PROCEEDINGS OF SCIENCE

Webnucleo.org

Bradley S. Meyer[*†]
Clemson University
E-mail: mbradle@clemson.edu

The author and his students have written a number of online tools and code modules that are useful for computing various aspects of stellar and explosive nucleosynthesis. This effort is the "Webnucleo.org" project. The code modules are open-source and are freely available to any interested party. This paper briefly describes those modules and *NucNet Tools*, a set of computational tools for nuclear astrophysics written in C++ on top of the webnucleo modules. The paper then illustrates the tools with example calculations that include neutrino-nucleus interactions in nucleosynthesis.

XII International Symposium on Nuclei in the Cosmos,
August 5-12, 2012
Cairns, Australia

[*]Speaker.

[†]This work was supported by NASA Grant NNX10AH78G. The author acknowledges the hospitality of the Division of Theoretical Astronomy at the National Astronomical Observatory of Japan where part of this work was performed.

PoS(NIC XII)096

1. Webnucleo Modules

To study nucleosynthesis and related topics, the author and his students have written a number of code modules that they have released on SourceForge.net and have described on their web site *http://www.webnucleo.org*. The code modules are libraries with well-developed and well-documented Application Programming Interfaces (APIs); thus, the modules are meant to be compiled and linked into codes written by other users. The webnucleo.org modules are open-source and released under the GNU General Public License, which grants the user freedom to modify and/or redistribute them (please see *http://www.gnu.org/licenses/licenses.html* for more information).

The modules themselves depend on two well-established libraries, libxml2 and gsl. libxml2 is the Gnome XML parser and toolkit. It is available for download from *http://www.xmlsoft.org*, but it typically comes with most distributions of linux or unix. Any version later than 2.6.18 works well with the webnucleo.org modules. The other required library is gsl, the GNU Scientific Library. This is available for download from *http://www.gnu.org/software/gsl/*.

The modules particularly relevant for studying nucleosynthesis are:

- **wn_matrix:** This module handles sparse matrices such as those typically encountered in nucleosynthesis calculations. Elements are stored in an efficient hash, and the usual matrix operations are available through the wn_matrix API. See *http://sf.net/projects/wnmatrix/*.

- **libnucnet:** This module stores and manages nuclear reaction networks, that is, nuclear species and reactions among them. The distribution of this module now includes over 50 example codes that demonstrate the use of libnucnet's extensive API. libnucnet requires wn_matrix. See *http://sf.net/projects/libnucnet/*.

- **libstatmech:** This module computes the thermodynamics of fermions and bosons. The default is to use the fully-relativistic, non-interacting formulation, but the user may supply his or her own functions or integrands that include effects such as interactions. See *http://sf.net/projects/libstatmech/*.

- **libnuceq:** This module computes the various equilibria that might occur during nucleosynthesis. The user selects the equilibrium by choosing a particular subset of nuclei using an XPath expression and supplying an abundance constraint on that subset of nuclei. In this way a user may compute full nuclear statistical equilibrium (with or without weak equilibrium), quasi-equilibrium, $(n,\gamma)-(\gamma,n)$ or $(p,\gamma)-(\gamma,p)$ equilibrium, or some more complex choice. libnuceq requires all three of the preceding modules. See *http://sf.net/projects/libnuceq/*.

The webnucleo.org web site has tutorials that demonstrate how to download, install, and compile all code modules. The module distributions include example codes that demonstrate how a user can write his or her own code based on the webnucleo.org modules. Many of the example codes are useful in their own right. For instance, several of the example codes in the libnucnet distribution allow a user to run reaction network calculations using either an exponential expansion of the density and temperature or a table giving the temperature and density as a function of time. Each module also has at least one technical report that describes details behind the codes, their input, or ways to link user-supplied routines to the modules. Interested readers will want to consult those reports.

Nuclide	Atomic Number	Mass Number
o16	8	16
na23	11	23

Table 1: Sample data.

