Software tools for research science: `xmds` and PyScript

Paul Cochrane

Australian Centre for Quantum-Atom Optics, Physics Dept.,
University of Queensland, Australia

Introduction

- Software tools in research science
- `xmds`: the eXtensible Multi-Dimensional Simulator
  - what is `xmds`?
  - who uses it?
  - why is it useful?
  - examples
- PyScript: Postscript graphics in Python
  - PyScript overview
  - uses of PyScript
  - examples
- Conclusion

Software tools in research science

- Three general phases of research
  - doing the work and getting the results
    - e.g. `xmds`, C, Fortran, Python, Matlab, . . .
  - visualising and interpreting the data
    - e.g. Matlab, Scilab, Gnuplot, . . .
  - presenting the results, and interpretation to the scientific community
    - e.g. PyScript, LaTeX, PowerPoint, . . .
- Developing tools for doing research facilitates more effective research
- Make scientists' life easier so they can focus on science, and not the "donkey-work"

What is xmds?

- eXtensible Multi-Dimensional Simulator
  - any number of components
  - any number of dimensions
  - any number of random variables
- A system for integrating differential equations:
  - one writes a high-level description of simulation in XML
  - `xmds` converts the XML into C language code
  - the C code is compiled into a binary executable, which runs about as fast as code hand-written by an expert
- `xmds` provides a way of both performing and documenting a simulation

xmds: who uses it?

- Anyone who models systems via differential equations
- This includes:
  - physicists
  - geophysicists and earth scientists
  - chemists
  - biologists
  - weather forecasters
  - economists
  - risk analysts
  - ...

Why use xmds?

- fast development time, and fast execution time
- reduces user-introduced bugs
- solves ODEs, PDEs, and stochastic ODEs and PDEs
- automatic parallelisation of stochastic and deterministic problems
- allows simple and transparent comparison of simulations with other researchers
- simulation script (and therefore parameters) are output with the simulation data, so the data and the variables that generated it are kept together for future reference
- open source and documentation, see `http://www.xmds.org/`

xmds details

- xmds is designed to integrate PDEs of the general form:

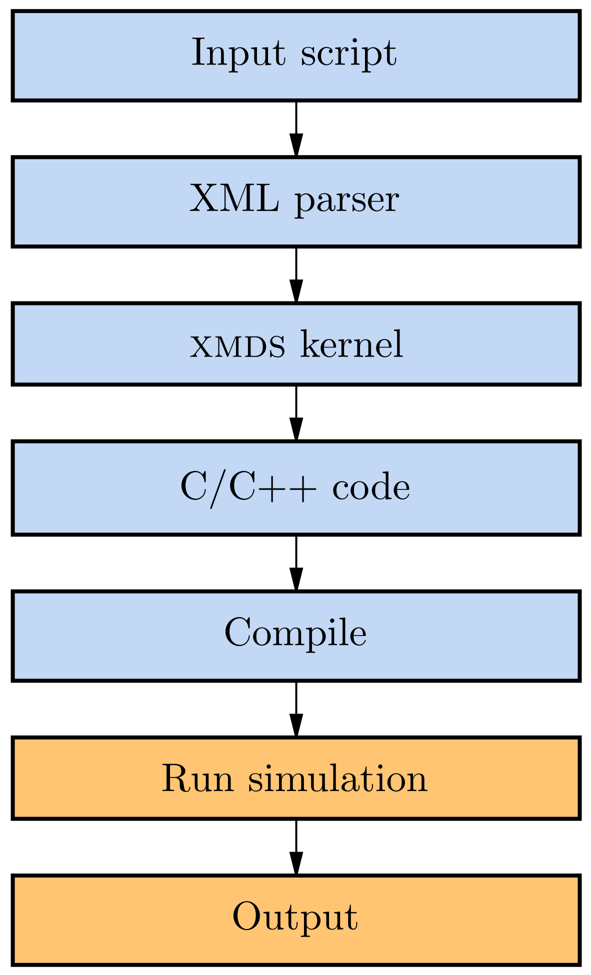
$$\frac{\partial}{\partial x^0}\mathbf{a}(\mathbf{x}) = \mathcal{N}\left(\mathbf{x}, \mathbf{a}(\mathbf{x}), \mathbf{p}(\mathbf{x}), \mathbf{b}(\mathbf{x}),\ \xi(\mathbf{x})\right),$$

