How to be a Computational Physicist

after you've taken all the classes

Travis Hoppe

Drexel University

March 5, 2011

(milage may vary)

Caveats ...

- This is a personal list
- This is not comprehensive (backups? reg-ex? debugging? database management?)
- This is a non-essential list

1 How to present

2 How to learn

3 How to ask for help

4 How to manage

5 How to navigate

6 Putting it together

How to present

LaTeX

$\LaTeX$

- LaTeX is NOT WYSIWYG
- LaTeX forces you to focus on content
- LaTeX is Portable
- LaTeX is Pretty!
- LaTeX is Free!

Typesetting - Kerning

Table

Table

Typesetting - Line breaks and hyphenations

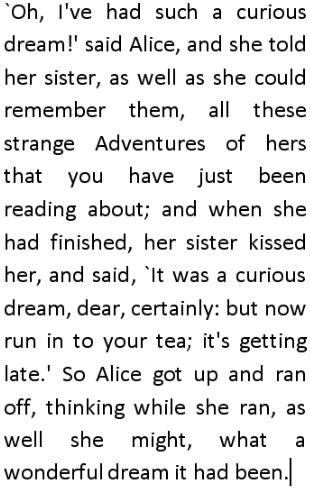

`Oh, I've had such a curious
dream!' said Alice, and she told
her sister, as well as she could
remember them, all these
strange Adventures of hers
that you have just been
reading about; and when she
had finished, her sister kissed
her, and said, `It was a curious
dream, dear, certainly: but now
run in to your tea; it's getting
late.' So Alice got up and ran
off, thinking while she ran, as
well she might, what a
wonderful dream it had been.

Figure: Microsoft Office

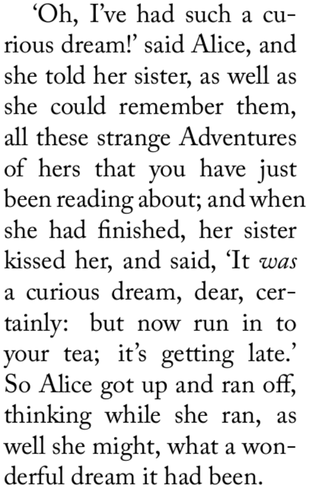

‘Oh, I’ve had such a cu-
rious dream!’ said Alice, and
she told her sister, as well as
she could remember them,
all these strange Adventures
of hers that you have just
been reading about; and when
she had finished, her sister
kissed her, and said, ‘It *was*
a curious dream, dear, cer-
tainly: but now run in to
your tea; it’s getting late.’
So Alice got up and ran off,
thinking while she ran, as
well she might, what a won-
derful dream it had been.

Figure: $\LaTeX$

Extensible as markup

Wang-Landau Density of States Calculation in Crowded Protein Environments

Travis Hoppe, Jian-Min Yuan
Drexel University
hoppe@drexel.edu

Tweet: **#hoppeSFposter**

Introduction

The Wang-Landau scheme, a recent development in the calculation of thermodynamic quantities, is used to sample the entire density of states efficiently for a biological system. We show examples of lattice proteins in crowded environments and the calculation of a two parameter density of states using a multi-pass algorithm.

Model

Wang-Landau

Wang-Landau sampling [4] is a generic algorithm to calculate Ω, the density of states, of a given system. Once Ω has been calculated to the desired accuracy, all thermodynamic quantities follow from it as $S = k \ln \Omega$. The WL algorithm attempts to find the Ω iteratively, by generating a flat histogram in energy space. Given a moveset, the WL acceptance rate from configuration A to B is:

$$P(A \to B) = \min \left(1, \frac{\Omega(E_A)}{\Omega(E_B)} \frac{n_{B \to A}/n_B}{n_{A \to B}/n_A} \right) \tag{1}$$

Where the factor $\left(\frac{n_{B \to A}/n_B}{n_{A \to B}/n_A} \right)$ accounts for movesets that are not reversible in one step [5] and $\Omega(E)$ is the currently measured density of states at energy E. As the process iterates, the density of states is updated as $\Omega(A) \to \Omega(A) f$ where f is a modification factor that monotonically decreases during the simulation.