2. Input to libnucnet

The easiest way to get nuclear data and reaction data into libnucnet-based codes is through eXtensible Markup Language (XML) files. Data in XML files are self describing since each datum is contained within user-defined tags: <tag>datum</tag>. The data may be arranged hierarchically; thus, for example, the data for table 1 could be arranged in XML as

```
<table>
  <nuclide>
    <name>o16</name>
    <z>8</z>
    <a>16</a>
  </nuclide>
  <nuclide>
    <name>na23</name>
    <z>11</z>
    <a>23</a>
  </nuclide>
</table>
```

While the XML data format is somewhat verbose, it has a number of advantages. First, the data may be stored hierarchically. Second, it is possible to write XML schemas that define the "grammar" of the file. With a schema, it is then possible to use a schema checker tool such as xmllint to validate the file to ensure that all required data are present and are valid (for example, that the mass number of a nucleus must be a positive, non-zero integer). Third, XML data may be passed across the internet; thus, for example, a user may run his or her code locally but request reaction data in proper XML format from a remote site. libnucnet technical reports describe the appropriate schemas for libnucnet XML input and examples demonstrate how to validate data and read and update them across the web.

3. NucNet Tools

To facilitate study of nucleosynthesis in stars, supernovae, and related environments, the author and his students have constructed a set of open-source, freely available computer codes called *NucNet Tools*, available from SourceForge.net at *http://sourceforge.net/p/nucnet-tools/home/Home*. The codes are built on the webnucleo.org modules described in §1. Posts at the author's blog, at *http://sourceforge.net/u/mbradle/blog* describe how to download, install, and run calculations with

NucNet Tools. The blog describes how to get *NucNet Tools* up and running in a Windows, Mac, or Linux environment.

3.1 The reaction network

This section demonstrates how to run an explosive nucleosynthesis calculation at constant entropy with *NucNet Tools*. It assumes that the user has installed *NucNet Tools* according to the instructions in the blog post at *http://sourceforge.net/u/mbradle/blog/2012/07/test/*. Once the user installs *NucNet Tools* and changes into the *examples/network* directory, he or she changes the line in *Makefile* to read

```
HYDRO_CODE=wind
```

The user then makes the necessary code by typing

```
make run_constant_entropy
```

This code computes nucleosynthesis at the constant entropy input by the user. Most supernova explosive nucleosynthesis environments are fairly well approximated by such constant entropy expansions.

In order to perform the calculations, the user must first retrieve the necessary data. The nuclide and reaction rate data are in a network XML file *my_net.xml*. This file was constructed from the Joint Institute for Nuclear Astrophysics database, as described in the author's blog post *http://sourceforge.net/u/mbradle/blog/2012/10/merging-nuclear-and-reaction-data/*. The user can retrieve such a file by typing

```
make data
```

The user then can retrieve the data for the explosive calculations described in this paper by typing

```
make neutrino_data
```

Since these commands retrieve data across the web, the user's computer must be connected to the internet.

Once the data are available, the user first chooses an initial temperature T_9 ($T_9 = T/10^9$ K) and an initial electron-to-nucleon ratio Y_e (these data are entered through a zone file, called *../../data_pub/zone_nu.xml* in the downloaded data). The code then finds the density ρ_0 such that the matter in nuclear statistical equilibrium at that temperature and Y_e has the input entropy per nucleon. It then allows the material to expand following a density vs. time function $\rho(t)$ and radius vs. time function $r(t)$. For the assumption of mass flow from a spherically symmetric source,

$$\dot{M} = 4\pi r^2 v \rho, \tag{3.1}$$

where $\dot{M}$ is the mass-loss rate and v is the outflow velocity. The default wind routines are based on the assumption that $\dot{M}$ and v are constant in time; thus,

$$r(t) = r_0 + v_0 t, \tag{3.2}$$

where r_0 is the initial radius of the outflowing matter and v_0 is the constant velocity, and

$$\dot{M} = 4\pi r_0^2 (1 + t/\tau)^2 \rho, \tag{3.3}$$

where $\tau = r_0/v_0$. Since we take $\dot{M}$ to be constant, we then find

$$\rho(t) = \frac{\rho_0}{(1 + t/\tau)^2}. \tag{3.4}$$

Should the user choose to use a different function for $\rho(t)$, he or she should modify or add the relevant code in the directory *user/hydro* as desired. Note that quantities relevant for a zone in a *NucNet Tools* calculation can be passed in as optional properties. These can be added to the *zone_nu.xml* file or input directly in the code by the *NucNet Tools* API routine *zone.updateProperty()*. The property is retrieved for a given zone by the routine *zone.getProperty()*. Please note that properties are strings and must be converted to whatever type is required in the program.