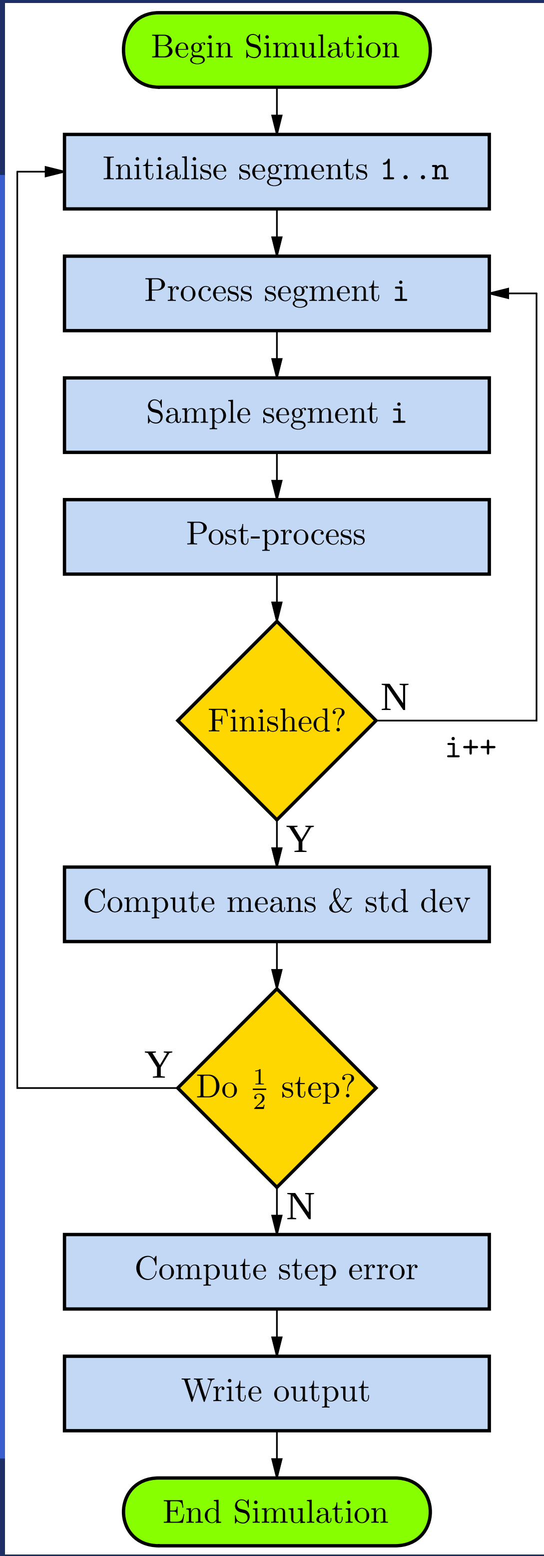
$$p^i(\mathbf{x}) = \mathcal{F}^{-1}\left[\Sigma_j \mathcal{L}^{ij}\left(x^0, \mathbf{k}_\perp\right)\mathcal{F}\left[a^j(\mathbf{x})\right]\right],$$

$$\frac{\partial}{\partial x^c}\mathbf{b}(\mathbf{x}) = \mathcal{H}\left(\mathbf{x}, \mathbf{a}(\mathbf{x}), \mathbf{b}(\mathbf{x})\right)$$

  - $\mathbf{x}$: spatial dimension
  - $\mathbf{a}(\mathbf{x})$: main field
  - $\mathbf{b}(\mathbf{x})$: cross-propagating field
  - $\mathbf{p}(\mathbf{x})$: field defined in Fourier space
  - $\xi(\mathbf{x})$: noise terms

