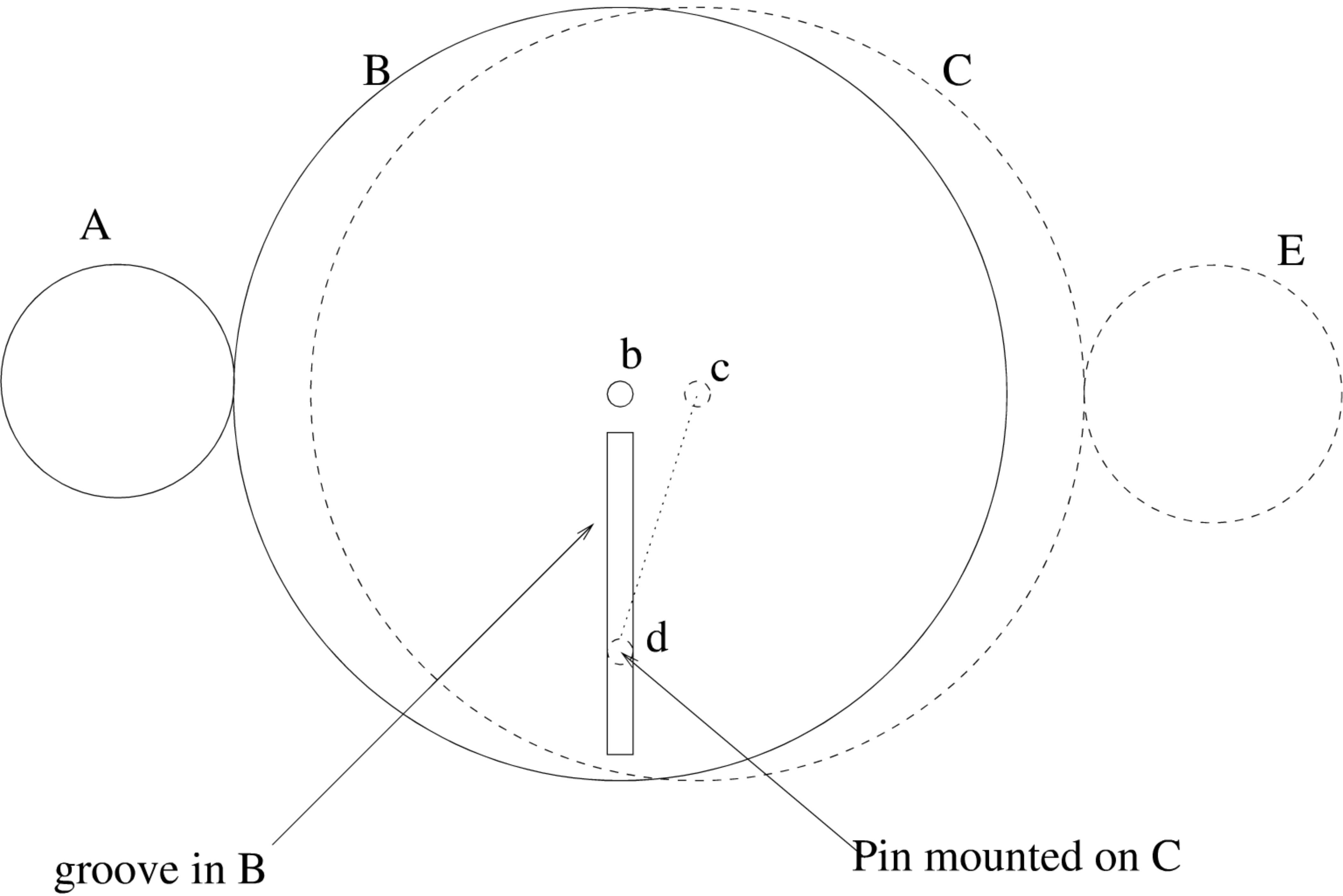


nius



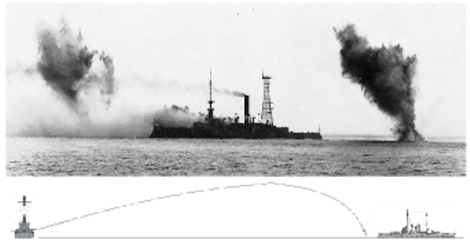
implements Appolonius eccentric orbital model


Moon
APOLLONIUS
Earth
center of circle
Moon
KEPLER
Earth
nodes of ellipse


B
C
A
E
b
c
d
groove in B
Pin mounted on C

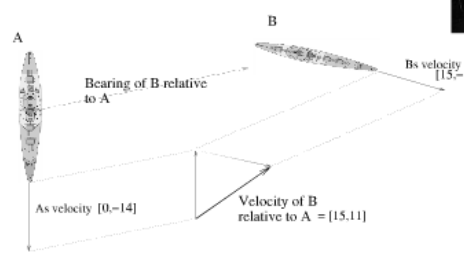
re Turing
l use

Analogue

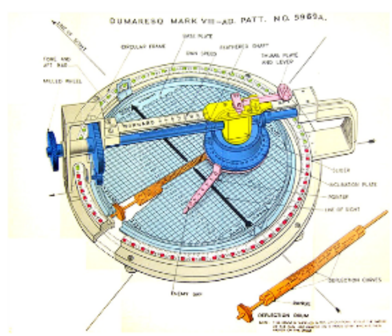


100 years ago navies had to solve real-time vector arithmetic problems and regression problems that were way beyond what could be done manually.

Ranges were about 10 miles
relative velocities up to 60 mph
time of flight of the order 20secs

