

Planning for the future:

**Securing continued funding
for deal.II**

Current model

deal.II development is currently funded by an NSF grant:

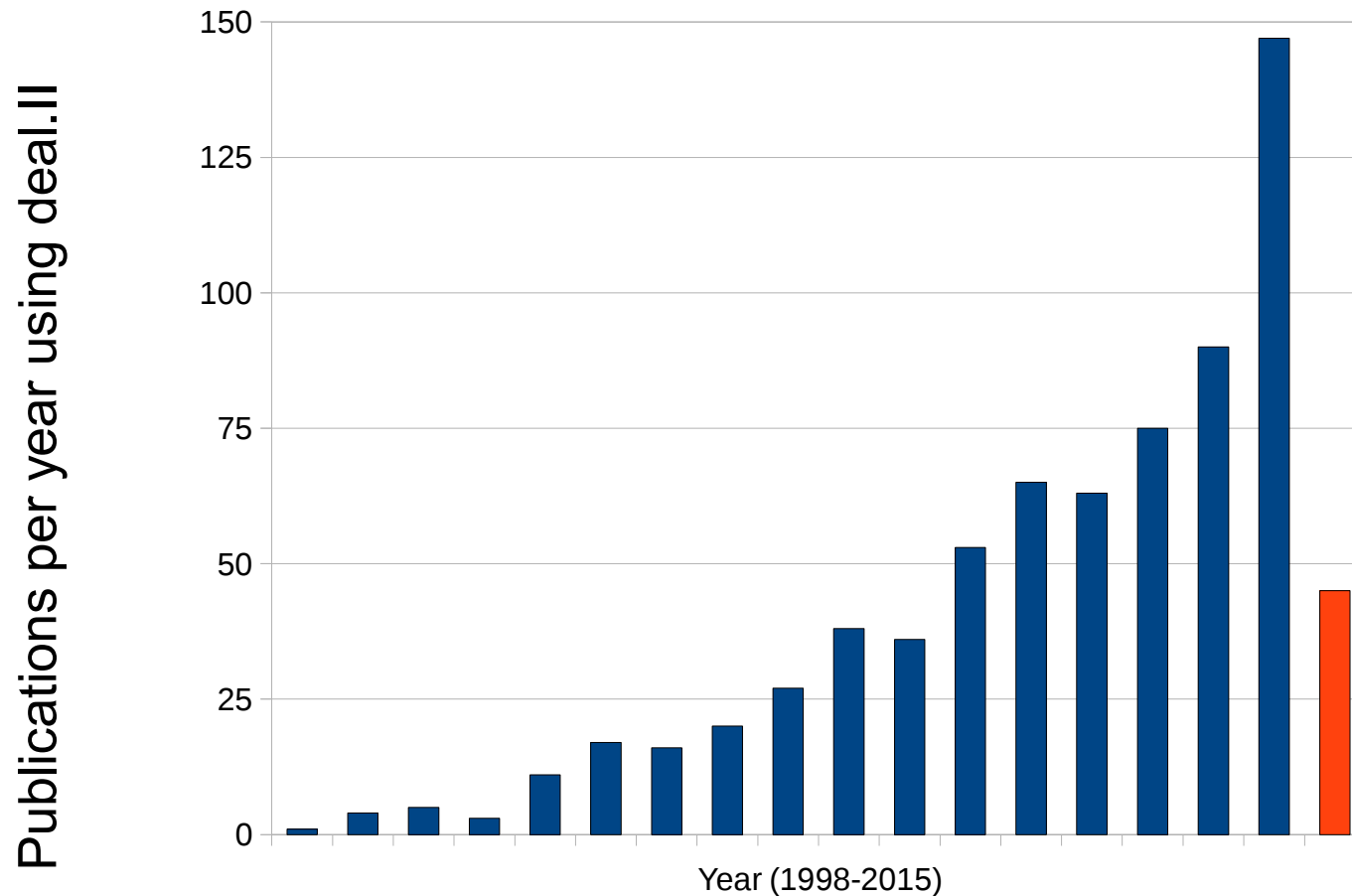
- \$1.5M for 4 years (2012-2016)
- Pays for workshops, postdocs, 2 grad students, small fraction of my salary
- Left-over money will carry us to 2017

- Follow-up proposal will have to be written in early 2016
- Need to find “compelling story”
- To propose “continued development” not compelling

My thinking on this proposal

What I think might be compelling:

- Highlight the broad applicability
- Highlight growing use:



My thinking on this proposal

What I think might be compelling:

- Highlight the broad applicability
- Highlight growing use
- Run a survey that elucidates
 - what features are currently in use
 - what features people like
 - what features people don't like
 - what features people would need in the future

Things that are there

- Many different finite elements
- h , p , hp refinement
- Many different output formats
- Support for shared memory parallelism
- Support for distributed memory parallelism
- Interfaces to many other packages
 - PETSc
 - Trilinos
 - BOOST
 - UMFPACK, ARPACK, ...
- Many tutorial programs
- Mostly cross-platform, downloadable packages
- Anisotropic refinement
- Video lectures
- Secondary effect: provide larger user base for underlying projects

What do you like?

- Ease of installation (basic installation vs complex dependencies)
- Complex features also available at large scale
- Lead time is small: simple syntax, existing starting points
- Good intro to coupled complex problems
- Documentation at the right levels
- Mailing lists are searchable, archived
- Tutorials! (To learn, as a starting point)
- ParameterHandler (who uses the GUI?)
- Chasing the cutting edge

Other ideas for the proposal

- Need to copy mission/vision into proposal
- How does DUNE do it to get multiple funding streams
- Google summer of code

What do you not like/find confusing?

- Not able to tag forum messages by topic (move to stackoverflow – does it support email interface)
- Python wrappers
- Multiphysics can not easily be described as a collection of individual physics modules
- Solve standard problems with fewer lines of code
- Higher-level assembly
- Documentation of parameters unclear
- deallog
- More complex applications (e.g., Navier-Stokes)
- No Finite Difference/Finite Volume support
-
- Support for triangles (not going to happen!)
- High level interface (not going to happen :-)

Needed tutorials

- How to contribute
- How to create a new finite element

Things one could propose

- More finite elements
- Better support for shared memory parallelism
- Better support for distributed memory parallelism
- GPUs (?)
- More tutorial programs
- Better Windows support

- Different kinds of documentation/training
- Annual workshops
- More “whole” applications

- ...?