MATH 676

–

Finite element methods in scientific computing

Wolfgang Bangerth, Texas A&M University

Lecture 23:

Learning to use modern tools, part 3:

The role of documentation
and
Doxygen – a documentation tool

Documenting code

Definition:

Code documentation is the collection of additional pieces of information *about the code*!

- This information may be *implicit in the code* (e.g. naming conventions)
- Or it may be *part of the actual code* (e.g. assertions)
- Or it may be *embedded in the code* (e.g. comments)
- Or it may be in entirely separate documents

Documenting code

Observation 1:

Documentation is necessary not just for others!

- Yes, if others are to read your code, there needs to be documentation
- But you will also often not remember in just 6 months
  - how this algorithm is supposed to work
  - what this cryptically named function does
  - how these 7 functions are working together

Documenting code

Observation 2:

Documentation is necessary on many levels!

- Within the implementation of individual functions
- What does this function do?
- What does this group of functions/this class do?
- How does this set of classes (a *module*) interact?
- What is the overall design of this software?
- How to use this software (e.g. a README file)?

Documenting code

Observation 3:

Documenting codes is a *process*!

- Writing documentation as an afterthought to coding doesn't work
- You need to do it as an *integral part* of programming!

Documenting code

Observation 3a:

Documenting your code can make it better!

Writing documentation first (“design by contract”):

- You *specify up front* and in detail what the code is supposed to do
- Allows unit tests to be written *before* the code is actually there
- Roles of functions are clearly defined, no “the purpose of this function is sort-of to make sure that this other thing works correctly and fix up its output”

Documenting code

Observation 3a:

Documenting your code can make it better!

Writing code first:

- You *specify after writing the code* and in detail what the code is supposed to do
- Have to *review* the code again after writing it to describe what it does
- Reviewing code is an opportunity for clean-up and improvements

Documenting code

Observation 3a:

Documenting your code can make it better!

In practice:

- Design by contract is more successful and leads to better code
- Documenting after the fact often leads to little documentation because of the pressure to move on.

Thus:

Use design by contract!
Write documentation first!

Documenting code

Observation 4:

Keeping documentation up to date in separate files is almost impossible!

- If you change the code and forget to update the documentation, then the two quickly drift apart
- For proper external documentation, part of the code will have to be copied; this creates another source of divergence

Thus:

Use tools that extract documentation from code!

Doxygen

***Doxygen* is a tool that helps you with all of this:**

- Extracts documentation from code
- Allows you to annotate code in the form of comments
- Allows you to include additional documents

- **Cross references everything!**

Information on *doxygen*:

http://www.doxygen.org/

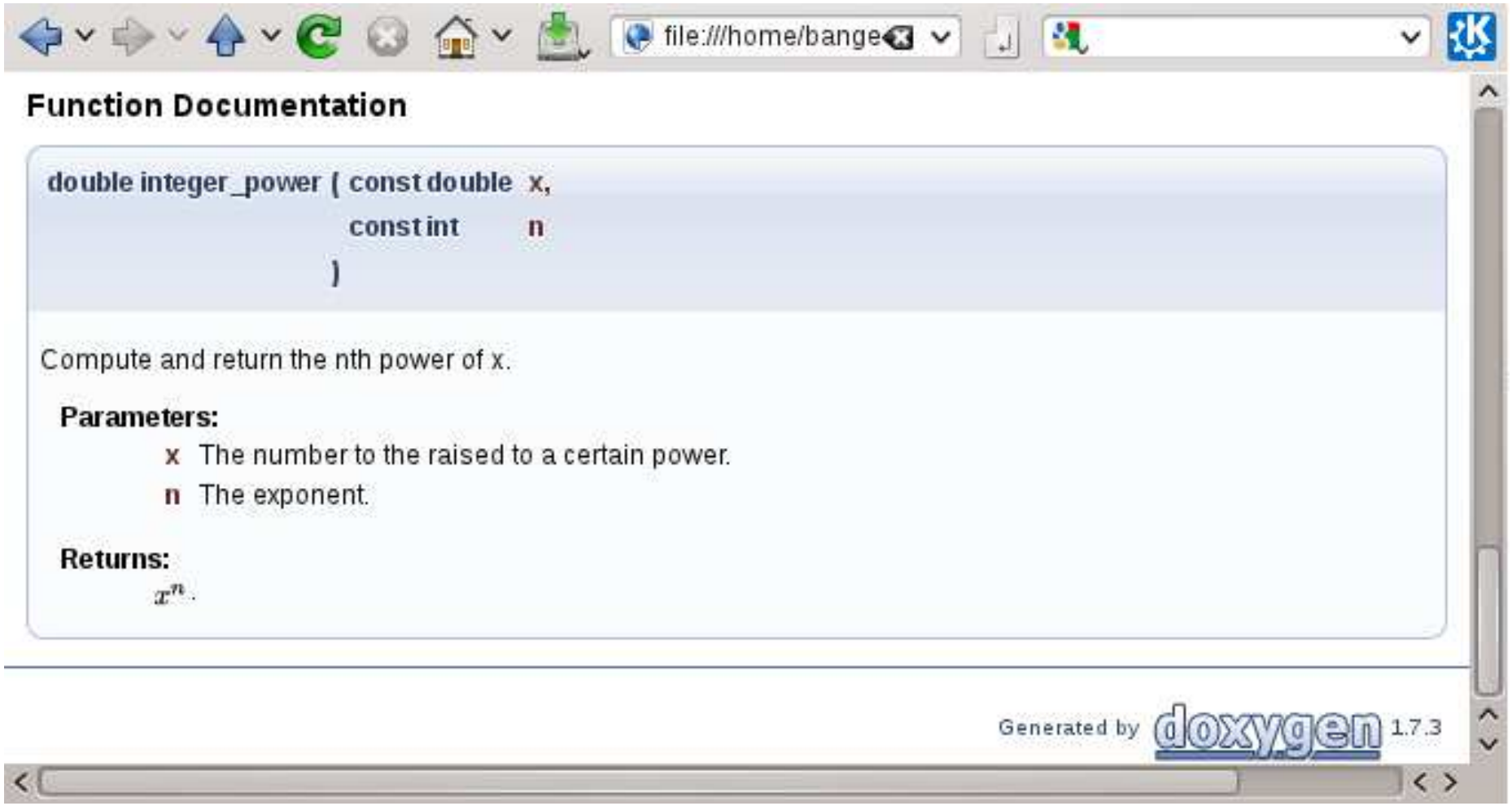
Doxygen

Example 1: Source code with doxygen markup

```
/**
 * Compute and return the <code>n</code>th power of
 * <code>x</code>.
 *
 * @param x The number to be raised to a certain power.
 * @param n The exponent.
 * @return @f$ x^n @f$.
 */
double integer_power (const double x,
                      const int    n);
```

Doxygen

Example 1: HTML output from doxygen



Doxygen

Example 2: Source code with doxygen markup

```
/**
 * Frob a nix and dub the step (see the Dubstep class).
 *
 * @ingroup HipHapHop
 * @author Wolfgang Bangerth, 2012
 */
class Frobnix
{};
```

Doxygen

Example 2 HTML output from doxygen

Note: (i) the automatic link to the *Dubstep* class; (ii) the reference to group *HipHapHop*.

Doxygen

Final notes on *doxygen*:

- Cross references everything!
- Supports documentation at all levels
- Has dozens of @-commands to mark up text and give it semantic meaning

- Creates the thousands of pages of documentation for deal.II, including the tutorials

Information on *doxygen*:

http://www.doxygen.org/

MATH 676

–

Finite element methods in scientific computing

Wolfgang Bangerth, Texas A&M University