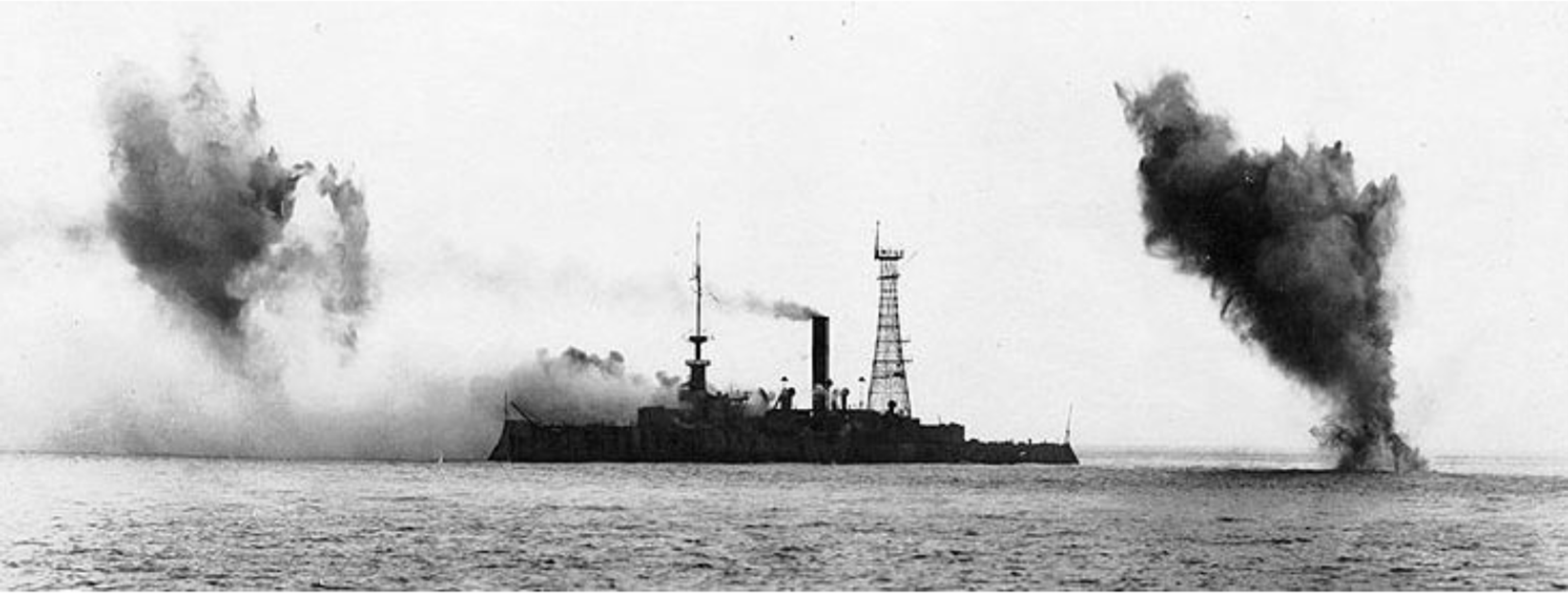


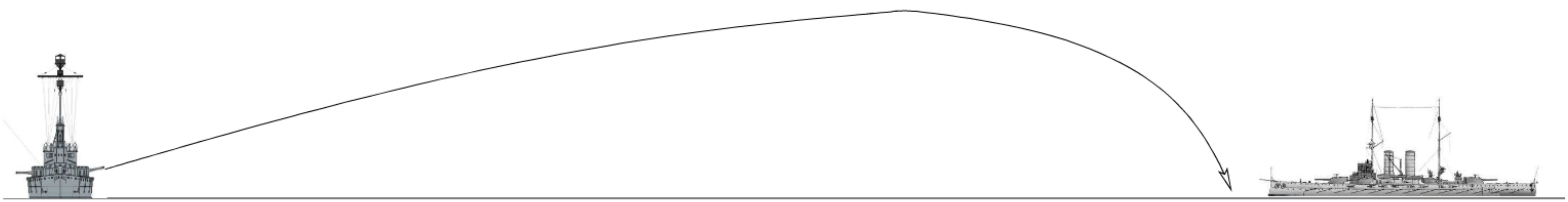
Rear Adrmiral John Dumaresq commander of Royal Australian Navy described by his colleagues in naval gunnery as a mathematical genius, and unsung pioneer of mechanical computing

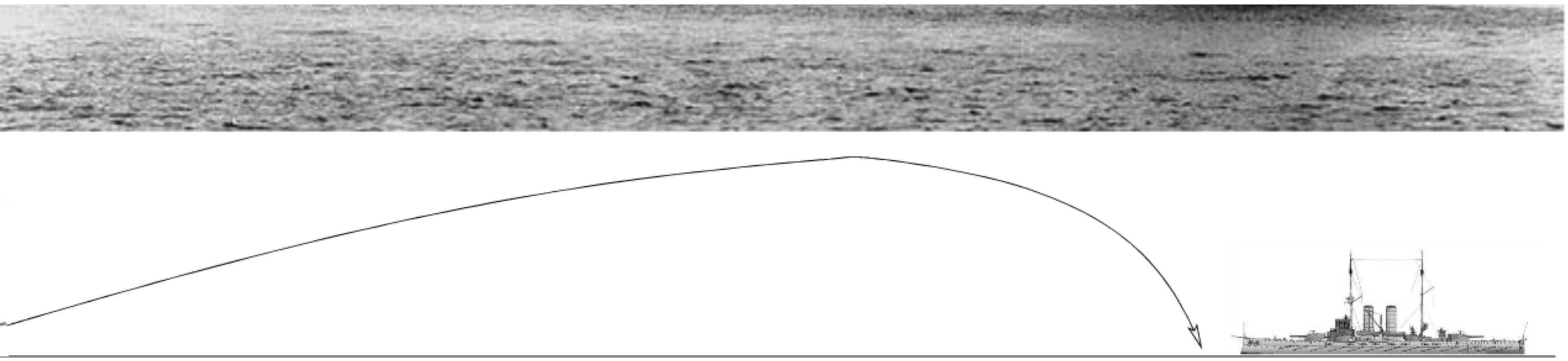


ogue

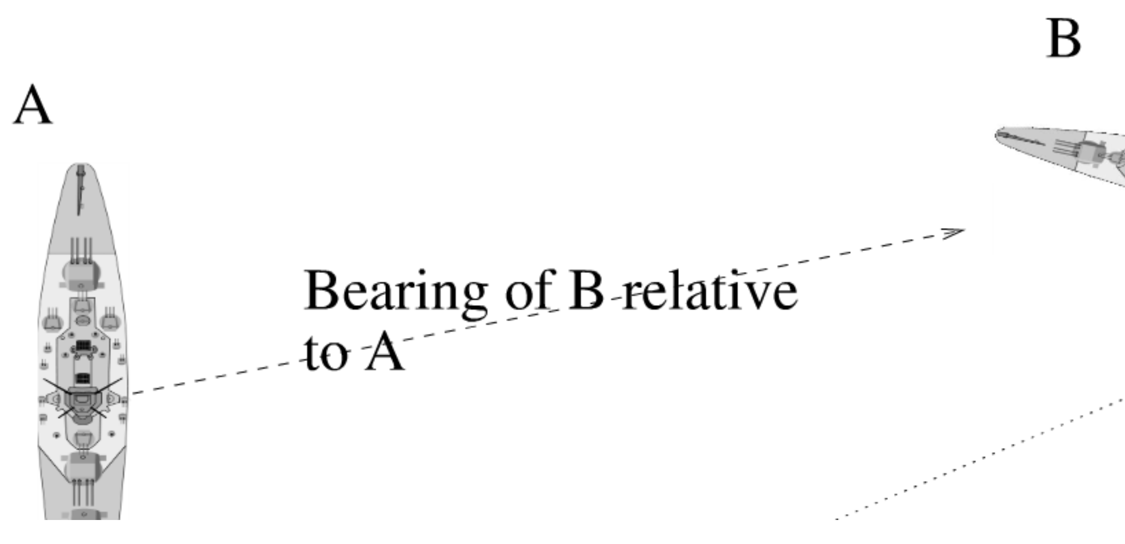
100 years ago navies had to solve real-time vector arithmetic problems and regression problems that were way beyond what could be done manually.