xmds processes



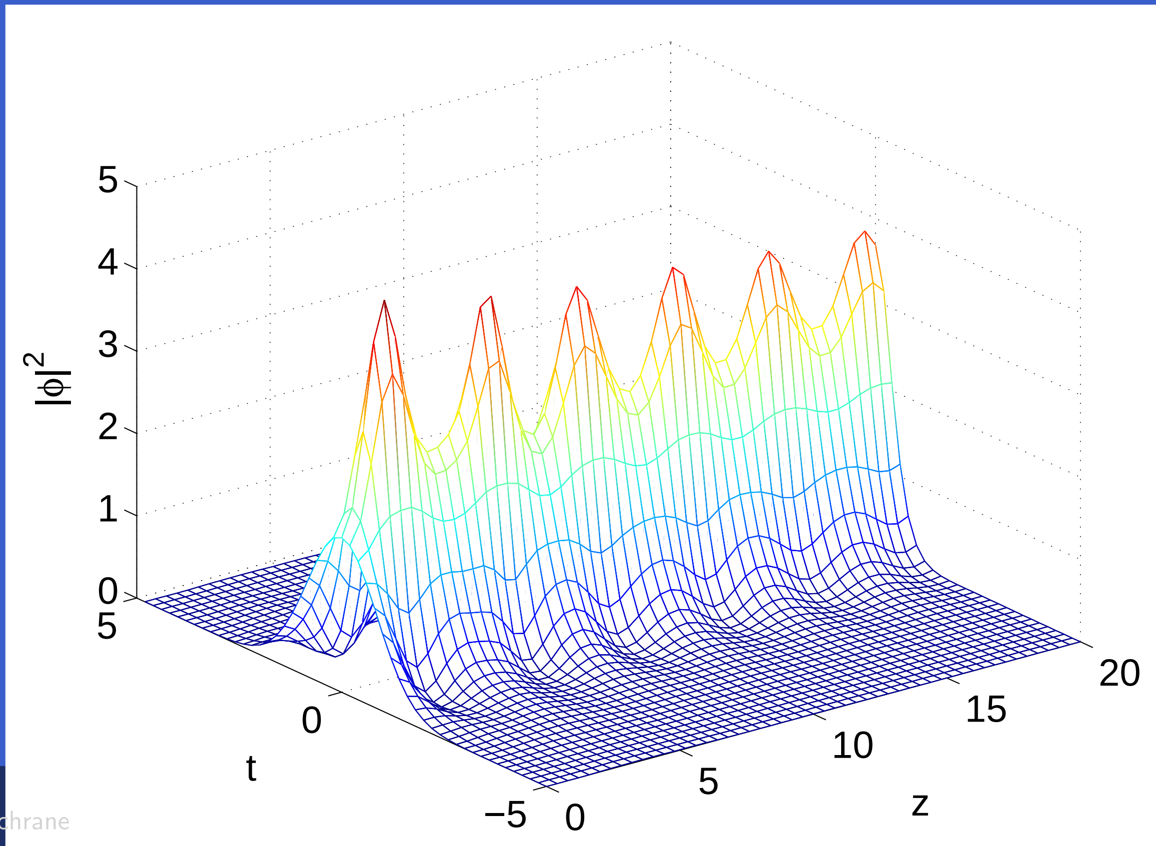


xmds examples

- Nonlinear Schrödinger Equation

$$\frac{\partial \phi}{\partial z} = i\left[\frac{1}{2}\frac{\partial^2 \phi}{\partial t^2} + |\phi|^2\phi + i\Gamma(t)\phi\right]$$

  - where ϕ is the field, z is the spatial dimension, t is time and $\Gamma(t)$ is a damping term.