Lattice Peptides

We use a coarse grained model; proteins are represented as a bead-sequence of amino acid residues and restricted to move on a lattice.

- A length N: Number of amino acids.
- A conformation represented by an ordered sequence $S_i \in \{1, \ldots, 20\}$, $i = 1 \ldots N$ of amino acids where each amino acid type is indexed by an integer corresponding to the twenty natural amino acids found in proteins.
- A conformation $c_i \in C_N$ denoting the location of each residue on the lattice.

Movesets

We use two movesets, pull moves [3] and bond-rebridging [1]. The combination of these moves has the advantage of small energy changes per move, which allows WL to focus down on the rare low-energy configurations. The bond-rebridging allows the system to sample the low-energy states in a more efficient manner.

Mean-Field Crowding

Crowding is implemented by modeling the physical process of excluded volume [2]. As the peptide folds we imagine the environment to favor compact conformations so we extend the physical volume of the protein by k lattice units ($k = 1$ in this study). The parameter Π modifies the acceptance rate by a factor of $(1 - \Pi)^{-v(C)}$ where $v(C)$ is the excluded volume of the protein.

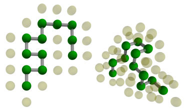


Figure 1: *Examples of 2D and 3D lattice proteins. The excluded volume zone is shown schematically with transparent yellow spheres.*

Multi-pass Wang-Landau

Once Ω has been calculated for a suitable parameter (usually energy), the density of states can be reused to estimate other observables. Since a converged WL algorithm will visit each state with equal probability, we can sample a different observable energy without having to worry about reaching all low-energy states. This approach is particularly well-suited if a correlation exists between E and the observable. Results are shown for $\Omega(E, R_g)$, the two parameter density of states for energy and radius of gyration.

Results: Crowding effects

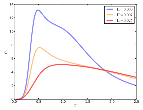


Figure 2: *Specific Heat Capacity curves for a 3D lattice homopolymer with $L = 36$ at various crowding parameters. When the crowding effect is absent there are two melting points, one for the collapse from an unfolded chain to molten globule, then to the native state. As the environment becomes more crowded, the first transition becomes less pronounced.*

Results: Crowding Contours

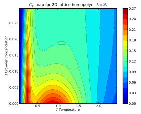


Figure 3: *Specific Heat Capacity curves for a 2D lattice homopolymer $L = 36$ at various crowding parameters shown as a contour map. A horizontal trace across the graph is the specific heat at constant Π. The effect of higher crowder concentration (as seen in the 3D case) is also observed here. Having the complete density of states allows smooth curves to be shown at any temperature without further calculation.*

Future work

Recently, we have introduced realistic potential energy functions to better capture the actual interactions between amino acid residues (4-bead and MJ matrix (not shown)). The penultimate goal is to study protein aggregation; ultimately we would like to calculate the full density of states of two or more lattice proteins and their folding cooperativity in the presence of crowders.

References

[1] J. M. Deutsch. Long range moves for high density polymer simulations. *J. Chem. Phys.*, 106(21):8849–8854, June 1997.

[2] T. Hoppe and J. Yuan. Entropic flows, crowding effects, and stability of asymmetric proteins. *Phys. Rev. E*, 80(1):011404, July 2009.

[3] N. Lesh, M. Mitzenmacher, and S. Whitesides. A complete and effective move set for simplified protein folding. In *Proceedings of the seventh annual international conference on Research in computational molecular biology*, pages 188–195, Berlin, Germany, 2003. ACM.

[4] F. Wang and D. P. Landau. Determining the density of states for classical statistical models: A random walk algorithm to produce a flat histogram. *Physical Review E*, 64(5):056101, Oct. 2001.