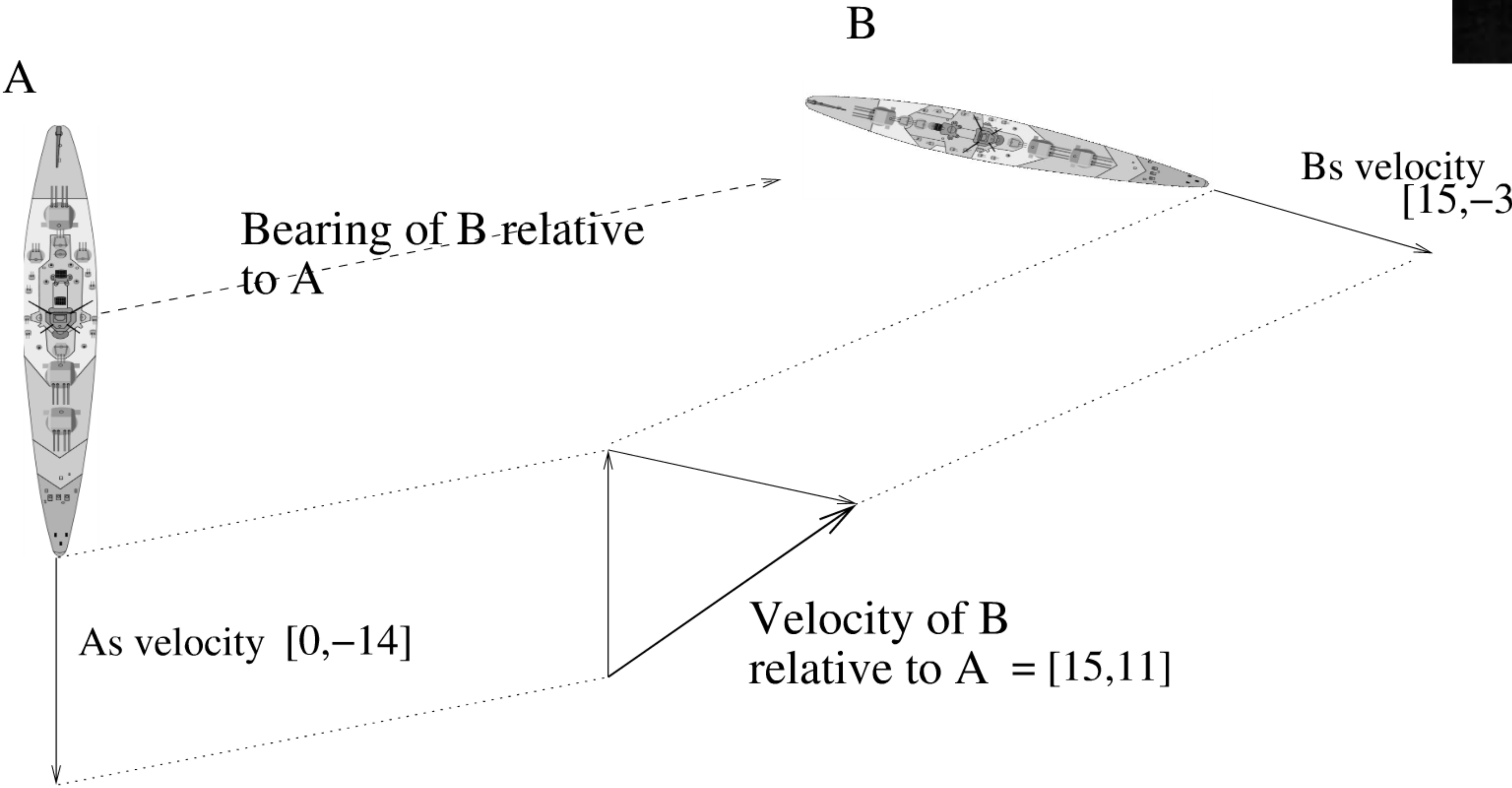


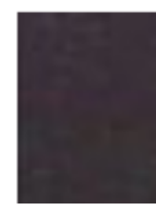


Ranges were about 10 miles
relative velocities up to 60 mph
time of flight of the order 20secs



