Past and future of deal.II

Wolfgang Bangerth
Department of Mathematics
Texas A&M University

How it all started

In reality, deal.II is a garage project.

Nothing was planned.

The dudes who started it had not the first idea.

How it all started (Guido)

- Bonn, 1992: Some FE codes with Franz-Theo Suttmeier; thoughts about adaptive meshes
- Heidelberg, 1993-~2000: Work on a library called DEAL (Differential Equations Analysis Library)
  - mixed triangular/quadrilateral 2D meshes
  - hanging nodes
  - bilinear elements
- Guido and Roland Becker add a multigrid component
- No templates, complicated data structures for mixed tri/quad meshes lead to difficult and error prone code base
- “Poor” software design:
  - little documentation
  - no assertions
  - no test cases

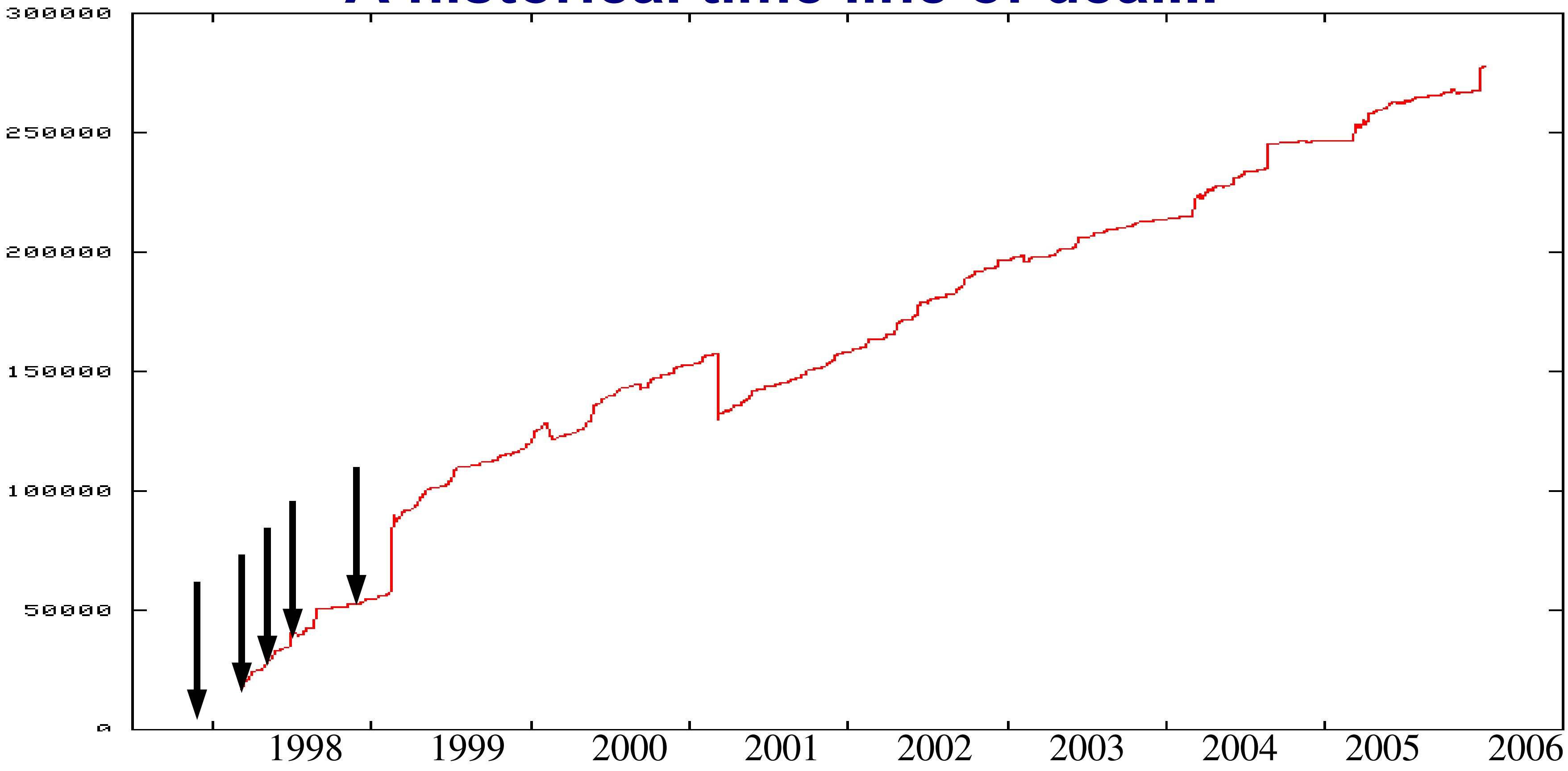
How it all started (Wolfgang)

- Heidelberg, 1997: Finite element code taught by Guido and Franz-Theo Suttmeier.
- Learns about a fair number of mistakes one can make with finite element codes, but `Point<dim>` is written
- Reads about DiffPack, the first big open source C++ finite element library
- Heidelberg, 1998: Doesn't like DEAL, needs something for diploma thesis, so writes something new

How it all started (Ralf)