xmds demonstration

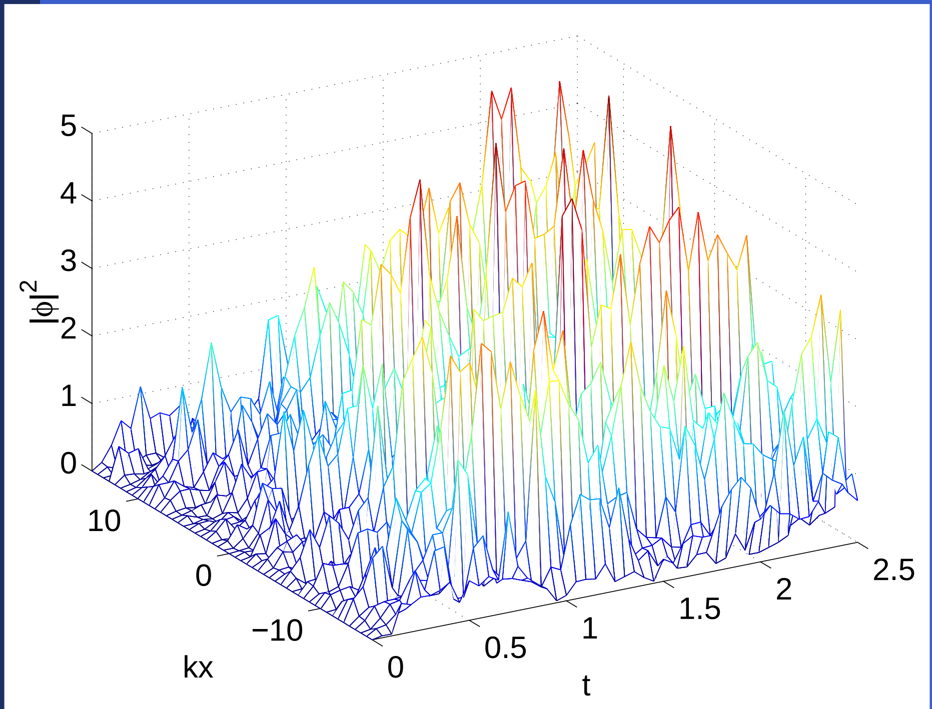
- Nonlinear Schrödinger Equation
  - You've seen the equation, you've seen the results, now see `xmds` in action

xmds examples (cont.)

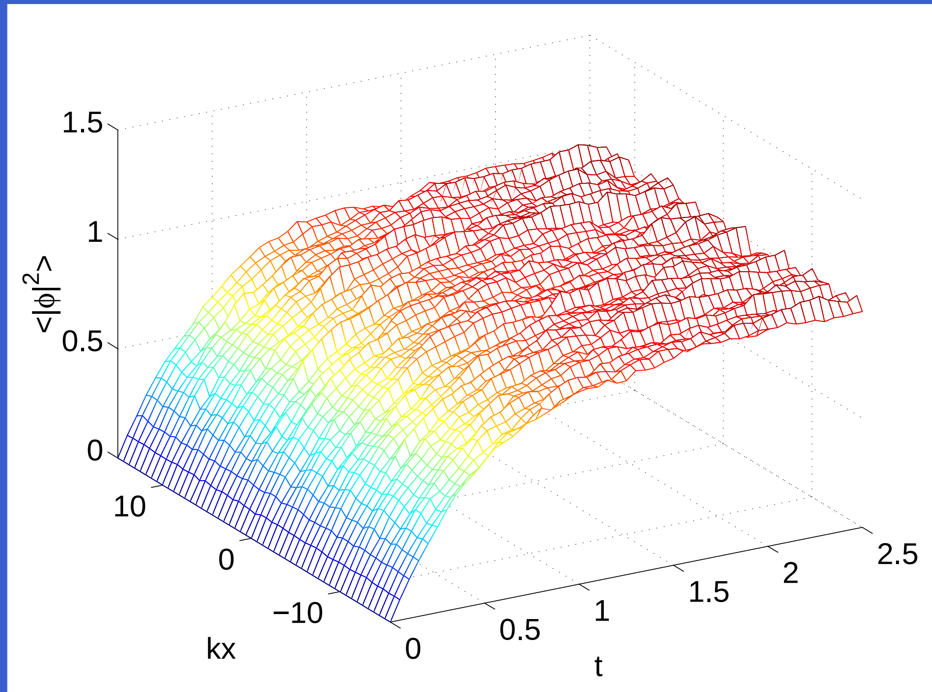
- Fibre Optic Laser Field

$$\frac{\partial \phi}{\partial t} = -i\frac{\partial^2 \phi}{\partial x^2} - \gamma\phi + \frac{\beta}{\sqrt{2}}\left[\xi_1(x,t) + i\xi_2(x,t)\right]$$

  - this is a stochastic simulation



single path



1024 path mean

Bestiary of xmds features

- Automatic numerical error checking
- Automatic parallelisation of stochastic and deterministic problems
- Handles cross-propagating fields
- Calculates trajectory means and variances of stochastic simulations
- ASCII and binary output
- Benchmarking of simulations
- `xmds` script template output
- Field initialisation from file
- Command line arguments to simulations
- User-defined preferences (for custom compiler flags etc.)
- ...