f the order 20secs

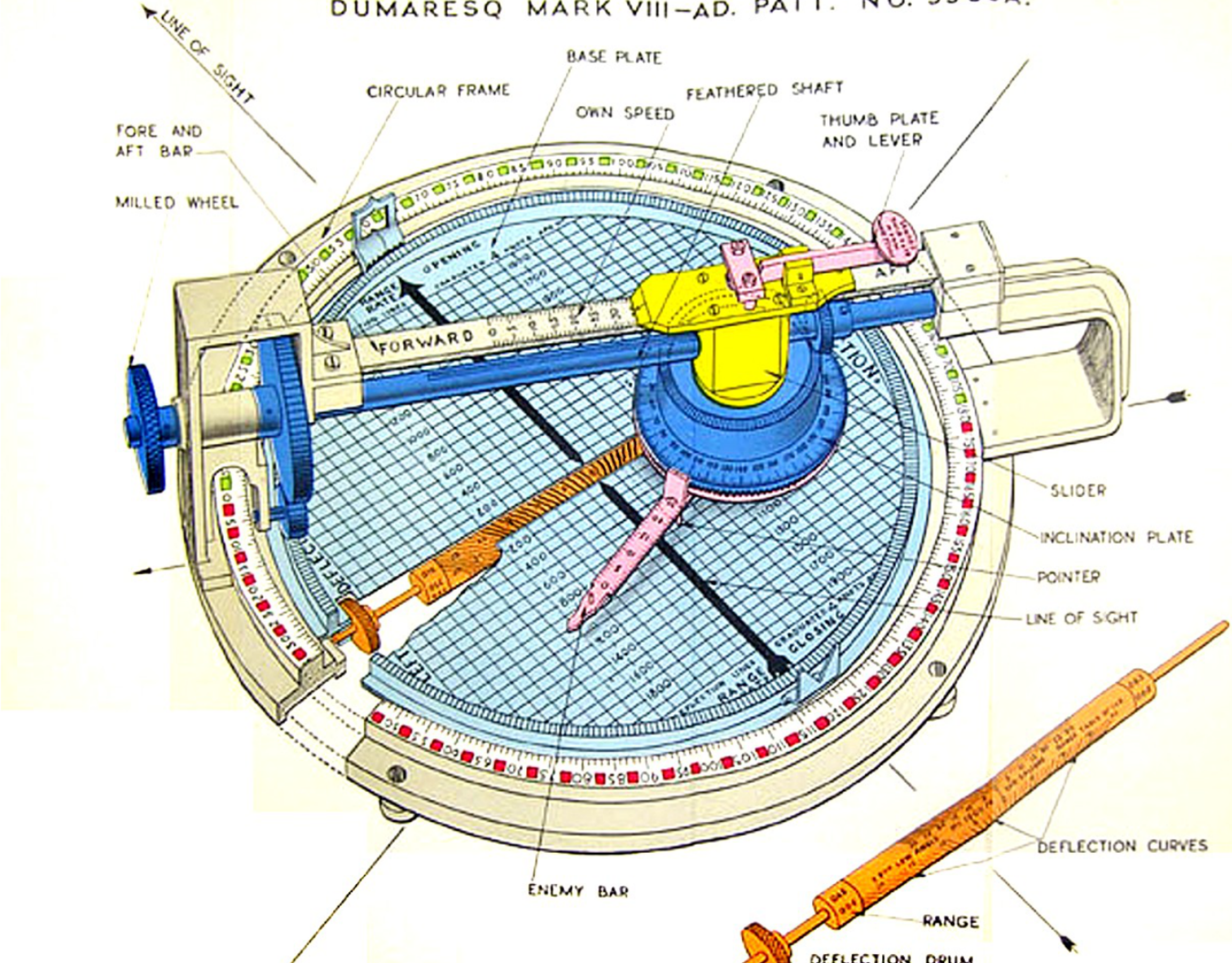




Dumaresq
l Australian
his colleagues in
mathematical
pioneer of
ing

Rear Adrmiral John Dumaresq commander of Royal Australian Navy described by his colleagues in naval gunnery as a mathematical genius, and unsung pioneer of mechanical computing

DUMARESQ MARK VIII-AD. PATT. NO. 5969A.



NOTE: THIS DRUM IS SUPPLIED BLANK, GRADUATIONS TO SUIT THE NATURE OF THE GUN ARE PRINTED ON A PAPER STRIP WHICH IS THEN PASTED ON THE DRUM

Self reference

Turing needed self reference
in order to address the decision problem.
But this property was the key to the
economical adoption of computer
technology. It gives Universality.
"This special property of digital computers, that they
can mimic any discrete-state machine, is described
by saying that they are universal machines."

Cheapness

The universal machine has the advantage that one design can be applied to any problem. This brings huge economy of scale to manufacturing.

rsal

Cheapness

e universal machine has
e advantage that one
sign can be applied to any
oblem. This brings huge
onomy of scale to
anufacturing.

Turing's

Universal

Digital

Computer

Digital

That the machine is digital however has a more subtle significance. It means firstly that numbers can be represented by strings of digits that can be as long as one wishes. One can therefore work to any desired degree of accuracy. This accuracy is not obtained by more careful machining of parts, control of temperature vari ations and such means, but by a slight increase in the amount of equipment in the machine. To double the number of significant figures, would involve increasing the amount of the equipment by a factor definitely less than two, and would also have some effect in increasing the time taken over each job. This is in sharp contrast with analogue machines, and continuous variable machines such as the differential analyser, where each additional decimal digit required necessitates a complete redesign of the machine, and an increase in the cost by as much as a factor of 10. (Turing, Lecture on the Automatic Computing Engine)

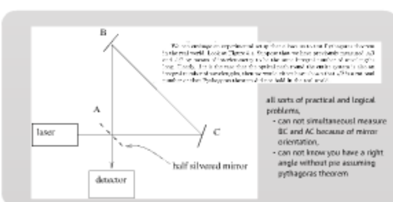
Turings argument there is very pragmatic. There has been a recent temptation to think that we can outperform digital computation by reverting to analogue computing.

This is based on a philosophical misconception that in reality everything is continuous, but we know this is false, everything is digital or quantised.

The notion of the continuum arose in classical Greek geometry from the proof of the irrationality of the length of the hypotenuse of a right triangle with unit sides.

If one assumes, as the Greeks did, that classical geometry was a true theory of the real world, this implied that space must be continuously subdivisible.

But do we know if Pythagorus theorem actually works in the real world? Could we experimentally test it?



More generally there is a fundamental limit to spatial accuracy provided by the Planck length of around 10^{-35} ,meters which limits the fundamental accuracy of any analogue computing device.

Most proposals for trans Turing computing are based on the illusion that real numbers are 'real' in the ontological sense.

They are based on continuum models of the world like Maxwells equations or Newtonian mechanics.

In the post-Turing era we have to see theories like Maxwells equations or Newtonian mechanics as software packages for making predictions about reality. When combined with a computer to do the maths they allow us to build models that mimic some part of reality.

But just like the Antikythera model, there are limits to the accuracy of our software packages. But our software packages are not reality itself.

Suppose that using some continuum model of mechanics we show that a system with computable boundary conditions has some points where some parameters are uncomputable, does this tell us that the real world can do things which are uncomputable?

No. It tells you that the software package, or physical theory you are using has a bug in it.

It was just such a 'bug' in Maxwells equations, the ultraviolet catastrophe, that let Einstein to invent the quantum theory.

Reality is digital, Turing computability rules.

That the machine is digital however has a more subtle significance. It means firstly that numbers can be represented by strings of digits that can be as long as one wishes. One can therefore work to any desired degree of accuracy. This accuracy is not obtained by more careful machining of parts, control of temperature vari ations and such means, but by a slight increase in the amount of equipment in the machine. To double the number of significant figures, would involve increasing the amount of the equipment by a factor definitely less than two, and would also have some effect in increasing the time taken over each job. This is in sharp contrast with analogue machines, and continuous variable machines such as the differential analyser, where each additional decimal digit required necessitates a complete redesign of the machine, and an increase in the cost by as much as a factor of 10. (Turing, Lecture on the Automatic Computing Engine)

Turings argument there is very pragmatic. There has machines such as the differential analyser, where each additional decimal digit required necessitates a complete redesign of the machine, and an increase in the cost by as much as a factor of 10. (Turing, Lecture on the Automatic Computing Engine)

Turings argument there is very pragmatic. There has been a recent temptation to think that we can outperform digital computation by reverting to analogue computing.

