



xmnds: eXtensible multi-dimensional simulator

G. Collecutt *, P.D. Drummond

The University of Queensland, St Lucia, QLD 4072, Australia

Abstract

A software package that efficiently solves a comprehensive range of problems based on coupled complex nonlinear stochastic ODEs and PDEs is outlined. Its input and output syntax is formulated as a subset of XML, thus making a step towards a standard for specifying numerical simulations. © 2001 Elsevier Science B.V. All rights reserved.

PACS: 02.60.Cb; 02.60.Lj; 02.70.-c; 02.50.Ey

Keywords: XML; Stochastic; Partial; Differential equation; C++

1. Introduction

Computational physics remains a rapidly growing field, and has been described by some as a third way of doing physics in addition to theory and experiment. In a recent review titled *Microscopic simulations in physics* [1] Ceperly writes:

Sadly, the lore of experimental and theoretical physics has not yet fully penetrated into computational physics. Before the field can advance, certain standards, which are commonplace in other technical areas, need to be adopted so that people and codes can work together.

We present here such a standard, written in a precise form using the international **xml** language for data exchange. The corresponding parser, **xmnds** in turn writes a low level computer program that a C++ compiler can compile and optimize. The executable file produced solves the problem about as quickly and efficiently as is possible, while the input script is concise and transparent. The result of this work is an open source software package, which is available at the following URL: <http://www.physics.uq.edu.au/xmnds>

* Corresponding author.

E-mail address: collecug@physics.uq.edu.au (G. Collecutt).

2. Functionality

xmds is a software package designed to write a program that manipulates an n -dimensional m -component field, primarily using a given set of coupled nonlinear ODEs and PDEs (possibly with stochastic noise terms) to propagate this field along a dimension of propagation. The equations have the general form of Eq. (1).

$$\frac{\partial}{\partial t} a_i(t, \vec{x}) = \mathcal{L}(a_i, \vec{k}) + \mathcal{V}(\vec{a}, t, \vec{x}, \vec{\xi}). \quad (1)$$

Here the vector $\vec{a}$ represents the m -component field, though for some problems the field may only have one component. The propagation dimension has been labeled t , and the vector $\vec{x}$ represents the n -dimensional transverse space — though if you wish you may specify a field of zero transverse dimensionality to simulate an ordinary differential equation (ODE). The term $\vec{k}$ represents coordinates in Fourier space. As an example, if the field was a vector field representing three-dimensional fluid flow then there would be three components to the field, which itself would exist in a three-dimensional transverse space, and propagate forward into time as a fourth dimension.

The $\mathcal{L}()$ term on the RHS represents local, linear, and diagonal functions of the field components in $\vec{k}$ -space. Products of the Fourier domain transverse dimensions, k_j , may be used here to represent spatial derivatives of arbitrary order. The $\mathcal{V}()$ term represents local, possibly non-diagonal, functions of the field components in $\vec{x}$ -space, and can include additive or multiplicative stochastic terms. The forward evolution of the field is performed using a split step semi-implicit method [2,3] in which derivative terms in the PDEs are evaluated in the Fourier domain using the $\mathcal{L}()$ term. The $\mathcal{V}()$ terms are evaluated in normal space semi-implicitly. The Fourier transforms employ the FFTW libraries [4]. Gaussian stochastic noise terms are generated with the correct variances, using the Box–Mueller technique combined with *drand48()*.

There are two main processes that **xmds** can apply to a field: forward evolve (integrate/propagate) it according to a set of PDEs, and reshape (filter) it according to a set of functions. These two actions may serve as the building blocks for a sequence of operations to be performed on the field. The key to **xmds** is that it is a *code generator*. **xmds** requires the user to write their particular PDEs (or ODEs) as a few lines of C code, which are transplanted from the input script to the relevant points in the output code. This technique generates efficient and flexible code. To save on memory and also size of output file we may sample on some reduced lattice rather than at every point in the main field lattice. We can choose which transverse dimensions to transform to Fourier space when sampling for a particular group of moments.

We are now in a position to draw diagrammatically the functionality of **xmds**, including the steps taken to estimate both the time-discretization and sampling errors. This is shown in Fig. 1.

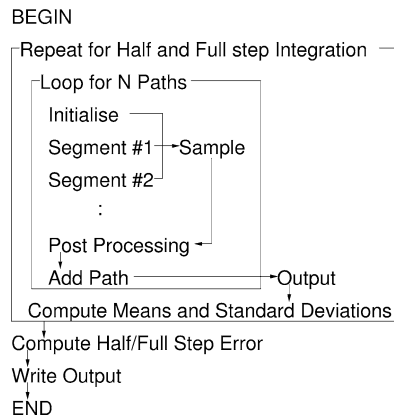


Fig. 1. **xmds** — a functional diagram.

3. Examples

XML is very much like HTML. However, even without any experience in HTML, it is simple to learn XML with the help of examples. A simple, one-component case is treated here. More complex cases with multiple component fields are available for down-load from the referenced web-site URL.