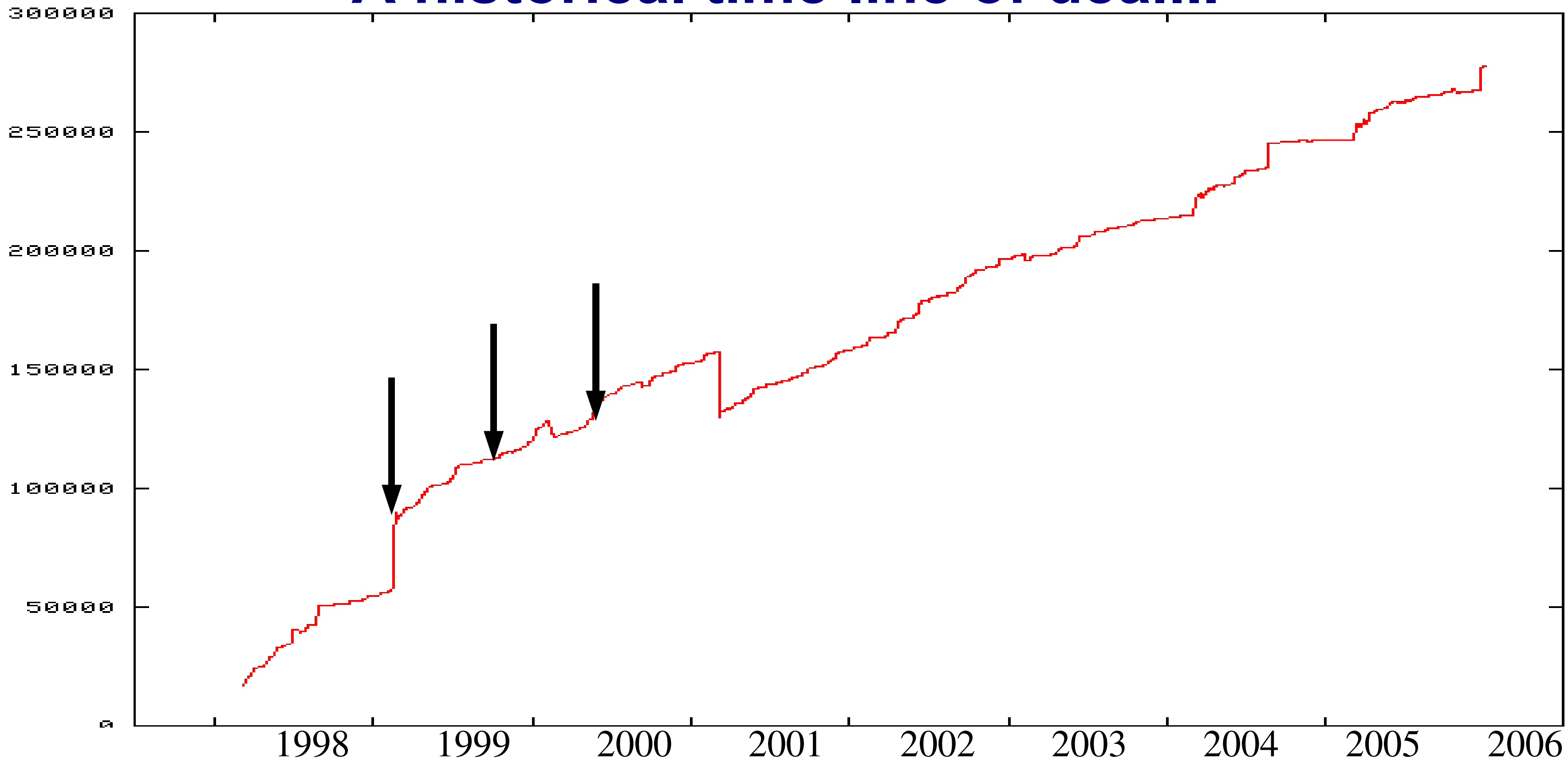
- Heidelberg, 1997: Finite element code taught by Guido and Franz-Theo Suttmeier.
- Learns about a fair number of mistakes one can make with finite element codes
- Heidelberg, 1998: Uses DEAL for diploma thesis
- Heidelberg, 1999: Wants something better for his PhD thesis

A historical time line of deal.II



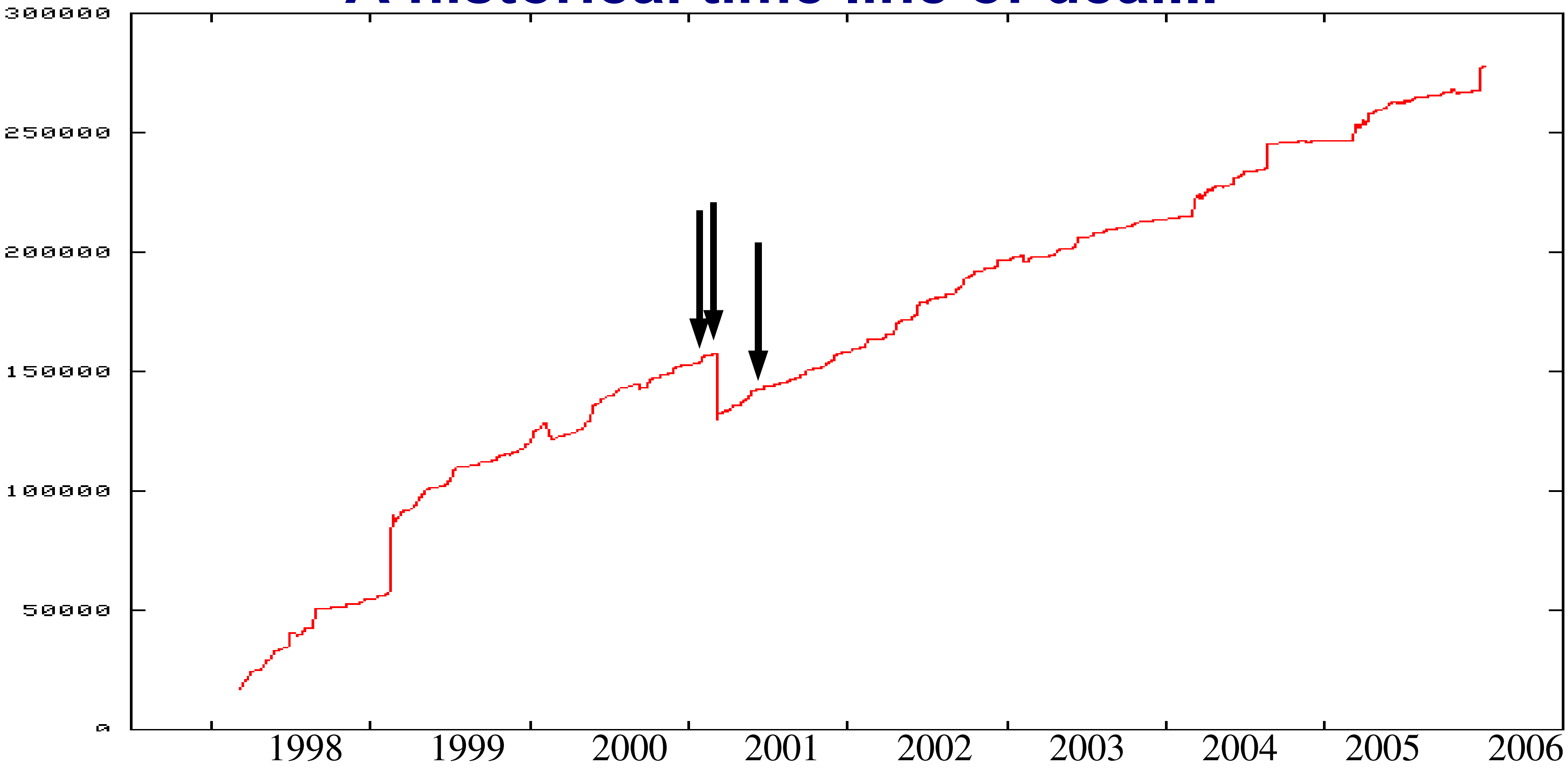
- December 1997: Start of project
- 1998/03/06: First CVS repository with already some 17,000 lines of code
- 1998/06/26: First version of bi-quadratic finite elements; cubic and quartic ones follow soon during a phase of depression
- August 1998: Guido starts to contribute code
- 1998/12/18: Ralf contributes first code

