Nuclear TALENT, perspectives and future plans

Morten Hjorth-Jensen, National Superconducting Cyclotron Laboratory and Department of Physics and Astronomy, Michigan State University, East Lansing, MI 48824, USA & Department of Physics, University of Oslo, Oslo, Norway

Nuclei workshop, MSU and FRIB, June 10-13 2015

Motivation

- Develop structured modules which will provide our students with a modern education in nuclear physics
- Modules/courses should contain a high-level of synchronization
- A computational perspective is essential
- The FRIB theory center can function as the national coordinating unit

All material at my github address, go to `https://github.com/mhjensen`, click on the seminars link.

Nuclear Talent v2.0

- Is it possible to integrate material developed in different Talent courses, offering thereby a coherent source for educating the next generation of nuclear physicists? Keyword: Modularization of topics.
- How many basic courses can an institution offer, and which courses should be offered?
- How can the coming FRIB theory center be used to coordinate an advanced training in nuclear physics?
- Can we integrate the (*ad hoc*) Nuclear Talent courses/initiative in our education?
- **Nuclear Talent initiative and Asia** . The Chinese community is very enthusiastic about the Talent initiative

Local situation at MSU

We have at MSU a

- basic survey course PHY802 and three basic nuclear physics courses
- structure,
- reactions and dynamics and
- **Nuclear Astrophysics**.

These three basic courses have a duration each of 30-40 hours (2-3 credits).
They can be taught as a regular one-semester course or half-semester course. There are also experimental courses.

Advanced modules, Nuclear Talent

1. Nuclear forces (INT 2013, new version 2017)
2. Many-body methods (**GANIL July 2015**)
3. Few-body methods for nuclear physics (**ECT* July-August 2015**)
4. Density functional theory and self-consistent methods (ECT* 2014 and **York 2016**)
5. Theory for exploring nuclear structure experiments (GANIL 2014)
6. Theory for exploring nuclear reaction experiments (GANIL 2013)
7. Nuclear theory for astrophysics (MSU 2014 and **INT 2015**)
8. Theoretical approaches to describe exotic nuclei (planned for 2016, Chalmers, Gothenburg)
9. High-performance computing and computational tools for nuclear physics
   - ECT* 2012, Shell model and variational Monte Carlo
   - LANL/ORNL in 2016, Monte Carlo methods

Talent v2.0: Scientific writing and publishing for the future

Scientific writing = LaTeX

1. Pre 1980: handwriting/typewriting + publisher
2. Post 1985: scientists write LaTeX
3. Post 2010: a few scientists explore new digital formats
4. Big late 1990s question: Will MS Word replace LaTeX? It never did!
5. LaTeX PDF is mostly suboptimal for the new devices
6. The book will survive (LaTeX is ideal)
7. The classical report/paper will survive (LaTeX is ideal)
8. But there is an explosion of new platforms for digital learning systems!
9. How to write scientific material that can be easily published through old and new media?

Can I assemble lots of different writings to a new future document (book)?

Suppose I write various types of scientific material

1. LaTeX document,
2. blog posts (HTML),
3. web pages (HTML),
4. Sphinx documents,
5. IPython notebooks,
6. wikis,
7. Markdown files, ...

and later want to collect the pieces into a larger document, maybe some book - is that at all feasible?

Popular tools anno 2014 and their math support

1. LaTeX: de facto standard for math-instensive documents
2. pdfLaTeX, XeLaTeX, LuaLaTeX: takes over (figures in png, pdf) - use these!
3. MS Word: too clicky math support and ugly fonts, but much used
4. HTML with MathJax: "full" LaTeX math, but much tagging
5. Sphinx: somewhat limited LaTeX math support, but great support for web design, and less tagged than HTML
6. reStructuredText: similar to Sphinx, but no math support, transforms to lots of formats (LaTeX, HTML, XML, Word, OpenOffice, ...)
7. Markdown: somewhat limited LaTeX math support, but minor tagging, transforms to lots of formats (LaTeX, HTML, XML, Word, OpenOffice, ...)
8. IPython notebooks: Markdown code/math, combines Python code, interactivity, and visualization, but requires all code snippets to sync together

DocOnce: one file to rule them all

DocOnce offers minimalistic typing, great flexibility wrt format, especially for scientific writing with much math and code. Developed by Hans Petter Langtangen, University of Oslo and Simula Research lab

1. Can generate LaTeX, HTML, Sphinx, Markdown, MediaWiki, Google wiki, Creole wiki, reST, plain text
2. Made for large science books and small notes
3. Targets paper and screen
4. Many special features (code snippets from files, embedded movies, admonitions, modern LaTeX layouts, extended math support for Sphinx/Markdown, ...)
5. Very effective for generating slides from ordinary text
6. Applies Mako: DocOnce text is a program (!)
7. Much like Markdown, less tagged than LaTeX, HTML, Sphinx