PyScript

What is PyScript?

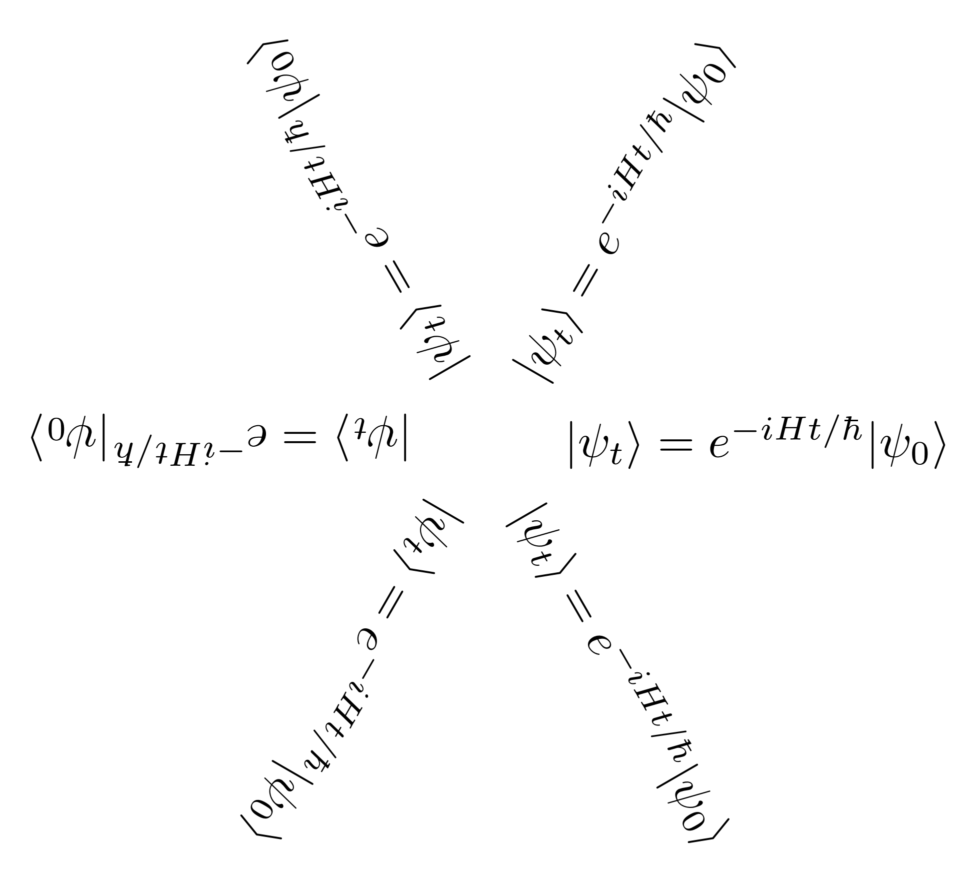
- Generate Postscript figures/diagrams/documents using the Python programming language
- Instead of using a mouse to draw objects, write code to draw objects
- Have full use of the Python language and modules
- Produces small file-size, high-quality Postscript output
- Uses LaTeX to produce high quality fonts
- Free software, released under the GPL
- `http://pyscript.sourceforge.net`

The uses and features of PyScript

- PyScript uses:
  - figures and diagrams in journal articles
  - posters and seminars for conferences
  - amaze your colleauges with the size of your Postscript files
- PyScript features:
  - scaling, rotating, translating, any affine transformation
  - embed EPS graphics
  - embed LaTeX graphics and symbols
  - properly kerned text objects
  - scriptable and accurate control of object location
  - expanding suite of libraries for common objects and tasks

