MCS 320 Project One : PSLQ = Partial Sum of Least Squares
The PSLQ Integer Relation Algorithm

Let $\mathbf{x}$ be a vector of n real numbers. An integer relation algorithms finds n integers $\mathbf{a}$ such that

$$a_1x_1 + a_2x_2 + \cdots + a_nx_n = 0. \tag{1}$$

The Euclidean algorithm solves this problem for $n = 2$.

The PSLQ algorithm by David Bailey and Helaman Ferguson [2] is the best-known integer relation algorithm. PSLQ was named one of ten "algorithms of the century" by Computing in Science and Engineering [1]. The algorithm requires high-precision arithmetic software: at least $d \times n$ digits, where d is the size (in digits) of the largest of the integers a_k.

One application of the PSLQ-algorithm is to find polynomials with integer coefficients that vanish at approximate roots. If α is an algebraic number (the root of a polynomial), then `evalf`(α) returns an approximation for the corresponding root. The output of PSLQ establishes the opposite direction: with the polynomial with integer coefficients that vanishes at an approximate root, we can declare a corresponding algebraic number. To find polynomials which vanish at an approximate root `s` we give a sequence consecutive powers of `s` as input to the PSLQ-Algorithm, as illustrated below:

```
[> p := randpoly(z, degree = 5);
[> factor(p);
[> s := fsolve(p, z);
[> x := [seq(s^i, i = 0 .. 5)];
[> a := (IntegerRelations[PSLQ])(x);
[> (LinearAlgebra[Multiply])((LinearAlgebra[Transpose])(Vector(a)), Vector(x));
```

We computed one approximate root of an irreducible quintic and gave its consecutive powers as input to the PSLQ algorithm. The last command above verifies whether the approximation satisfies the integer relation PSLQ found. As the outcome of that verification was unsatisfactory, we increased the precision:

```
[> Digits := 100;
[> t := fsolve(p, z);
[> y := [seq(t^i, i = 0 .. 5)];
[> b := (IntegerRelations[PSLQ])(y);
[> (LinearAlgebra[Multiply])((LinearAlgebra[Transpose])(Vector(b)), Vector(y));
[> sort(p);
```

We see that if the working precision is 100 decimal places, we can recover the polynomial we started with.

In general it is not known in advance how large the working precision ought to be for PSLQ to return correct results. The main goal of this project is to conduct an experimental study to see the relationship between the degree of the polynomial and the working precision required by PSLQ.

Assignment One: experimental mathematics

We can use PSLQ to "discover" the following identities [3] for π:

$$\pi = 48 \arctan\left(\frac{1}{49}\right) + 128 \arctan\left(\frac{1}{57}\right) - 20 \arctan\left(\frac{1}{239}\right) + 48 \arctan\left(\frac{1}{110443}\right) \tag{2}$$

and

$$\pi = 176 \arctan\left(\frac{1}{57}\right) + 28 \arctan\left(\frac{1}{239}\right) - 48 \arctan\left(\frac{1}{682}\right) + 96 \arctan\left(\frac{1}{12943}\right). \tag{3}$$

Assignment One. Give all Maple commands (with their output) needed to use PSLQ to find the integer coefficients of the relations (2) and (3), given the arguments of the arctan() function.

Suppose one would not know where to evaluate the arctan(), describe then a method to find the proper arguments of the arctan() function. (*hint*: use enumeration)

Use a properly documented and well structured Maple worksheet to formulate your answer to this assignment.

Assignment Two: the precision PSLQ needs

In the example with the algebraic root of a polynomial, we experienced that the standard default working precision of 10 decimal places was insufficient to recover the coefficients of the quintic.

Assignment Two. Take *irreducible* polynomials of increasing degrees with integer coefficients, generated by `randpoly`. For each polynomial, determine the minimal value of Digits needed to recover the polynomial when PSLQ is applied to the consecutive powers of an approximate root.

Use your UIN number (the blue numbers on your UIC i-card) as argument to the Maple command `randomize()` at the start of your worksheet. This command initializes the seed of the random numbers used in `randpoly`.

Start at degree 3 and go up to degree 8. Do at least three experiments with three different polynomials of the same degree. To summarize your experiment, display the minimal value for Digits obtained in each experiment in an orderly fashion, e.g., like in a table.

The deadline is Wednesday 28 February 2007 at 10AM

Bring *your* solution to the project to class. The *your* is emphasized to stress that your solution is the result of an *individual* effort. Collaborations are **not** permitted.

The solution to this project consists in two parts:

1. A print out of the Maple worksheet that you bring to class.

   Deliver a well written document, with grammatically correct and complete sentences, without spelling mistakes, appropriately structured into sections and subsections.

2. The Maple worksheet that you email as an attachment to me.

   The worksheet should run as a computer program, from top to bottom with consistent output and without errors.

If you have questions or difficulties with the assignments, feel free to come to my office for help.

References

[1] D. Bailey. Integer relation detection. *Computing in Science & Engineering*, 2(1):24–28, 2000.

[2] D. Bailey and H.R.P. Ferguson. Numerical results on relations between fundamental constants using a new algorithm. *Mathematics of Computation*, 53(188):649–656, 1989.

[3] J. Borwein and D. Bailey. *Mathematics by Experiment. Plausible Reasoning in the 21st Century.* A.K. Peters, 2004.