The following script describes to **xmids** how to write a program that solves the Nonlinear Schroedinger equation in one dimension. Eq. (2) describes a common soliton-formation process:

$$\frac{\partial \phi}{\partial t} = i \left[\frac{1}{2} \frac{\partial^2}{\partial x^2} + |\phi|^2 + i\gamma(x) \right] \phi. \quad (2)$$

Here ϕ is the single component complex field (**phi** in the input file), t is the propagation dimension, x is the local transverse coordinate. The term $\gamma(x)$ leads to linear absorption, and is used to apodize or absorb field components reaching the periodic boundaries, which would otherwise cause aliasing errors or wrap-around. We will consider what happens above the soliton formation threshold. The XMDS script is as follows:

```
<?xml version="1.0"?>
<simulation>
  <prop_dim>          t      </prop_dim>
  <field> <dimensions>    x      </dimensions>
    <lattice>          180    </lattice>
    <domains>         (-5,5) </domains>
    <components>    phi    </components>    </field>
  <sequence>
    <initial>
      <fourier_space>no</fourier_space>
      <samples> 1 </samples>
      <![CDATA[ phi = rcomplex(1.4*exp(-x*x/2.88),0);]]>
    </initial>
    <integrate>
      <interval>      10      </interval>
      <lattice>      1000    </lattice>
      <samples>       50      </samples>
      <k_propagation>
        <![CDATA[dphidt = rcomplex(0,-kx*kx/2);]]>
      </k_propagation>
      <x_propagation>
        <iterations> 3      </iterations>
        <![CDATA[ dphidt = i*conj(phi)*phi*phi
          - (1-exp(-pow(x*x/4/4,5)))*phi;]]>
      </x_propagation>    </integrate>    </sequence>
  <output><group><sampling>
    <fourier_space> no </fourier_space>
    <lattice>        60 </lattice>
    <moments>pow_dens</moments>
    <![CDATA[ pow_dens = conj(phi)*phi;]]>
    </sampling> </group> </output>
</simulation>
```

The corresponding output is in Fig. 2.

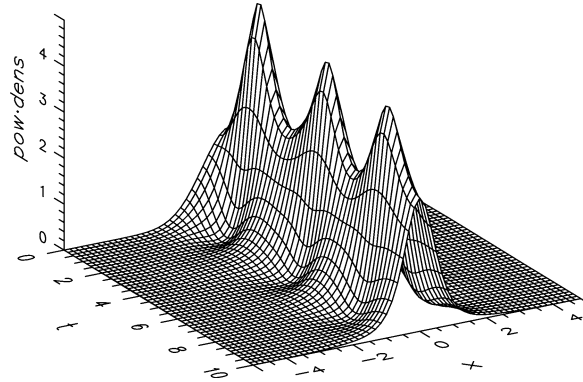


Fig. 2. Results for nlse.xmids.

xmids can also perform stochastic or Monte-Carlo style simulations. For example, Eq. (3) describes a one-dimensional damped field which is subject to a complex noise.

$$\frac{\partial \psi}{\partial t} = -i \frac{\partial^2 \psi}{\partial x^2} - \gamma \psi + \frac{\beta}{\sqrt{2}} (\xi_1(x, t) + i \xi_2(x, t)). \quad (3)$$

This stochastic PDE has an analytic solution, as shown in Eq. (4).

$$\langle |\psi(k, t)|^2 \rangle = e^{-2\gamma t} |\psi_0(k)|^2 + \frac{\beta^2 L_x}{4\pi\gamma} (1 - e^{-2\gamma t}), \quad (4)$$

where L_x is the length of the x domain.

Fig. 3 displays the results of this simulation in Fourier space including an average over 1024 trajectories, which agrees with the expected results within the sampling error. The sampling error is also computed by the program, using the central-limit theorem and the actual variance to estimate the variance in the mean, from $\sigma_{\bar{x}}^2 = \sigma_x^2 / (N - 1)$, for N stochastic trajectories.

We have not included the simulation scripts here as they are very similar to the `nlse.xmids` script above, and are included as examples supplied with the **xmids** package. Finally, multiple path stochastic problems such as this may be parallelized using MPI routines, and **xmids** is capable of writing output code incorporating MPI routines to distribute the paths over multiple processors.

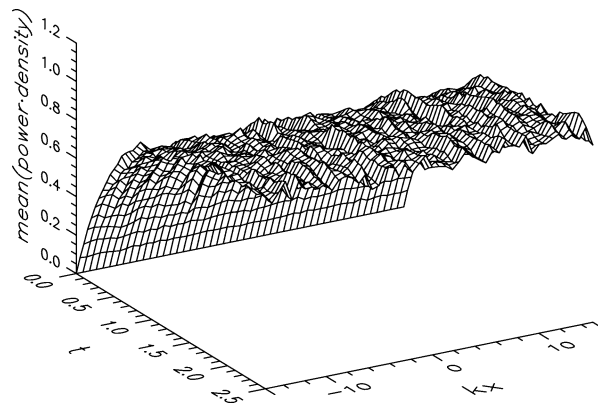


Fig. 3. Results for laser.xmids: 1024 path mean.

4. Summary

We have presented a brief introduction to a flexible approach for efficient numerical simulation of a comprehensive range of problems based on complex nonlinear stochastic ODEs and PDEs. A benefit of the technique is the use of arbitrary implied transverse dimensions. This allows the dimensionality of a problem to be changed easily, with greatly reduced likelihood of coding errors. The simulation source, being written as a high level script in XML, retains transparency, while the low level code generated is highly efficient at solving the problem. This approach encourages collaboration in research through exchange of simulation scripts, and assists reproducibility in the rapidly growing field of computational physics. The strategy used here can be used in any field requiring standardized numerical modeling.

References

- [1] D.M. Ceperly, *Rev. Mod. Phys.* 71 (1999) 438–443.
- [2] P.D. Drummond, *Comput. Phys. Commun.* 29 (1983) 211–225;
P.D. Drummond, I.K. Mortimer, *J. Comp. Phys.* 93 (1991) 144–170;
M.J. Werner, P.D. Drummond, *J. Comp. Phys.* 132 (1997) 312–326.
- [3] G.P. Agrawal, *Nonlinear Fiber Optics*, Academic Press, San Diego, 1995.
- [4] <http://www.fft.w.org/>.