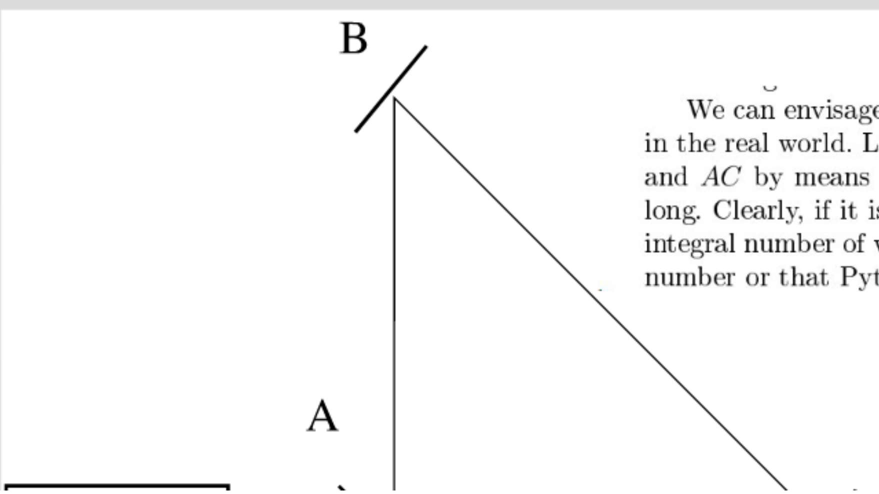
This is based on a philosophical misconception that in reality everything is continuous, but we know this is false, everything is digital or quantised.

The notion of the continuum arose in classical Greek geometry from the proof of the irrationality of the length of the hypotenuse of a right triangle with unit sides.

If one assumes, as the Greeks did, that classical geometry was a true theory of the real world, this implied that space must be continuously subdivisible.

If one assumes, as the Greeks did, that classical geometry was a true theory of the real world, this implied that space must be continuously subdivisible.

But do we know if Pythagorus theorem actually works in the real world? Could we experimentally test it?

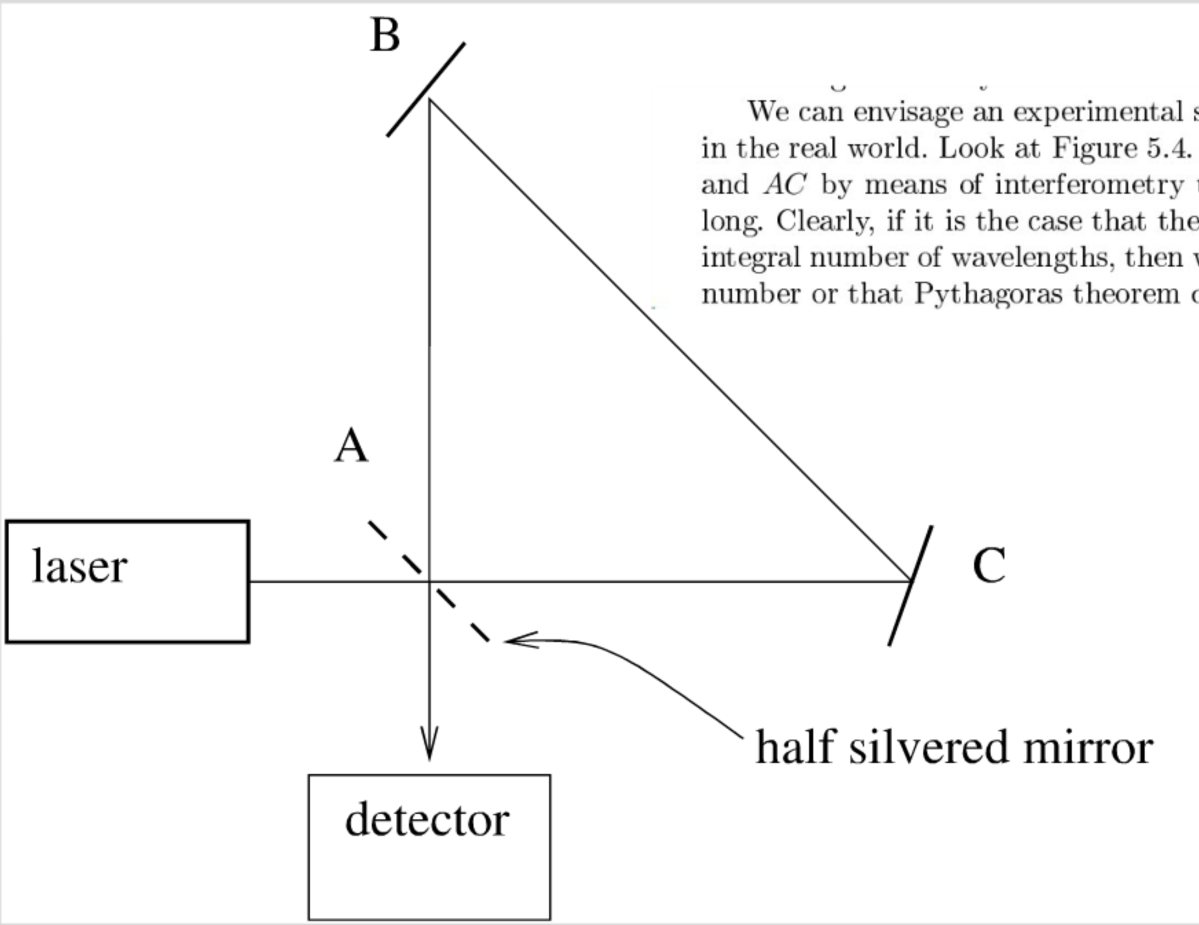


We can envisage an experimental setup that allows us to test Pythagoras theorem in the real world. Look at Figure 5.4. Suppose that we have previously measured AB and AC by means of interferometry to be the same integral number of wavelengths long. Clearly, if it is the case that the optical path round the entire system is also an integral number of wavelengths, then we would either have shown that $\sqrt{2}$ is a rational number or that Pythagoras theorem did not hold in the real world.

all sorts of practical and logical problems,

- can not simultaneousl measure

works in the real world? Could we experimentally test it?



We can envisage an experimental setup that allows us to test Pythagoras theorem in the real world. Look at Figure 5.4. Suppose that we have previously measured AB and AC by means of interferometry to be the same integral number of wavelengths long. Clearly, if it is the case that the optical path round the entire system is also an integral number of wavelengths, then we would either have shown that $\sqrt{2}$ is a rational number or that Pythagoras theorem did not hold in the real world.

all sorts of practical and logical problems,

- can not simultaneousl measure BC and AC because of mirror orientation,
- can not know you have a right angle without pre assuming pythagoras theorem

More generally there is a fundamental limit to spatial accuracy provided by the Planck length of around 10^{-35} ,meters which limits the fundamental accuracy of any analogue computing device.

Most proposals for trans Turing computing are based on the illusion that real numbers are 'real' in the ontological sense.

They are based on continuum models of the world like Maxwells equations or Newtonian mechanics.

In the post-Turing era we have to see theories like Maxwells equations or Newtonian mechanics as software packages for making predictions about reality. When combined with a computer to do the maths they allow us to build models that mimic some part of reality.