[5] T. Wust and D. P. Landau. Versatile approach to access the low temperature thermodynamics of lattice polymers and proteins. *Phys. Rev. Lett.*, 102(17):178101–4, May 2009.

Two Parameter Density of States

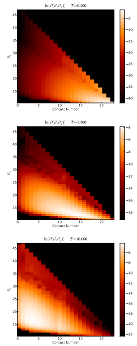


Figure 4: *Probability $P(E', R_g') = \frac{\Omega(E', R_G') \exp(-E'\beta)}{\sum \Omega(E, R_G) \exp(-E\beta)}$ of an uncrowded 3D lattice homopolymer $L = 36$ using the density of states $\Omega(E, R_g)$, the energy and the radius of gyration. $\Omega(E)$ was calculated using the standard WL algorithm, then a second pass was used to determine the $\Omega(E, R_g)$. The approach ensures that the rare low energy states are well sampled. To illustrate the shift from low-temperature compact conformations to high-temperature unfolded conformations $\ln P$ is shown, however a free energy map $\Delta G(T, E, R_G) = -kT \ln(Z)$ could have been used as well.*

Sample LaTeX code

Adapted from :
`http://amath.colorado.edu/documentation/LaTeX/basics/example.html`

```
\documentclass[12pt]{article}
\title{My Sample \LaTeX{} Document}
\author{Travis Hoppe}
\begin{document}
\maketitle                        % automatic title!

This is (very) short primer to LaTeX

\section{Formulae; inline vs. displayed}

I insert an inline formula by surrounding it with a pair of
single \$ symbols;  what is $x = 3 \times 5$?
For a \emph{displayed} formula, use double-\$
before and after --- include no blank lines!
$$\mu^{\alpha+3} + (\alpha^{\beta}+\theta_{\gamma}+\delta+\zeta)$$

Use the \emph{equation} environment to get numbered formulae, e.g.,
\begin{equation}
    y_{i+1} = x_{i}^{2n} - \sqrt{5}x_{i-1}^{n} + \sqrt{x_{i-2}^7} -1
\end{equation}

\begin{equation}
    \frac{\partial u}{\partial t} + \nabla^{4}u + \nabla^{2}u +
        \frac12    |\nabla u|^{2}~ =~ c^2
\end{equation}
\end{document}             % End of document.
```

Sample LaTeX code: output

My Sample LaTeX Document

Travis Hoppe

March 11, 2010

This is (very) short primer to LaTeX

1 Formulae; inline vs. displayed

I insert an inline formula by surrounding it with a pair of single $ symbols; what is $x = 3 \times 5$? For a *displayed* formula, use double-$ before and after — include no blank lines!

$$\mu^{\alpha+3} + (\alpha^{\beta} + \theta_{\gamma} + \delta + \zeta)$$

Use the *equation* environment to get numbered formulae, e.g.,

$$y_{i+1} = x_i^{2n} - \sqrt{5}x_{i-1}^{n} + \sqrt{x_{i-2}^{7}} - 1 \tag{1}$$

$$\frac{\partial u}{\partial t} + \nabla^4 u + \nabla^2 u + \frac{1}{2}|\nabla u|^2 \ = \ c^2 \tag{2}$$

How to learn

Project Euler

Problem: How does one learn a new programing language?

Often we are faced with learning a new programming language (or some new library from a known one).

- Textbooks: Will give a good detailed instruction to the language. Often too comprehensive - will detail many features irrelevant to you
- Verbal instruction (that's me!): Allows immediate feedback with questions. Less of a hands-on approach, the information is given but rarely self-analyzed
- Practice : Programing is as much of an art as it is a science - by far best way to learn

Choosing good material is hard - after all you don't know the language in the first place!

Project Euler

Time and time again, you will encounter the same problems in a different setting.

- Project Euler is a great collection of problems that require three elements needed for physics, mathematics, critical thinking and tight coding practices.
- Once you solve a problem you are given access to a forum where you can see answers from other users. These answers span the gamut of languages from C++, Python, Ruby, Perl, Assembly, Scheme, Delphi, etc...
- Often your solution is not as clever as others - use them to learn!