The evolution of the network proceeds by maintaining the constant entropy. Consider that the network abundances have been computed for time t. The network then takes a time step Δt. It computes the density $\rho(t + \Delta t)$. It then calls a one-dimensional root finding routine to get the temperature for that density and the constant entropy. For each root finding iteration, the code integrates the relevant thermodynamic quantities and can, if desired, call the reaction network. Once the correct temperature is found, all quantities are updated and the code proceeds to the next time step.

Figures 1 and 2 show a particular calculation at constant entropy per nucleon $50k$, initial $Y_e = 0.5$, and $\tau = 0.05s$. The execution line for this calculation was, in the *examples/network* directory:

```
./run_constant_entropy ../../data_pub/my_net.xml
../../data_pub/zone_nu.xml out.xml "[z <= 50]"
```

The above command is all on one line. The *../../data_pub/my_net.xml* file is the input file for nuclear and reaction data. *zone_nu.xml* contains the initial conditions for the calculation. *out.xml* is the output for the calculation. It is an XML file that can be read with other *NucNet Tools codes* (see *http://sourceforge.net/u/mbradle/blog/2012/07/analyzing-a-first-network-calculation/*). The term "[z <= 50]" is an XPath expression that restricted the reaction network to include only nuclides with charge less than or equal to 50. A user may omit the XPath expression to use the full network, choose a different XPath expression for the nuclei, or add an XPath restriction on the reactions.

Figure 1 shows the temperature vs. time in the calculation. Also shown is a constant ϕ expansion, where ϕ is the photon-to-nucleon ratio such that $\phi \propto T^3/\rho$. The dashed line shows the the temperature that would be present if the calculation maintained a constant ϕ equal to its initial value. Clearly the temperature is higher for the constant entropy expansion than the appropriate constant ϕ expansion for a given ρ because of the reheating of the material by energy released from nuclear reactions and pair annihilations.

Figure 2 shows the entropy in the three components in the system, namely, the baryons (i.e., the nucleons and nuclei), the electrons and positrons, and the photons. Their sum is always $50k$ in the expansion. These entropies were computed with the *NucNet Tools* code *examples/analysis/compute_thermo_quantity.cpp*. As is evident from Figure 2, the entropy starts

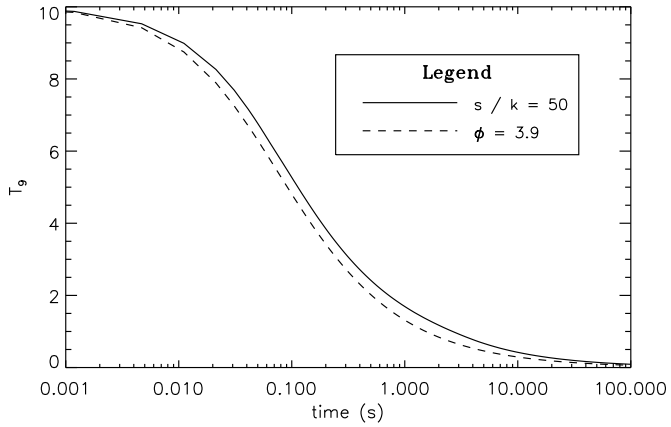

Figure 1: Temperature (in billions of Kelvins vs. time in the constant entropy = 50k and the corresponding constant ϕ expansions.

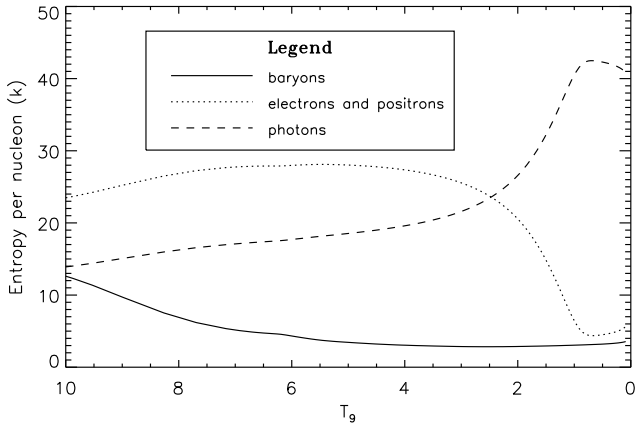

Figure 2: Entropy per nucleon in the various components during the constant $s/k = 50$, $\tau = 0.05$s expansion.

roughly equally shared among the three components. As the expansion proceeds and free nucleons assemble into heavier species, entropy is transfered from the baryons to the leptons and photons. Near $T_9 = T/10^9\mathrm{K} = 2$, the electron-positron pairs annihilate. This dumps their entropy into the photons. Late in the expansion, the remaining electrons become non-relativistic. This causes entropy to transfer back from the photons into the leptons and baryons since now their density $\rho \propto T^{3/2}$ for constant entropy.