A historical time line of deal.II



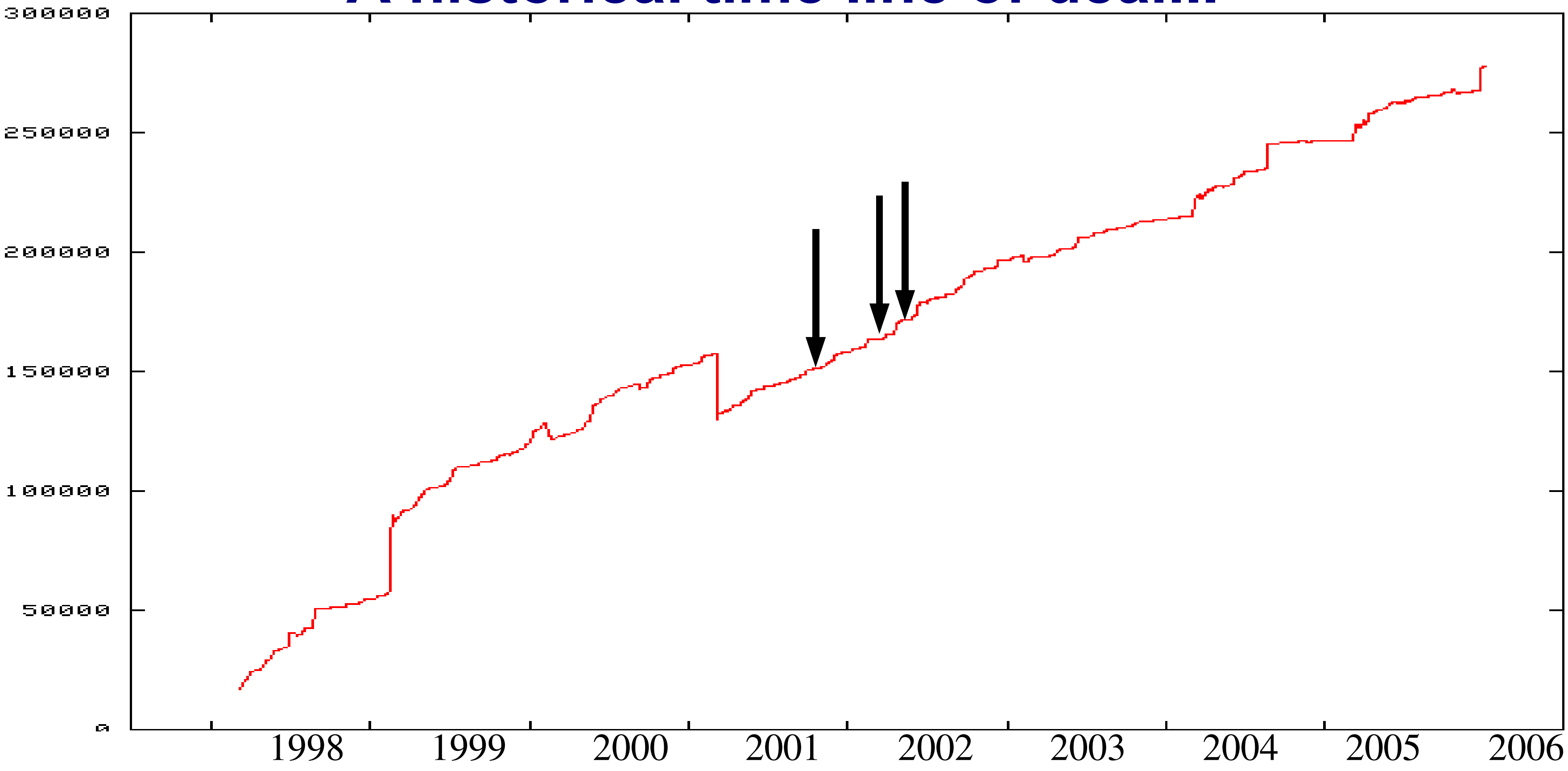
- 1999/2/16: Wolfgang merges the 3d components that had been worked on on a branch for a few weeks before. Most of the code for the shape functions is Maple-generated explicit code
- 1999/10/18: Wolfgang needs some block vectors and matrices, so adds them to the library
- 2000/4/13: First public version, deal.II 3.0, 124,000 lines of code