But just like the Antikythera model, there are limits to the accuracy of our software packages. But our software packages are not reality itself.

Suppose that using some continuum model of mechanics we show that a system with computable boundary conditions has some points where some parameters are uncomputable, does this tell us that the real world can do things which are uncomputable?

No. It tells you that the software package, or physical theory you are using has a bug in it.

It was just such a 'bug' in Maxwells equations, the ultraviolet catastrophe, that let Einstein to invent the quantum theory.

atial

sed

d
.

e

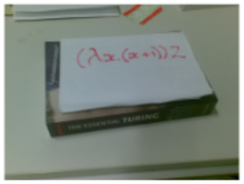
e

ts

Reality is digital, Turing computability rules.

Why Computer

Universal Computer
Equivalent to Lambda Caclulus?
No!



As a controled experiment a simple lambda expression was left overnight on top of book with relevant theory, but it failed to evaluate.

But put into the Lambda Can it worked

Type a term, Help for info or Quit to exit.

> (\x .(x+1)) 2

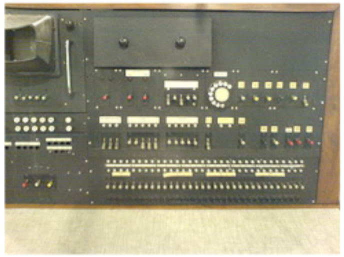
3
(7 reductions, 0.00s CPU)
lci>

The Lamda calculus is only equivalent to the Universal Computer if by the Lambda calculus we mean either

- *A Lambda interpreter on a Universal Computer*
- *Or a Mathematician, a blackboard and a definition of the calculus that the mathematician understands.*

Turing brings out the importance of physical embodiment for calculation.
By introducing a Machine he introduces MECHANICS and indirectly Physics as a support for mathematics.

Pilot Ace Console



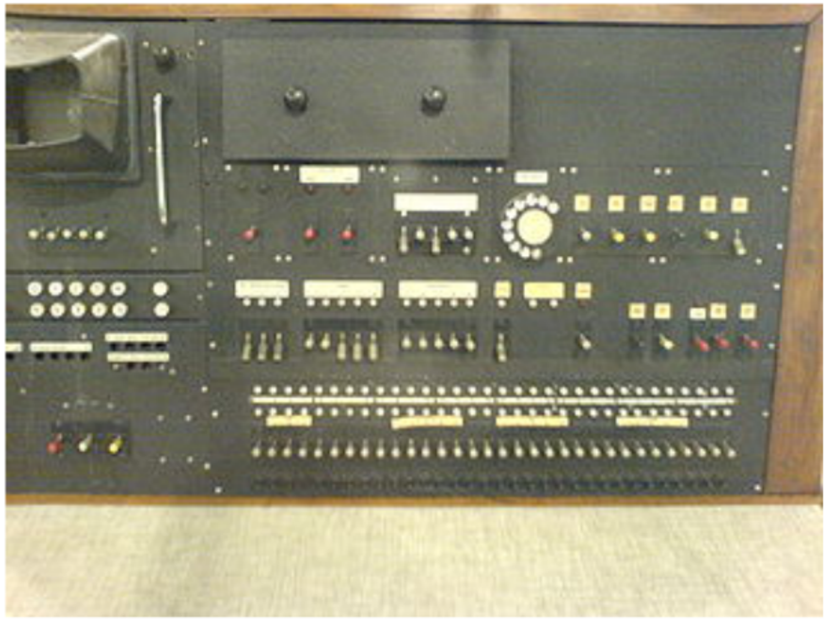
This was something real, not a thought machine.

Turings idea wa thus something much more fundamental and lasting than that of Church