How to ask for help

Stack Overflow

Where to go when Google fails?

- The typical response to a dumb (not naive) question is to Google it (JFGI).
- There are times when Google fails - this is often because you don't know how to ask the right question in the first place!
- Similar to looking up how to spell a word in a dictionary when you don't know how to spell it!

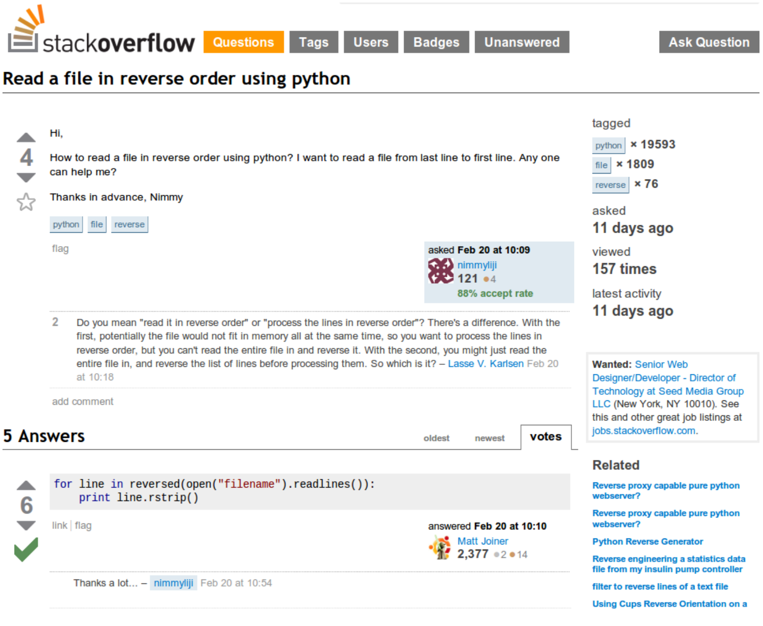
Stack Overflow

When Google fails and Wikipedia is not specific enough - turn to the most helpful programming message board created:

stackoverflow

The site leverages the communication of a form, while encouraging participation through points. In short, it's the ultimate nerd video game.

Stack Overflow: How it works



Read a file in reverse order using python

4

Hi,

How to read a file in reverse order using python? I want to read a file from last line to first line. Any one can help me?

Thanks in advance, Nimmy

python file reverse

flag

asked **Feb 20 at 10:09**
nimmyliji
121 ●4
88% accept rate

2 Do you mean "read it in reverse order" or "process the lines in reverse order"? There's a difference. With the first, potentially the file would not fit in memory all at the same time, so you want to process the lines in reverse order, but you can't read the entire file in and reverse it. With the second, you might just read the entire file in, and reverse the list of lines before processing them. So which is it? – Lasse V. Karlsen Feb 20 at 10:18

add comment

5 Answers

oldest newest **votes**

6

```
for line in reversed(open("filename").readlines()):
    print line.rstrip()
```

link | flag

answered **Feb 20 at 10:10**
Matt Joiner
2,377 ●2 ●14

Thanks a lot... – nimmyliji Feb 20 at 10:54

tagged
python × 19593
file × 1809
reverse × 76

asked
11 days ago

viewed
157 times

latest activity
11 days ago

Wanted: Senior Web Designer/Developer - Director of Technology at Seed Media Group LLC (New York, NY 10010). See this and other great job listings at jobs.stackoverflow.com.

Related

Reverse proxy capable pure python webserver?

Reverse proxy capable pure python webserver?

Python Reverse Generator

Reverse engineering a statistics data file from my insulin pump controller

filter to reverse lines of a text file

Using Cups Reverse Orientation on a

How to manage

BitBucket

Source Code Repository (BitBucket)

- Small projects : 100 lines of code
- Large projects : (Linux kernel 2.6.32 12m LOC) (Windows Server 2003 50m LOC)
- Impossible to manage with one central location for the code, graphics, UI, etc...
- Solution: Code repository, allow pieces to be checked out when needed

How to navigate

Unix

man cat just sounds funny

Learning to navigate across your system is akin to learning to use the mouse. Is it necessary?