A historical time line of deal.II



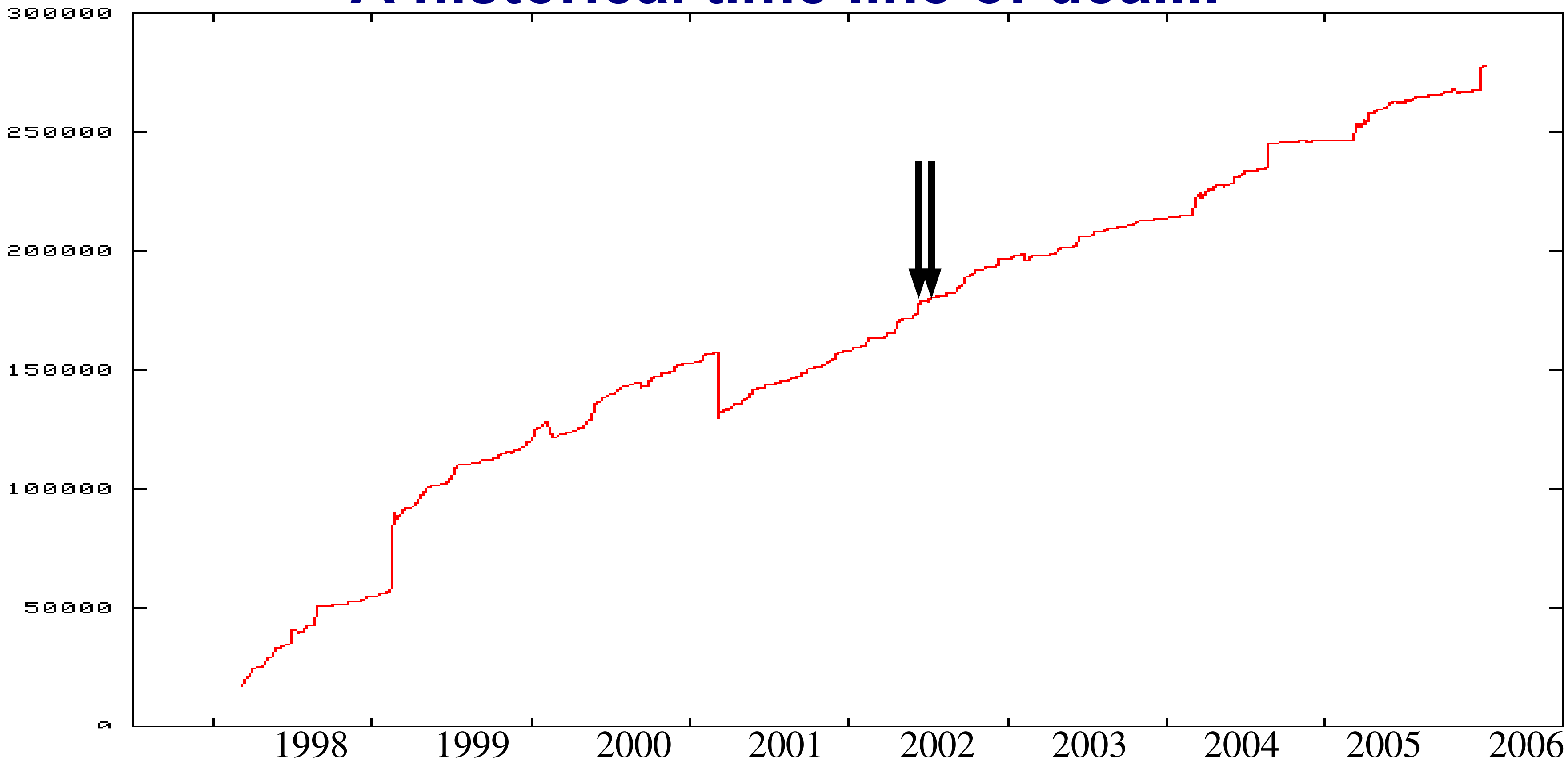
- 2001/2/20: deal.II 3.1 released, 157,000 lines of code
- 2001/3/6: Ralf replaces the old machine generated code for shape functions and their derivatives by code that uses polynomial representations. The new code is much faster, cleaner, and removes 28,000 lines of code
- 2001/5/23: First non-gcc compiler (Intel ICC compiler) supported. Compaq's CXX follows 2 months later

A historical time line of deal.II



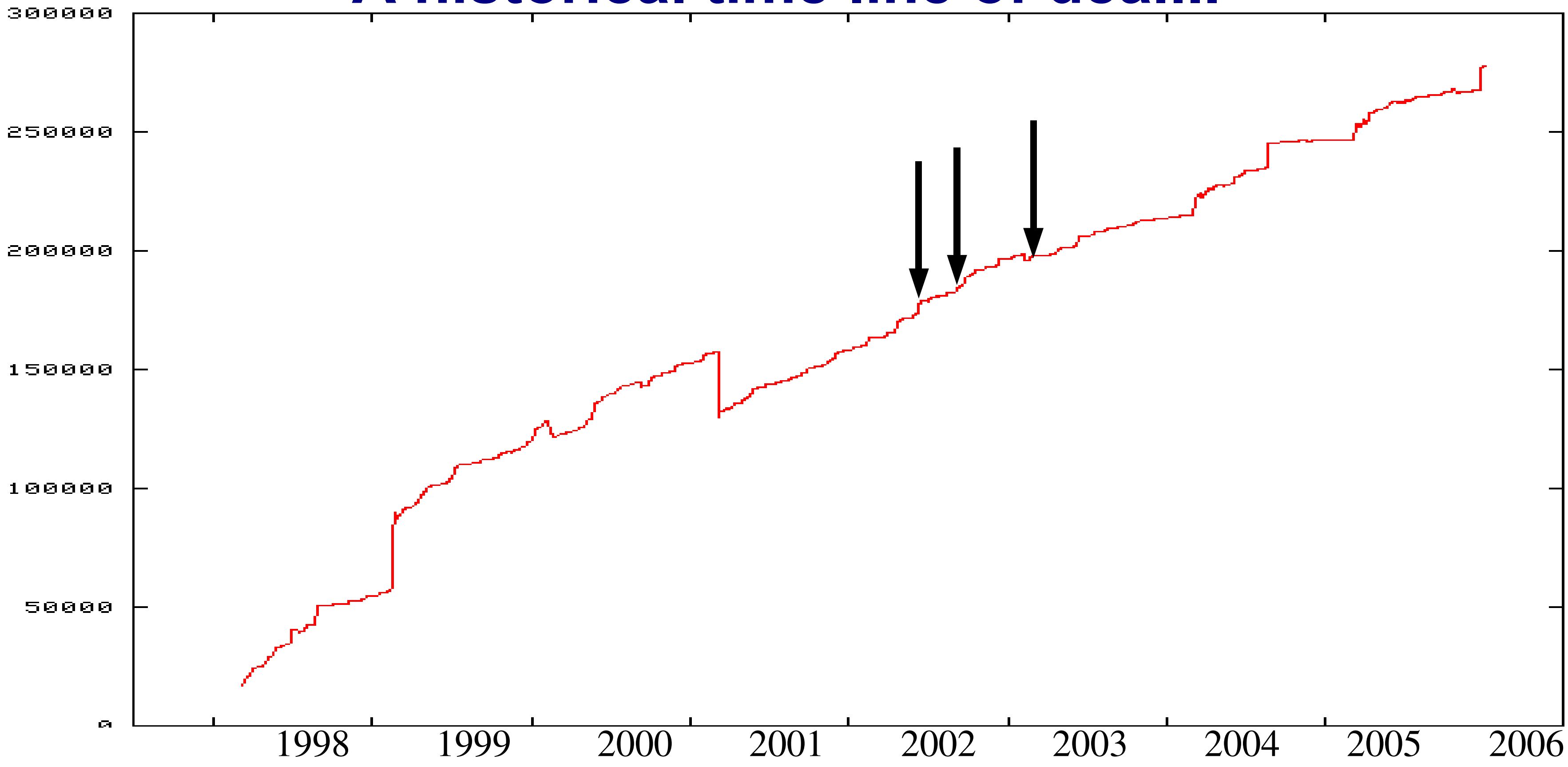
- 2001/9/28: deal.II 3.2 released, 150,000 lines of code. It contains a generalization of mappings and now supports not only Q1, but arbitrary polynomials mappings implemented by Ralf and Guido. Step-12 shows DG discretizations
- 2002/02/01: deal.II 3.3 released, 160,000 lines
- 2002/03/07: The www.dealii.org domain comes online