3.2 Equilibrium codes

Considerable insight can be gained by comparing the results of reaction network calculations to nuclear equilibrium (see, for example, [1]). For this reason, *NucNet Tools* contains an example

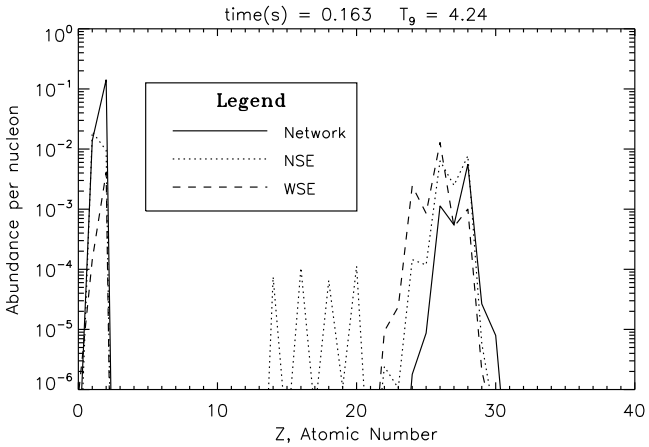

Figure 3: Network elemental abundances vs. the corresponding abundances in NSE and WSE (see text for details.

code *examples/analysis/compare_equil.cpp* that compares the network calculation abundances to various nuclear statistical equilibria.

Figure 3 shows a comparison of the elemental abundances in the calculation described in the previous subsection to those that would be present at the same density and entropy if the matter were in nuclear statistical equilibrium with weak equilibrium (WSE) or simply nuclear statistical equilibrium (NSE) at the same Y_e as the network. It should be noted that the WSE in this case is a "dynamical WSE" since the neutrino chemical potential is $-\infty$ (free-streaming neutrinos) [2]. A dynamical WSE is an NSE for which the time rate of change of Y_e is zero. Had we chosen a finite neutrino chemical potential for the calculation, the WSE calculation would have found the NSE for which the neutrons and protons were in weak equilibrium with each other and the electrons and neutrinos. This is an equilibrium in which the neutrinos are "trapped".

The particular time chosen for Figure 3 was when the network was at a mass density of 4.47×10^5 g/cc (at a time of 0.163 seconds into the expansion). The Y_e of the network at that time was $Y_e = 0.5$. Y_h, the abundance of heavy nuclei (nuclei with atomic number larger than or equal to six) was $Y_h = 0.007308$. From the *examples/analysis* directory, the execution line to compute the equilibrium elemental abundances was

```
./compare_equil ../network/out.xml "[@label1 = '40']"
```

Here *out.xml* was the output XML file from the network calculation discussed in the previous subsection. The XPath expression

```
"[@label1 = '40']"
```

selects out time step number 40.

As seen in Figure 3, the network abundances are not in WSE or NSE at this time. The QSE abundances, on the other hand, match the network abundances extremely well (and are therefore not shown in the figure); thus, the heavy nuclei are in equilibrium under exchange of free neutrons

and protons, but the number of heavy nuclei is not that required for NSE (or WSE). The QSE abundances are higher charge than those in NSE (or WSE), which indicates the QSE has too few heavy nuclei compared to NSE, a characteristic of high-entropy expansions. It is also interesting that the Y_e for the WSE, that is, the value of Y_e the network would evolve to if the system were held at fixed temperature and density at that point in the expansion is 0.466. This is close to the Y_e of ^{56}Fe (0.464) and explains why the WSE distribution peaks at iron while the network peaks at nickel (^{56}Ni is the most bound nuclide with $Y_e = 0.5$). The interested user is encouraged to try similar calculations and comparisons at other time steps in the expansion.