Commands to know

- **locate** find files by name
- **ssh** OpenSSH SSH client (remote login program)
- **scp** secure copy (remote file copy program)
- **grep** print lines matching a pattern
- **awk** pattern scanning and text processing language
- **man** an interface to the on-line reference manuals
- **history** GNU History Library
- **cat, head, tail** concatenate files and print on the standard output
- **chmod chown** change file owner and group
- **top** display Linux tasks
- **ps, pkill** look up or signal processes based on name and other attributes

Putting it together

Lit-Py

Lit-Py Input

```
# {\Large Mandbrot Set} \\
# The Mandelbrot set $M$ is defined by a family of complex quadratic polynomials
$P_c:\mathbb C\to\mathbb C$ given by: $P_c: z\to z^2 + c$ where $c$ is a complex parameter.
For each $c$, one considers the behavior of the sequence $(0, P_c(0), P_c(P_c(0)), P_c(P_c(P_c(0))),
\ldots)$ obtained by iterated function $P_c(z)$ starting at critical point $z = 0$, which either escapes
to infinity or stays within a disk of some finite radius. The Mandelbrot set is defined as the set of all
points $c$ such that the above sequence does not escape to infinity.\\
# \textbf{Create a grid}
from pylab import *
X = linspace(-1.5, .8, 200)
Y = linspace(-1,  1, 200)
XG,YG = meshgrid(X,Y)
G = zeros(XG.shape)
#  \textbf{Define the Mandelbrot function}
def MBset(c, z=0):
    for n in xrange(80):
        if abs(z)>2: break
        z = z**2 + c
    return n

# Test the function to see if it is working properly
print MBset(1 + .5J)

#  \textbf{Compute Mandelbrot set}, note that XG and YG are the grid coordinates
for ix in ndindex(G.shape):
    G[ix] = MBset( XG[ix] + YG[ix]*1J )

#  \textbf{Plot the result}
imshow(G, extent=(-1.5, .8,-1,1), interpolation='nearest')
show()
```

Lit-Py Output

Mandbrot Set

The Mandelbrot set M is defined by a family of complex quadratic polynomials $P_c : C \to C$ given by: $P_c : z \to z^2 + c$ where c is a complex parameter. For each c, one considers the behavior of the sequence $(0, P_c(0), P_c(P_c(0)), P_c(P_c(P_c(0))), \ldots)$ obtained by iterated function $P_c(z)$ starting at critical point $z = 0$, which either escapes to infinity or stays within a disk of some finite radius. The Mandelbrot set is defined as the set of all points c such that the above sequence does not escape to infinity.

Create a grid

```
from pylab import *
X = linspace(-1.5, .8, 200)
Y = linspace(-1,  1, 200)
XG,YG = meshgrid(X,Y)
G = zeros(XG.shape)
```

Define the Mandelbrot function

```
def MBset(c, z=0):
   for n in xrange(80):
      if abs(z)>2: break
      z = z**2 + c
   return n
```

Test the function to see if it is working properly

```
print MBset(1 + .5J)
```

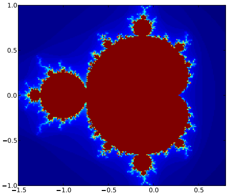
```
≫ 2
```

Compute Mandelbrot set, note that XG and YG are the grid coordinates

```
for ix in ndindex(G.shape):
   G[ix] = MBset( XG[ix] + YG[ix]*1J )
```

Plot the result

```
imshow(G, extent=(-1.5, .8,-1,1), interpolation='nearest')
show()
```


1.0
0.5
0.0
−0.5
−1.0
−1.5
−1.0
−0.5
0.0
0.5