A historical time line of deal.II



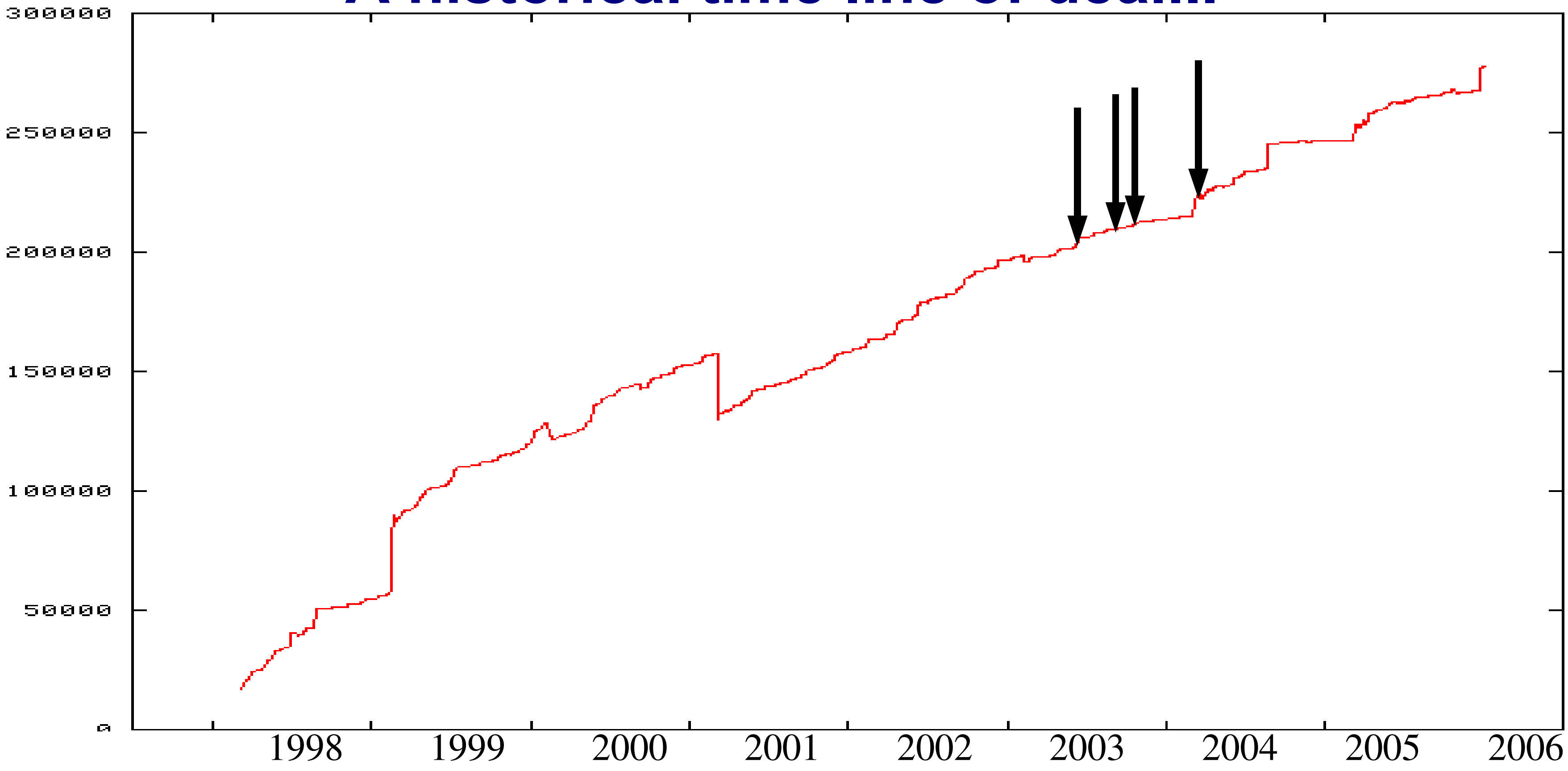
- April 2002: Wolfgang moves to Zurich, Switzerland, to support Rannacher for a guest lecturership. Step-13 and step-14 are written to show adaptive discretizations as well as error estimation techniques
- 2002/6/6: deal.II 3.4 released, 173,000 lines. New features only FE_DGP and the two example programs