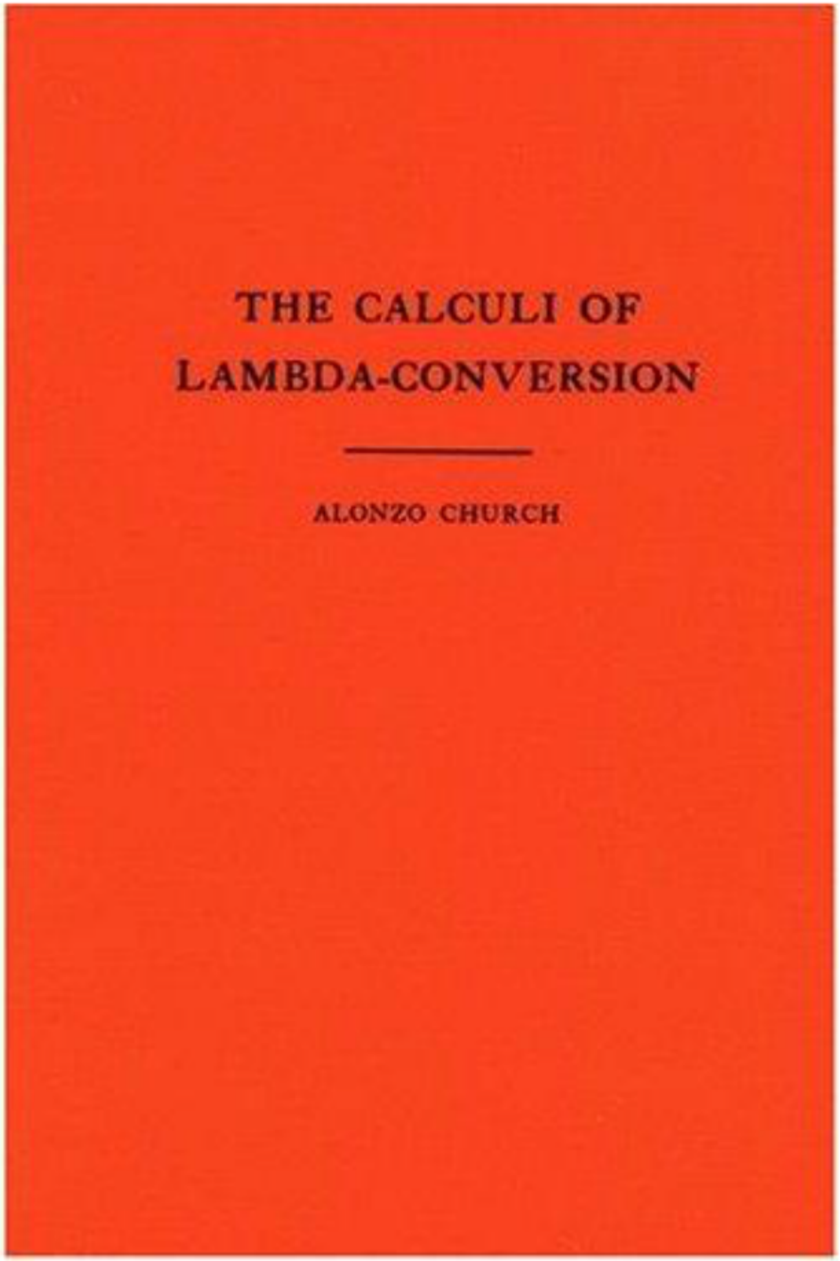
ook
t

Can

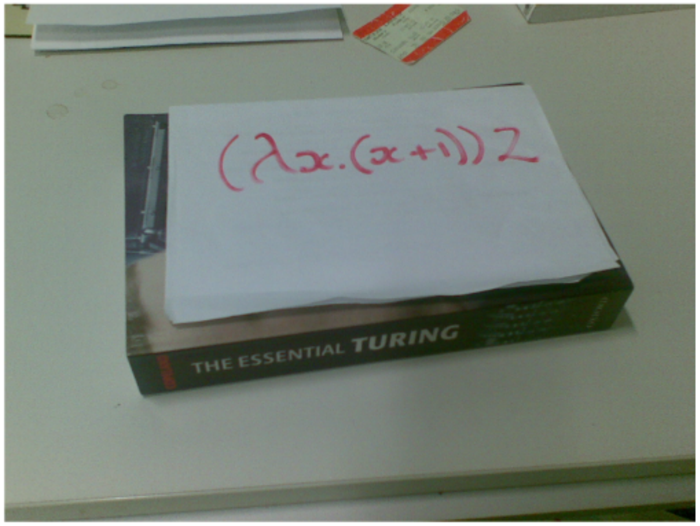
Pilot Ace Console



This was something real, not a thought machine.


THE CALCULI OF
LAMBDA-CONVERSION
ALONZO CHURCH

Universal Computer
Equivalent to Lambda Caclulus?
No!



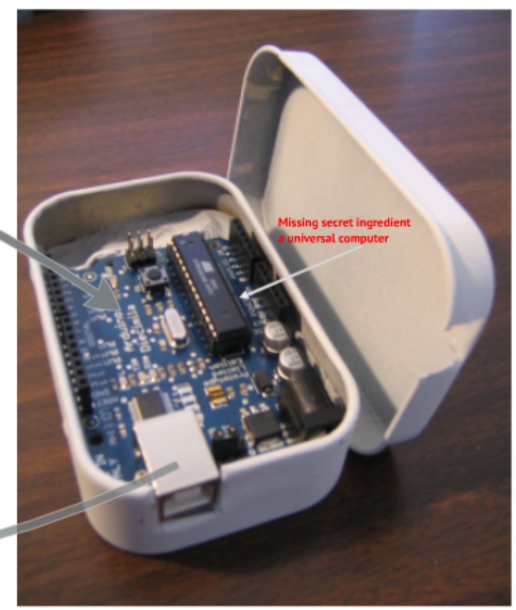
As a controled experiment a simple lambda expression was left overnight on top of book with relevant theory, but it failed to evaluate.

But put into the Lambda Can it worked

Type a term, Help for info or Quit to exit.

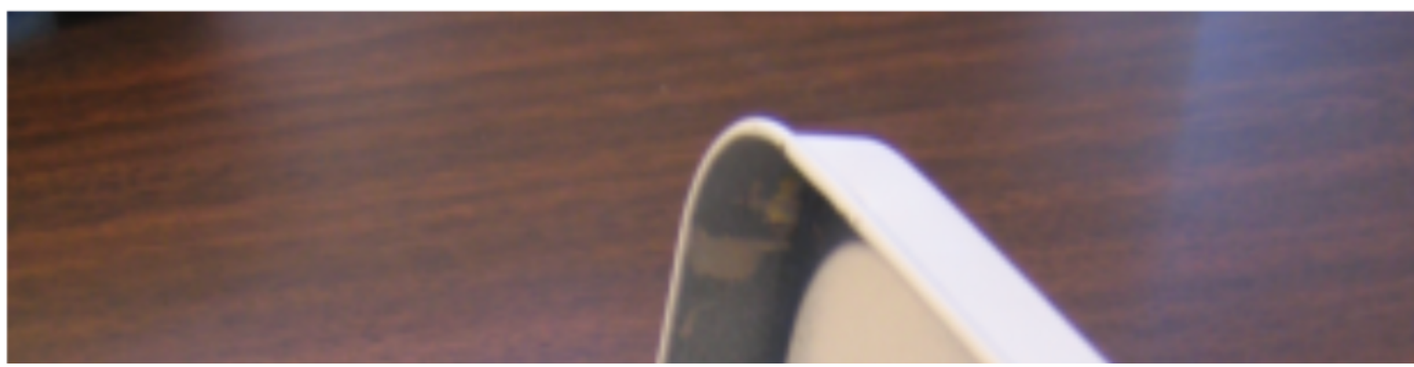
> (\x .(x+1)) 2

3
(7 reductions, 0.00s CPU)
lci>



elevant theory, but it
to evaluate.

