Computational and Experimental Mathematics: Further Experiments

Neil Calkin and Dan Warner

Department of Mathematical Sciences
Clemson University

November 15, 2010

A wealthy (15th Century) German merchant, seeking to provide his son with a good business education, consulted a learned man as to which European institution offered the best training. "If you only want him to be able to cope with addition and subtraction," the expert replied, "then any French or German university will do. But if you are intent on your son going on to multiplication and division – assuming that he has sufficient gifts – then you will have to send him to Italy."

From Georges Ifrah, The Universal History of Numbers: From Prehistory to the Invention of the Computer, translated from French, John Wiley, 2000, pg. 577. (Emphasizing the importance of the discovery of modern positional notation decimal arithmetic.)

Overview

In this session we'll explore some more advanced topics

- Bailey, Borwein and Plouffe's BBP algorithm for computing hexadecimal digits of π.
- Helaman Ferguson's PSLQ algorithm for finding integer relations between real numbers
- Using the continued fraction expansion of a number to output its digits.
- Lambert's continued fraction expansion for π.

This session will be accessible to those with some knowledge of undergraduate mathematics: the topics may include some unfamiliar ideas, but even the hard techniques will be elementary.

Setting up a Sage session

To get started with Sage open a browser, preferably Firefox, and

- Surf to the Sage Server at Clemson
  `https://clemix.clemson.edu:34567/`

Setting up a Sage session

To get started with Sage open a browser, preferably Firefox, and

- Surf to the Sage Server at Clemson `https://clemix.clemson.edu:34567/`
- Log in to your account (create one if you have not already done so)
- Click on Published
- Select the worksheet `CE_Math_III_01`
- Click on "Edit a copy"

Executing commands in Sage

Recall

- The Sage notebook uses blocks of text
  essentially Python statements that get executed
- Lines that start with the pound sign are comments
- A block is evaluated by clicking on the evaluate link
  or by typing shift-enter.

Computing digits of π

- Early 70's: everything was known about computing digits of π

Computing digits of π

- Early 70's: everything was known about computing digits of π
- 1976: Salamin and Brent's quadratically convergent algorithm

Computing digits of π

- Early 70's: everything was known about computing digits of π
- 1976: Salamin and Brent's quadratically convergent algorithm
- 1987: Borwein and Borwein's *Pi and the AGM*

Computing digits of π

- Early 70's: everything was known about computing digits of π
- 1976: Salamin and Brent's quadratically convergent algorithm
- 1987: Borwein and Borwein's *Pi and the AGM*
- 1990: Rabinowitz and Wagon's "spigot" algorithm

Computing digits of π

- Early 70's: everything was known about computing digits of π
- 1976: Salamin and Brent's quadratically convergent algorithm
- 1987: Borwein and Borwein's *Pi and the AGM*
- 1990: Rabinowitz and Wagon's "spigot" algorithm
- 1996: Bailey, Borwein and Plouffe compute the nth hexadecimal digit of π

Computing digits of π

- Early 70's: everything was known about computing digits of π
- 1976: Salamin and Brent's quadratically convergent algorithm
- 1987: Borwein and Borwein's *Pi and the AGM*
- 1990: Rabinowitz and Wagon's "spigot" algorithm
- 1996: Bailey, Borwein and Plouffe compute the nth hexadecimal digit of π *without computing previous digits*

Bailey-Borwein-Plouffe

Making use of the following marvellous identity (discovered by Plouffe using PSLQ)

$$\pi = \sum_{k=0}^{\infty} \left[\frac{1}{16^k} \left(\frac{4}{8k+1} - \frac{2}{8k+4} - \frac{1}{8k+5} - \frac{1}{8k+6} \right) \right]$$

Bailey, Borwein and Plouffe were able to identify the (finite number of) terms which contribute to the nth hexadecimal digit, and then used this to give an algorithm to compute it.

How to find such an identity?
Combine various series and compute digits:
see whether you get an interesting real number.

How to find such an identity?
Combine various series and compute digits:
see whether you get an interesting real number.
What is an interesting number?
How can you tell if a number is interesting?

How to find such an identity?
Combine various series and compute digits:
see whether you get an interesting real number.
What is an interesting number?
How can you tell if a number is interesting?
What is 0.055431383959305681 50?

PSLQ

The PSLQ algorithm is similar to using the Euclidean algorithm on real numbers. However, rather than two input parameters, there are several.
Given real numbers $r_1, r_2, \ldots, r_k$, the algorithm attempts to find an integer relation

$$a_1 r_1 + a_2 r_2 + \ldots a_k r_k = 0.$$

This can be used, for example, to identify numbers which are a linear combination of e and π: $\text{pslq}(x, e, \pi)$ will either return an integer linear relation, or tell you that it can't find one.

Identifying algebraic numbers

Algebraic numbers, those $\alpha \in \mathbb{R}$ which satisfy an integer polynomial equation

$$a_n \alpha^n + a_{n-1} \alpha^{n-1} + \cdots + a_1 \alpha + a_0,$$

can be identified by PSLQ: call $\text{pslq}(1, \alpha, \alpha^2, \ldots, \alpha^n)$.

Products

By massaging which parameters you pass to it, PSLQ can identify products as well as sums: to identify $x = \pi^2 e^{1/2}$, call

$$\text{pslq}(\log x, \log \pi, 1)$$

and it should tell you that

$$2 \log x - 4 \log \pi + 1 \times 1 = 0$$

It is important to note that the *user* needs to tell PSLQ which parameters to investigate: unless you ask it to determine which reals to use, it can't find the relation.
Various packages, such as Maple's identify, the Inverse Symbolic Calculator, etc automate this to some extent.
However, all these tools have limitations: none of them are good, for example, at identifying sums of products of real numbers. And the automated versions fail to identify the following simple integer linear combination of three well known constants.

$$.055431383959305681 50$$

And now

And now The book giveaway.

Our Thanks

Once again, our thanks to

- The SuperComputing 2010 Conference, the Education Program Organizers and all the affiliated sponsors.

Our Thanks

Once again, our thanks to

- The SuperComputing 2010 Conference, the Education Program Organizers and all the affiliated sponsors.
- You, the audience, for your attention