A historical time line of deal.II



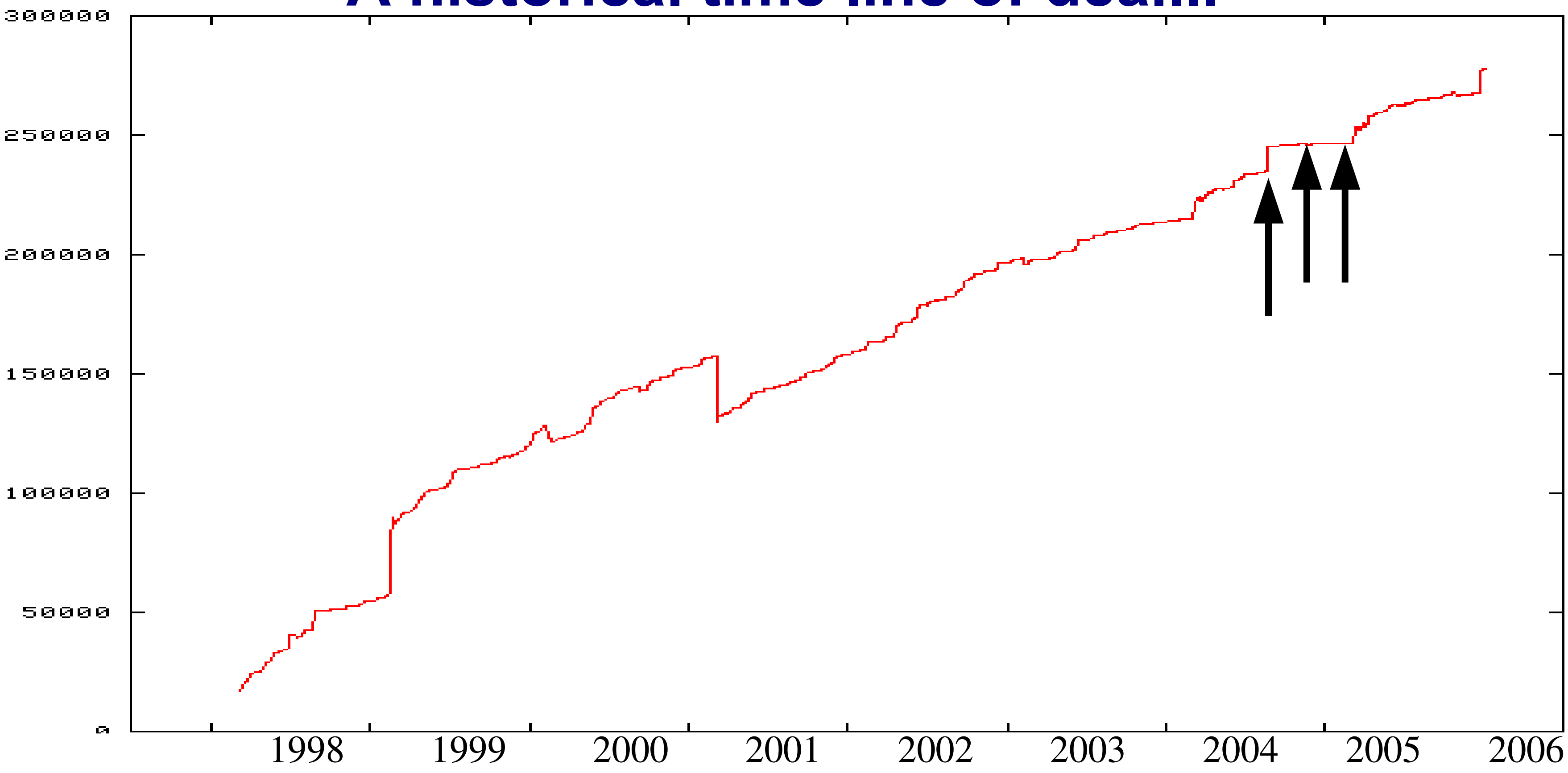
- 2002/6/10: Support for truly vector-valued elements and availability of the Nedelec element
- August 2002: Wolfgang moves to Austin, TX, for a postdoc
- 2003/2/21: Mike Anderson implements an algorithm that allows us to read and reorder the cells of complicated 2d meshes. The algorithm now has linear instead of exponential complexity.

A historical time line of deal.II



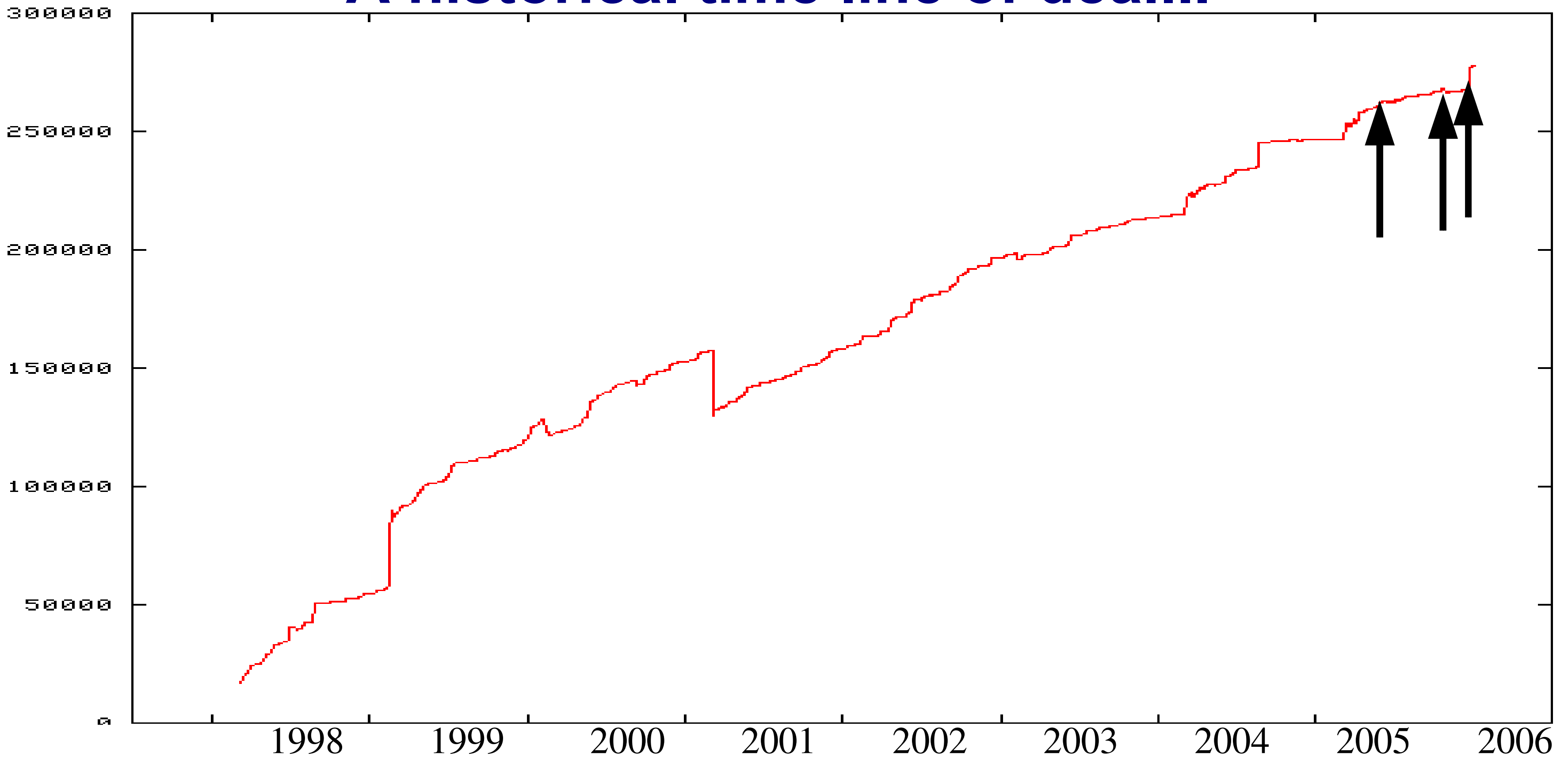
- 2003/4/27: deal.II exceeds 200,000 lines of code
- 2003/6/6: Exactly one year after version 3.4, deal.II 4.0 is released
- 2003/6/9: Raviart-Thomas elements are merged
- late 2003/early 2004: Wolfgang needs to do computations in parallel, so writes interfaces to PETSc and METIS. Step-17 shows such computations

