Python program for perturbative renormalisation group

Roland Bauerschmidt*, David Brydges† and Gordon Slade†

March 21, 2014

Abstract

We describe the included computer program for perturbative renormalisation group calculations.

Contents

The package consists of three files:

- `wick.py` is a collection of Python classes to handle the combinatorics involved in calculating Gaussian integrals and their fermionic analogs. It cannot be executed by itself.
- `sawpt.py` utilizes `wick.py` to compute the specific expressions for the supersymmetric self-avoiding walk field theory as described in [2].
- `phi4npt.py` similarly computes the specific expressions for the n-component $|\varphi|^4$ model as described in [1]. It has an adjustable parameter `N` (in the source file) which sets the number of components.

The output of `sawpt.py` respectively `phi4npt.py` is the field polynomial P_x defined in [2, (3.22)] and computed using the equivalent expression [2, (5.16)], without observables.

Example

The last seven lines of the output of `sawpt.py` is:

```
[...]
 + [ - 4 g \nu W(x-y)
 + 4 g \nu_+ W_+(x-y)
 - 8 g^2 W(x-y)^2
```

*Department of Mathematics, University of British Columbia, Vancouver, BC, Canada V6T 1Z2. Present address: School of Mathematics, Institute for Advanced Study, Princeton, NJ 08540 USA. E-mail: brt@math.ias.edu.

†Department of Mathematics, University of British Columbia, Vancouver, BC, Canada V6T 1Z2. E-mail: db5d@math.ubc.ca, slade@math.ubc.ca.

```
 + 8 g^2 W_+(x-y)^2
 + 2 z g \Delta W(x-y)
 - 2 z g \Delta W_+(x-y)
] \tau_x^2
```

which is to be interpreted as

$$4g\sum_{y}[\nu_+ w_+(x-y) - \nu w(x-y)] + 8g^2\sum_{y}[w_+(x-y)^2 - w(x-y)^2] \\ - 2zg\sum_{y}[\Delta w_+(x-y) - \Delta w(x-y)]. \quad (1)$$

The sum over y is implicit in the output of `sawpt.py` and `phi4npt.py`. Also, the output contains \ `nabla`$= \nabla$ and \ `nabla`$\wedge$`*`$= \nabla^*$ which have an implicit index e which is to be summed over the d unit vectors in $\mathbb{Z}^d$ in positive coordinate direction. Here ∇_e and $\nabla_e^* = \nabla_{-e}$ are discrete gradients:

$$\nabla_e f(x) = f(x+e) - f(x), \quad \nabla_e^* f(x) = f(x-e) - f(x). \quad (2)$$

Simplifications to computer output

To arrive at the expressions stated in [1, 2], we use the following simplifications to the output of `sawpt.py` and `phi4npt.py`:

- $\sum_y \nabla w(x-y) = \sum_y \nabla^* w(x-y) = \sum_y \Delta w(x-y) = \sum_y \nabla^*\nabla w(x-y) = 0$
- $\Delta x_1^2 = 2$ and thus $\sum_y \Delta w(x-y)(x_1-y_1)^2 = \sum_y w(x-y)\Delta(x_1-y_1)^2 = 2\sum_y w(x-y)$.
- $\nabla\nabla^* = -\Delta$

as well as the notation for sums introduced in [2].

References

[1] R. Bauerschmidt, D.C. Brydges, and G. Slade. Scaling limits and critical behaviour of the 4-dimensional n-component ϕ^4 model. Preprint, (2014).

[2] R. Bauerschmidt, D.C. Brydges, and G. Slade. A renormalisation group method. III. Perturbative analysis of weakly self-avoiding walk. Preprint, (2014).