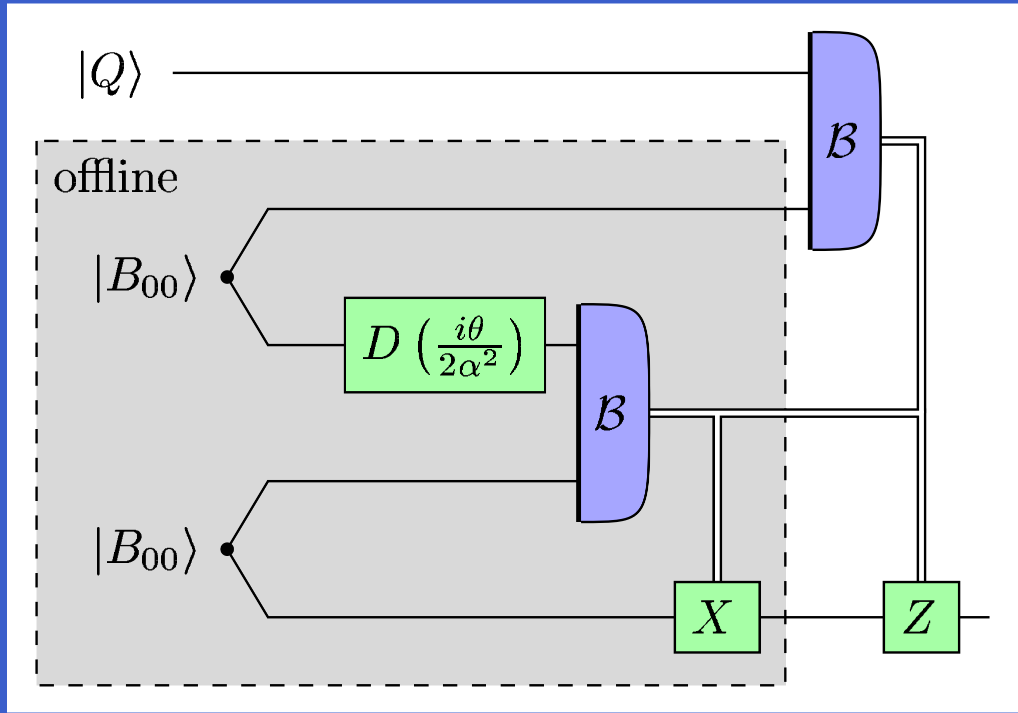
PyScript examples

```
from pyscript import *

tex=TeX(r'$|\psi_t\rangle=e^{-iHt/\hbar}|\psi_0\rangle$',w=P(.5,0))

g=Group()
for ii in range(0,360,60):
    g.append(tex.copy().rotate(ii,P(0,0)))

render(g,file=...)
```



PyScript examples (cont.)

```
offline=Rectangle(height=4,width=5.5,e=P(3.5,1.5),
                  dash='[3 ] 0',bg=Color(.85))

render(
    offline,
    TeX('offline',nw=offline.nw+P(.1,-.1)),

    Path(P(5,0),P(-.3,0),P(-.6,.5),P(-.3,1),P(2,1)),
    Path(P(2,2),P(-.3,2),P(-.6,2.5),P(-.3,3),P(3.7,3)),
    Path(P(-1,4),P(3.7,4)),

    Dot(P(-.6,.5)),
    Dot(P(-.6,2.5)),

    classicalpath(Path(P(2.1,1.5),P(4.5,1.5),P(4.5,0)),
                  Path(P(3,1.5),P(3,0)),
                  Path(P(3.8,3.5),P(4.5,3.5),P(4.5,1.5)),
                  ),

    BellDet(P(2,1.5)),
    BellDet(P(3.7,3.5)),

    Boxed(TeX(r'$D\left(\frac{i\theta}{2\alpha^2}\right)$'),
              c=P(1,2),bg=green),

    Boxed(TeX('$X$'),c=P(3,0),bg=green),
    Boxed(TeX('$Z$'),c=P(4.5,0),bg=green),

    TeX(r'$\ket{B_{00}}$',e=P(-.7,.5)),
    TeX(r'$\ket{B_{00}}$',e=P(-.7,2.5)),
    TeX(r'$\ket{Q}$',e=P(-1.1,4)),

    file=...,
    )
```



(18K—try doing that in Illustrator)

PyScript examples (cont.)

Conclusion

- `xmds` and PyScript are:
  - two software tools to aid research science
  - enabling technologies
  - able to let scientists focus on science
- `xmds` will solve your problems quickly
- PyScript will make the results look good

Acknowledgements

- xmds (http://www.xmds.org)
  - Greg Collecutt, Peter Drummond, Joe Hope
- PyScript (http://pyscript.sourceforge.net)
  - Alexei Gilchrist