A historical time line of deal.II



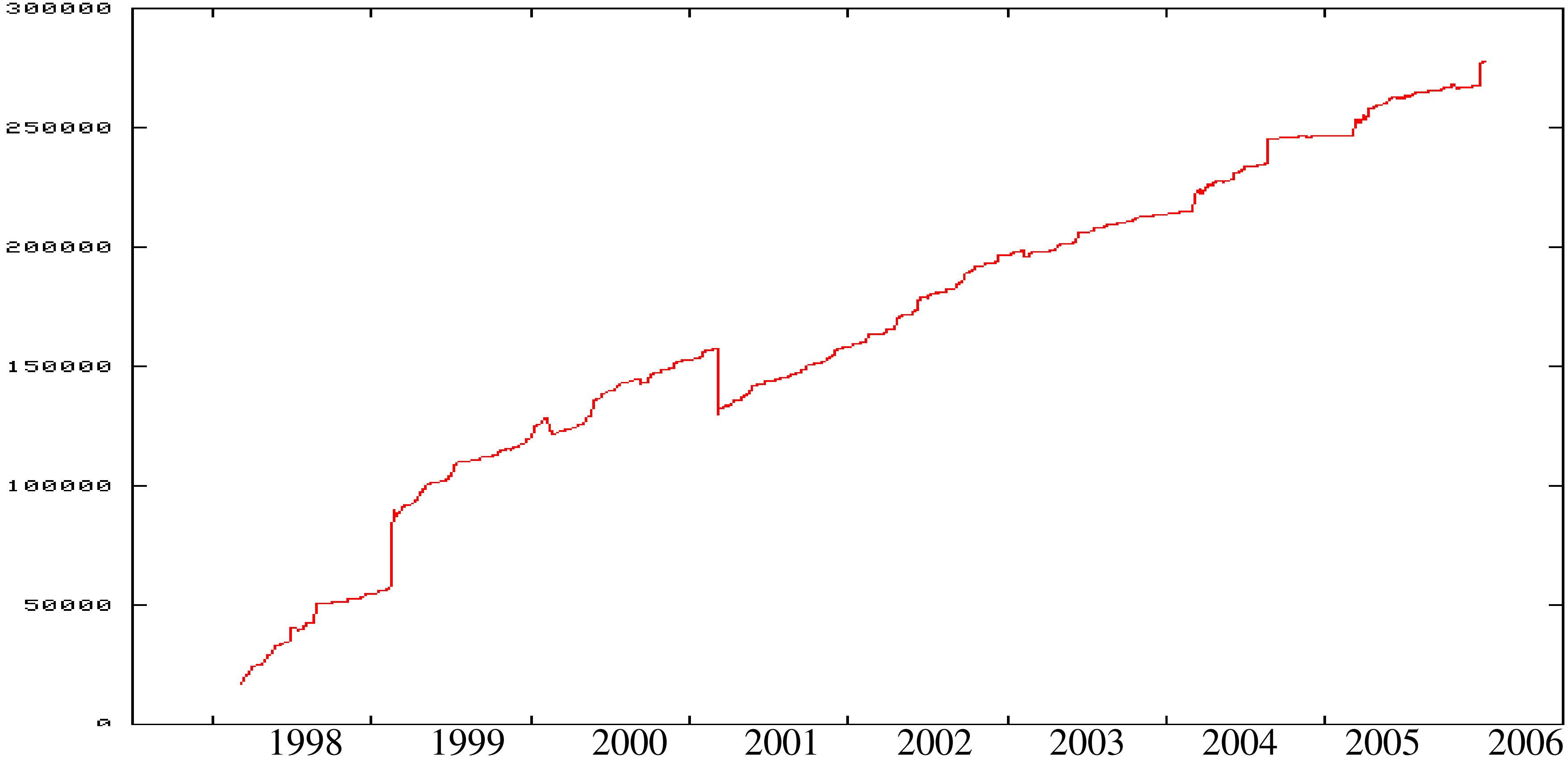
- 2004/8/21: Addition of the sparse direct solver UMFPACK
- 2004/10/21: Oliver Kayser-Herold implements code that allows to use hanging nodes in 3d for all Qp elements, not only for p=1,2
- 2004/12/24: Wolfgang applies for positions, so needs some publicity. The others agree to a new release, deal.II 5.1, now at 247,000 lines of code

A historical time line of deal.II



- Summer 2005: Two more tutorial programs, step-18 and 19 are developed, showing quasi-static elasticity and some parallel data handling stuff
- 2005/9/17: deal.II 5.2 is released
- 2005/12/20: Oliver Kayser-Herold merges the hp branch that he and Wolfgang had worked on since early 2003