4. Including neutrinos

It is straightforward to include neutrino interactions in a network calculation with the downloadable code. Standard rate parameterizations for libnucnet are input as tables, single values, or Non-Smoker fits [3]. Other parameterizations can be defined by the user, however, and this is the technique employed here for the neutrino interactions. The default routines in *NucNet Tools* uses neutrino interaction rates on free nucleons from [4], inelastic neutrino scattering rates on ^{4}He from [5], and rates on nuclei from [6]. The data are included in the downloaded XML file *nu.xml*. A user who wishes to create a similar file from his or her own data should refer to the example codes in the latest libnucnet distribution.

To include neutrino interactions in the network calculation, a user defines the relevant routines to compute the rates from input data. In *NucNet Tools*, these rates are defined in the file *my_neutrino_rate_functions.cpp* and the accompanying header *my_neutrino_rate_functions.h* in the *user* directory. If the user defines different neutrino interaction rates, it will be easiest to do that in these files.

Once the rate routines are defined, the user then registers them in the network with the appropriate key found in the input XML file. The rates also require data about the neutrino fluxes and energies, and such data are updated for each rate function at each time step. The codes *run_constant_entropy.cpp*, *user/evolve.cpp*, and *user/my_neutrino_rate_functions.cpp* demonstrate how to do this. For *NucNet Tools*, the rate per nucleus is derived from the the expression

$$\lambda = F_\nu \sigma_\nu, \tag{4.1}$$

where the neutrino flux F_ν is computed from the neutrino luminosity L_ν by

$$F_\nu = \frac{L_\nu}{4\pi r^2 \langle E_\nu \rangle}, \tag{4.2}$$

and the average neutrino energy is given by $\langle E_\nu \rangle = 3.15 T_\nu$, with T_ν the blackbody temperature of the neutrinos. The cross section σ_ν is computed from the rate expressions. The cross sections for neutrino-nucleus interactions are computed in a routine that interpolates the relevant value from the logarithm of the cross section per nucleus, which are data included from the *nu.xml* file.

Figure 4 shows the final abundances vs. mass number for three calculations. The first was that described in §3.1. It did not include neutrinos. The second included neutrinos. The neutrino temperatures, set as optional properties in *zone_nu.xml*, were $T_{\nu_e} = 4$ MeV, $T_{\bar{\nu}_e} = 5$ MeV, and $T_{\nu_x} = 7$ MeV, where ν_x is any of the other neutrino flavors (ν_μ, $\bar{\nu}_\mu$, ν_τ or $\bar{\nu}_\tau$). The neutrino luminosities

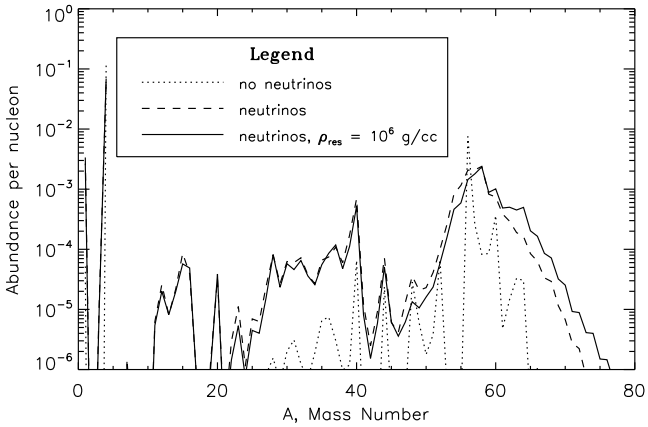

Figure 4: Final abundances vs. mass number in the network calculation without neutrinos, with neutrinos, and with neutrinos and a swapping of $\bar{\nu}_e$ and $\bar{\nu}_\tau$ below a mass density of 10^6 g/cc.

were all set to 3×10^{51} ergs/s and were taken to be constant in time. The execution line (a single line) for this calculation was

```
./run_constant_entropy ../data_pub/my_net.xml
../data_pub/zone_nu.xml out.xml "[z <= 50]" "" ../data_pub/nu.xml
```

where out.xml is the output in XML format. The third calculation was the same as the second except that the $\bar{\nu}_e$ and $\bar{\nu}_\tau$ were taken to swap completely at a density of 10^6 g/cc. This was done in the routine *swap_neutrinos()* in *user/my_neutrino_rate_functions.cpp*. In particular, in these runs, an optional property *resonant density* was defined and used as the density at which complete neutrino interconversion occurred. (In fact, the second calculation was performed with *resonant density* simply set equal to zero.) This choice of conversion is simply for illustration; the user should modify the relevant routines for his or her own purposes.

