Development of NZMATH

MATSUI Tetsushi

Department of Mathematics and Information Sciences
Tokyo Metropolitan University

ICMS 2006 at Castro Urdiales 2006-09-01

1. Table of Contents
2. Introduction
   - History
   - Features
3. Concepts
4. Development
5. Future
6. Demonstration

pre NZMATH

Once upon a time...

pre NZMATH

Once upon a time... there was a system called SIMATH.

pre NZMATH

Once upon a time... there was a system called SIMATH.
I talked about it at ICMS 2002 poster session.

dawn of NZMATH

When the development of SIMATH stopped in 2003, we began to create a new system.

NZMATH!

We named it NZMATH [niʒimaθ].
The name is not an acronym nor related to New Zealand.

Releases

We have released sevral versions.
The latest is 0.5.1 released in August 2006.

What kind of system NZMATH is?

- number theory
- Python
- free

Number Theory

The primary goal of the development is to implement various number theoretic algorithms.

Python

NZMATH is written in Python and provided as a library for Python. In `nzmath` package there are more than 20 modules.

Free

NZMATH is distributed under the BSD license.

Basic Concepts

1. user / developer fusion
2. speed of development

User / Developer Fusion

Ideally, there should be no distinction between users and developers.

Choice of Language

The implementation language should be the same language for users.

Choice of Language

The implementation language should be the same language for users.
⇒ we chose Python.

Speed of Development

We put emphasis on the development speed of system rather than the execution speed of resulting programs.

Practices in Development

Some practices seem useful for better development.

Open Source

We believe that there is no need to restrict usages of the system, and that users should have rights to fork or take over the project, if the development would stall.

Outsourcing

We have just started to use sourceforge.

NZMATH Project Page

`http://sourceforge.net/projects/nzmath/`

Manual

Manual is maintained on Wiki.

NZMATH Wiki

`http://nzmath.sourceforge.net/wiki/`

Agile Development

- tests
- reviews
- small releases
- incremental design

Test Code Example

```
class Arith1Test (unittest.TestCase):
    def testFloorsqrt(self):
        self.assertEqual(0, arith1.floorsqrt(0))
        self.assertEqual(1, arith1.floorsqrt(1))
        self.assertEqual(1, arith1.floorsqrt(3))
        self.assertEqual(2, arith1.floorsqrt(4))
        self.assert_(arith1.floorsqrt(2**60 - 1) ** 2
                       <= 2**60 - 1)
        self.assert_(arith1.floorsqrt(2**59 - 1) ** 2
                       <= 2**59 - 1)
```

Agile Development

- tests
- reviews
- small releases
- incremental design

Future Direction of Development

Awaited features.

- Web user interface
- Algebraic numbers
- Speed of execution

Web UI

There is no particular interface for NZMATH, in other words, the Python interpreter is the only way to use it.
Web UI, i.e. a user interface using a web browser is a possible user interfaces other than the Python interpreter.

Web UI

There is no particular interface for NZMATH, in other words, the Python interpreter is the only way to use it.
Web UI, i.e. a user interface using a web browser is a possible user interfaces other than the Python interpreter.

We will show you later a demonstration of a Web UI prototype.

Algebraic Numbers

Algebraic numbers and algebraic number fields are the next main topic of the development. We are working, now.

Speed of Execution

Though we put priority on speed of development rather than of execution, we are trying to make the program execution as fast as possible.

Long Term Plans

- Elliptic curves over the rational field
- Analytic number theory tools
- Connecting to other systems

End of the slides

Now, let's take a look at demos.

End

Thank you.