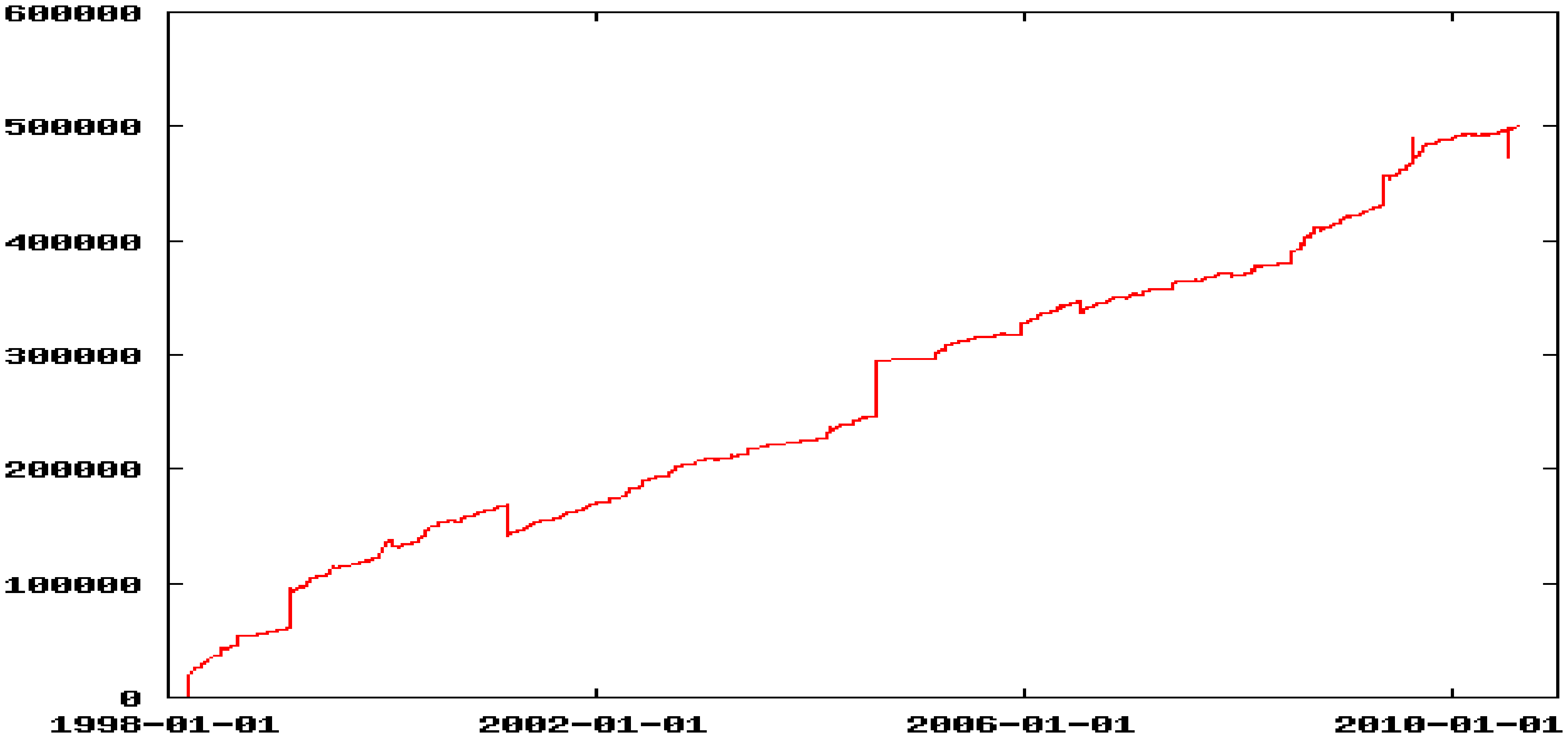
A historical time line of deal.II



~~Today:~~ Back in 2006

- deal.II has 277,418 lines of code
- some 100 downloads per month
- about 80 people on the mailing list
- about 15 people have contributed code directly
- active development of the hp functionality
- has its first User Meeting!

A historical time line of deal.II



Since 2006 and up to the present:

- deal.II has ~~277,418~~ 500,906 lines of code
- some ~~100~~ 500 downloads per month
- about ~~80~~ 260 people on the mailing list
- about ~~15~~ 40 people have contributed code directly
- active development of ~~hp~~ ~~aniso~~ ~~multigrid~~ ~~threading~~ distributed
- has its ~~first~~ second User Meeting!

www.dealii.org

Lessons learned: 1998 – 2010

Lessons on the code base (the technical side):

- deal.II has grown almost continuously at 3-4000 lines of code per month
- We have reasonably modular code:
  - a user does not need to know the internals of deal.II
  - developers do not need to know all of the library
- We have learned to write reasonable documentation for parts of deal.II through
  - doxygen modules
  - tutorial programs
- We have a "reasonable" testsuite
- We have more people who contribute code

- But the library has become a big piece of code: There is nobody any more who still knows everything

I think we have the technical side under control.

Lessons learned: 1998 – 2010

Lessons on the user base (the social side):

- We have a lot more users
  - many more help requests
- Nobody knows everything about everything any more
  - nobody can answer all questions
- We interface with many more external packages
  - much more complicated interactions
- We have a lot more actual and potential contributors
- The original developers have less time

I think we have to work on the social side!

What we're here for this week

- Hear about what deal.II is used for
- Make connections

- Hear about what's new in deal.II

- Get to know each other: people who know each other work better together
- Hatch plans for deal.II
- Figure out how to make the “social project” deal.II work better

What we're here for this week

- Hear about what deal.II is used for
- Make connections
- Hear about what's new in deal.II
- Get to know each other: people who know each other work better together
- Hatch plans for deal.II
- Figure out how to make the “social project” deal.II work better

Getting to know each other

People who know each other work better together:

- They know each other's opinions on a number of things
- They don't need to write long emails any more
- They know about resources the other person might provide
- They don't as easily misunderstand each other
- They can pro-actively adjust what they do to the other person's taste

Hatch plans

Working on common projects is more fun:

- Someone else may already have part of what you need
- Maybe we can come up with things deal.II could use and that already exist in application code
- What are the big things we want in deal.II in the future?
- Who can help implement these things?

deal.II as a “Social Project”

deal.II is a project with a “community” that consists of groups

- Library maintainers
- Subsystem maintainers
- Other developers

- Users

With every collection of groups, there are problems that we need to work on.

deal.II as a “Social Project”

For example: Library maintainers vs other developers

- How can we attract more developers?
- Are we encouraging enough in our attempts?
- Can we make the “entry bar” low enough?
- Do we give adequate credit to contributors?

- How do we ensure the long-term quality of deal.II?
  - Do even first time contributors have to write testcases?
  - How about documentation?
  - How can we enforce our coding styles?
  - Should we accept *every* contribution?

deal.II as a “Social Project”

For example: Developers vs users

- How can we organize answering mails on the mailing list?
- Can we organize it more equitably?
- Can we ensure adequate response times?
- Can we think of better ways of documenting stuff so that people can find them?

- How can we entice users to become contributors?
- How can we get users to work together on projects?

For now:

Go forth, make friends,
and proliferate the code