From Figure 4, it is apparent that neutrinos can have a significant effect on the resulting nucleosynthesis. In particular, the neutrinos allowed heavier nuclei on average to form during the expansion. Partly this is due to the fact that the neutrinos change the electron-to-nucleon ratio Y_e during the expansion (see Figure 5) and partly due to the fact that they keep the abundance of neutrons and protons higher than they would be in the absence of the neutrinos. The remaining nuclei can then capture these nucleons and thereby increase their mass (cf. [7]). These effects are both moderately enhanced by the $\bar{\nu}_e \leftrightarrow \bar{\nu}_\tau$ conversion.

These calculations are certainly not meant to be definitive but rather illustrative. The interested user is encouraged to run his or her own calculations and modify the code as needed for the particular problem of interest.

5. Conclusion

The author and his students have developed codes and code modules that should be helpful for

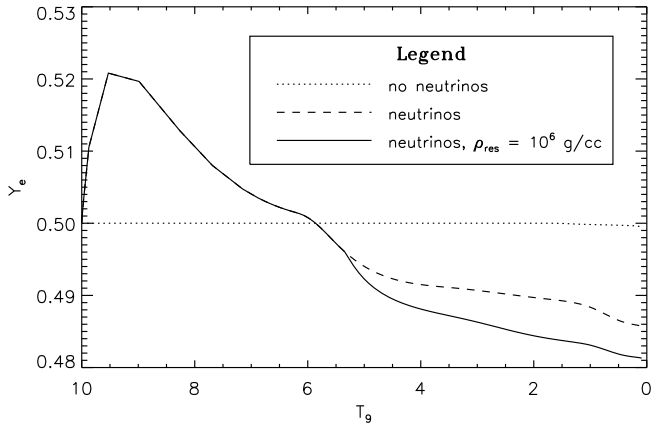

Figure 5: Y_e in the network calculation without neutrinos, with neutrinos, and with neutrinos and a swapping of $\bar{\nu}_e$ and $\bar{\nu}_\tau$ below a mass density of 10^6 g/cc.

studying nucleosynthesis. *NucNet Tools* and the associated modules are under continual development, and interested users should check the appropriate web sites and the author's blog for updates. Users are also encouraged to report bugs and request features as Tickets for the appropriate module.

Our codes, while useful in their present form, should not be viewed as final, immutable products. Rather they are intended to be building blocks for other tools. Users who develop their own resources based on webnucleo.org modules are welcome to announce them on the webnucleo.org public mail list.

References

[1] B. S. Meyer, Krishnan, T. D., and Clayton, D. D., ^{48}*Ca Production in Matter Expanding from High Temperature and Density*, *Ap. J.* **462** (1996) 825.

[2] A. Arcones, G. Martínez-Pinedo, L. F. Roberts, and S. E. Woosley, *Electron fraction constraints based on nuclear statistical equilibrium with beta equilibrium*, *Astron. Ap.* **522** (2010) A25.

[3] T. Rauscher and F.-K. Thielemann, *Astrophysical Reaction Rates From Statistical Model Calculations*, *At. Data Nucl. Data Tables* **75** (2000) 1.

[4] Y.-Z. Qian and S. E. Woosley, *Nucleosynthesis in Neutrino-driven Winds. I. The Physical Conditions*, *Ap. J.* **471** (1996) 331.

[5] S. E. Woosley, D. H. Hartmann, R. D. Hoffman, and W. C. Haxton, *The nu-process*, *Ap. J.* **356** (1990) 272.

[6] B. S. Meyer, G. C. McLaughlin, and G. M. Fuller, *Neutrino capture and r-process nucleosynthesis*, *Phys. Rev. C* **58** (1998) 3696.

[7] C. Fröhlich, G. Martínez-Pinedo, M. Liebendörfer, F.-K. Thielemann, E. Bravo, W. R. Hix, K. Langanke, and N. T. Zinner, *Neutrino-Induced Nucleosynthesis of A>64 Nuclei: The νp Process*, *Physical Review Letters* **96** (2006) 142502.