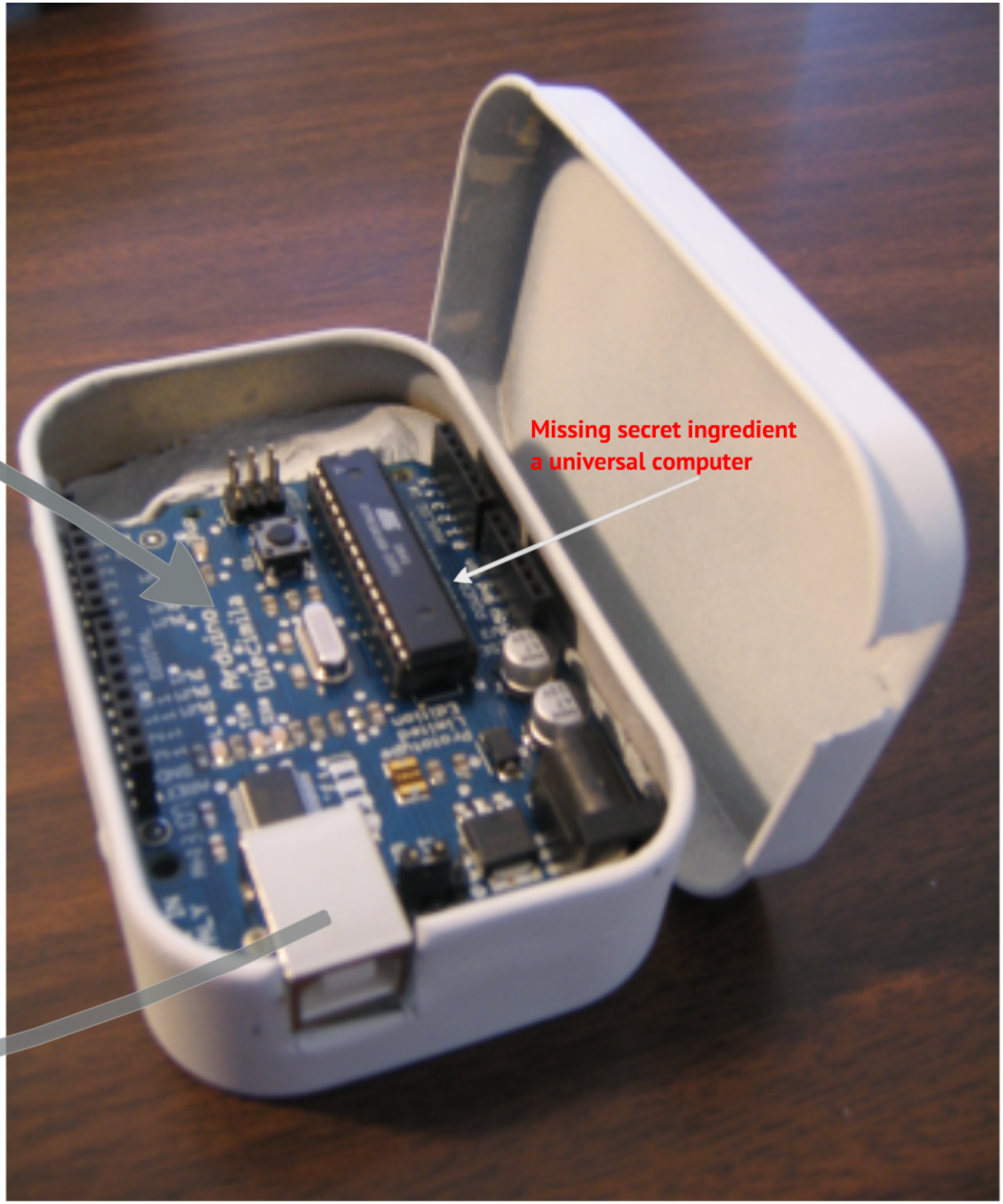
But put into the Lambda Can it worked



Type a term, Help for info or Quit to exit.

```
> (\x .(x+1)) 2

3
(7 reductions, 0.00s CPU)
lci>
```


Missing secret ingredient
a universal computer

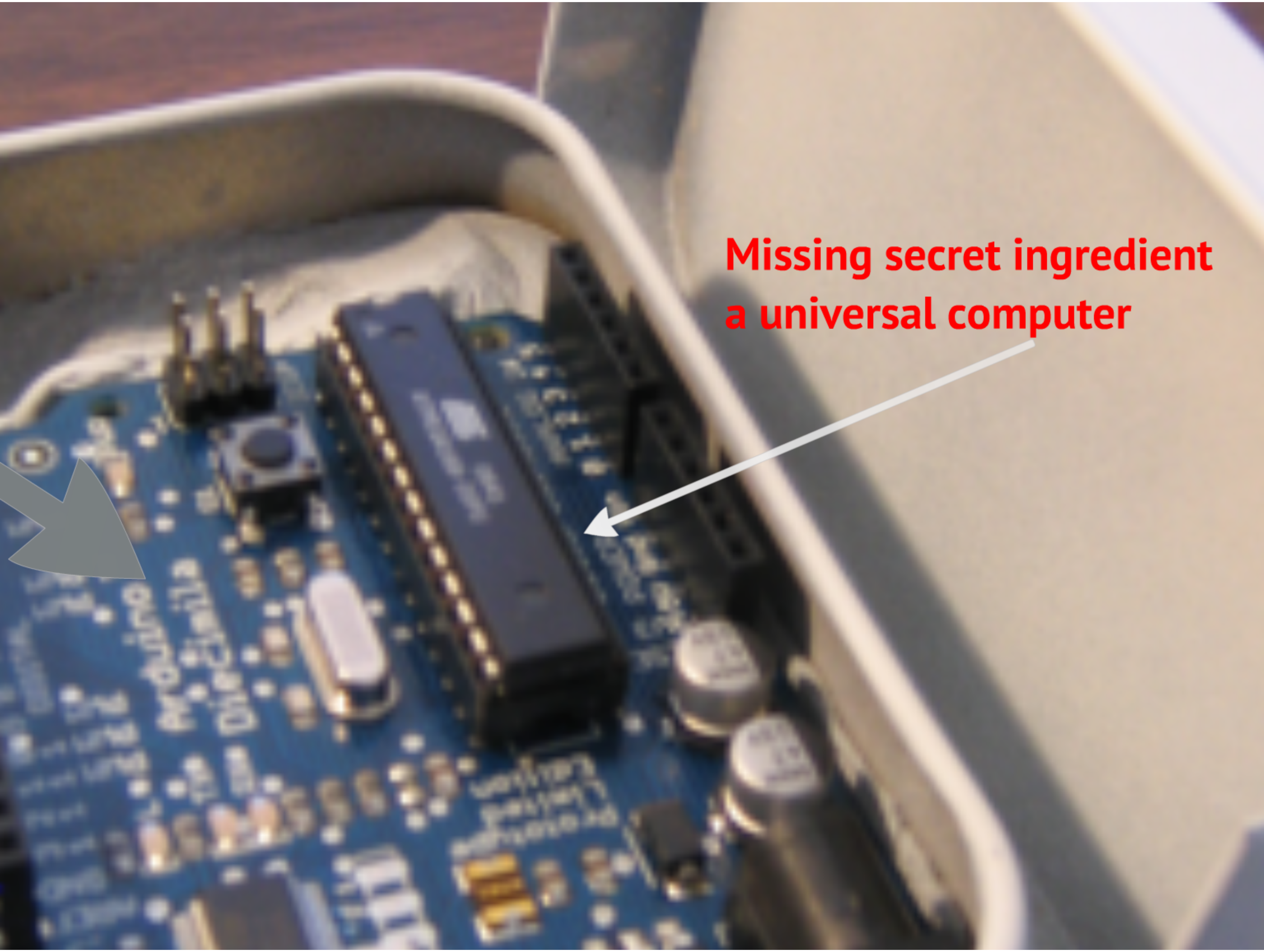
Type a term, Help for info or Quit to exit.

> (\x .(x+1)) 2

3

(7 reductions, 0.00s CPU)

lci>


Missing secret ingredient
a universal computer

The Lamda calculus is only equivalent to the Universal Computer if by the Lambda calculus we mean either

- *A Lambda interpreter on a Universal Computer*
- *Or a Mathematician, a blackboard and a definition of the calculus that the mathematician understands.*

Turing brings out the importance of physical embodiment for calculation.

By introducing a Machine he introduces MECHANICS and indirectly Physics as a support for mathematics.

Paul Cockshott

University of Glasgow

VIA VERITAS VITA

Greg Michaelson

HERIOT
WATT
UNIVERSITY