

Coulomb Gas Library Instructions

Adam Mielke

March 26, 2020

1 Introduction

This Matlab library is a condensed version of the one used in my PhD-thesis [1] and applied in the related articles [2, 3]. Parts of the descriptions are taken from there. This document contains a description of the files used. Ideally, the user should never have to access the folder “data.” The parameters are instead chosen in the files, and the naming convention ensures that the right data is chosen.

The fitting parameter is the β of the Coulomb gas, and the comparison of two distributions $f(x)$ and $g(x)$ is done with the Kolmogorov distance

$$D_{\text{KS}} = \max_x |F(x) - G(x)| \leq 1, \quad (1.1)$$

where $F(x)$ and $G(x)$ are the cumulative distributions of $f(x)$ and $g(x)$, respectively. This has the advantage of being unbinned and takes into account that the distributions are normalised.

Apart from the scripts described below, the library also includes GinibreNNN.m, which calculates the NNN-spacing of Ginibre.

2 Coulomb Gas Simulation

The generation of Coulomb gas realisations is done with the file CoulombGasSim.m. The Metropolis-Hastings algorithm is used [4, 5], description taken from [1]. Let $H(X)$ be the energy associated with the configuration X .

- Generate a new configuration X' . (In this case as a small Gaussian perturbation of X . The width of the Gaussian is chosen as $N^{-1/3}$ in accordance with [5].)
- Accept the new configuration with probability

$$p = \begin{cases} e^{-(H(X')-H(X))} & \text{if } H(X') > H(X) \\ 1 & \text{otherwise} \end{cases} \quad (2.1)$$

- Repeat this process until convergence. (In this case I simply make $100N$ perturbations of individual points, corresponding to roughly 100 iterations of this process. Convergence can be verified by comparing $\beta = 2$ to Ginibre.)

As it takes some time for each run, the results of CoulombGasSim.m for 200 particles, 10^4 realisations, and $\beta = 0, 0.1, \dots, 2$ are included as part of the library, and so are results for 300 particles, 10^4 realisations, and $\beta = 0.5, 1, 0.5, \dots, 4$ that was used for Figure 2 in the supplementary material of [2].

To save time in the individual calculations, the NN and NNN distances are calculated and stored with the file CoulombGasLoad.m. This has also been done for the library data.

They are then fit with the functions CoulombGasFit.m, CoulombGasFit2.m, CoulombGasFitTargeted.m, and CoulombGasFit2Targeted.m. They are implemented in Buzzard_Read.m. (This is explained in more detail in Section 4.) The first two go through all the given β -values and finds the smallest Kolmogorov distance. The 2 denotes that the function looks at the NNN-spacing rather than the NN-spacing. The second two (“Targeted”) takes a starting point and explores the β around it. This makes the fit faster, but may miss the global minimum if several local ones exist. The 2 again denotes the NNN-spacing.

2.1 Uncertainty

To estimate the uncertainty of the fit, the bootstrapping method described in [3] is used. The Kolmogorov fit does not give a clear connection to the uncertainty the same way a least-squares fit does. I generate a number of Coulomb gas realisations with a true $\beta = \beta_0$ and make a fit with the full distribution to find an estimate of the uncertainty on β . This is done in betaUncertainty.m. This file is designed for [3] and therefore looks at moving averages of a given size. It may be adapted to look at independent parts if needed.

This calculation takes a long time. It may be improved with a faster minimum distance calculation. The library data also contains the output for a moving average of 1, 5, and 10 using $\beta = 0-1.5$, $\beta = 0-1.2$, and $\beta = 0-1$, respectively. Note that the highest values of β used in the simulation will not have an accurate error. (Here only $\beta \leq \beta_0$ fits are possible.)

The actual calculation of the error is done in Buzzard_Read.m, see Section 4 and [3] (from which the following description is taken). Because the realisations are grouped, there is also some information about the cross-correlation between the years. I extract this information by an average over groups of distance k between the midpoint for a given β_0 and call the correlation found here $V_k^{\beta_0}$. This method does not represent the nests completely, because only realisations of the same β_0 are grouped, but going in to the individual β_0 of each year would defeat the purpose of grouping. Instead, to compare two groups with different β_0 and β_1 , I use the heuristic combination

$$V_k^{\beta_0, \beta_1} = \sqrt{\frac{V_k^{\beta_0^2} + V_k^{\beta_1^2}}{2}}. \quad (2.2)$$

I add them in quadrature to reflect the structure of the least-squares error, and the factor of 2 in the denominator ensures that $V_k^{\beta_0, \beta_0} = V_k^{\beta_0}$. From here the full covariance matrix of the grouped years may be constructed. This is done partially in Buzzard_Read.m and partially in CovCon.m.

3 Unfolding

To compare the local spacing distributions of two sets of data, the macroscopic density of both datasets must be made uniform. This is called unfolding and means that the fluctuations (fl),

that are supposedly universal, have to be separated from the global, averaged (av) spectral density which is system specific:

$$\rho(x, y) = \sum_{i=1}^N \delta^{(2)}(z - z_i) = \rho_{\text{av}}(x, y) + \rho_{\text{fl}}(x, y) , \quad (3.1)$$

where $z = x + iy$. For more details, see [1, 2]. The unfolding method used in [2, 3] is a Gaussian broadening. I approximate $\rho_{\text{av}}(x, y)$ by a sum of Gaussian distributions around each eigenvalue $z_j = x_j + iy_j$,

$$\rho_{\text{av}}(x, y) \approx \frac{1}{2\pi\sigma^2 N} \sum_{j=1}^N \exp\left[-\frac{1}{2\sigma^2}|z - z_j|^2\right] \quad (3.2)$$

and find the weighted distances

$$ds^2 = \rho_{\text{av}}(x, y)(dx^2 + dy^2) . \quad (3.3)$$

The unfolding process for 2D is done in the function `UnfoldingGaussian.m`, where the weighted distances. The tricky part is to determine the width σ of the Gaussian. The value $\sigma = 4.5\bar{s}$ in terms of the mean level spacing \bar{s} has been found by comparing to products of Ginibre matrices.

For 1D, this was first treated with a Gaussian broadening in [6]. There the density of nucleon levels (not the spectral density) is used to determine the width. Where this reaches a constant is considered the correct width. I instead use the Kolmogorov distance as a measure of when the spectral density becomes uniform.

It may be shown that the unfolded density in 1D is the cumulative density

$$\rho_{\text{Uniform}}(x) = \int_0^x \rho(x') dx' , \quad (3.4)$$

and for a given dataset, the optimal Gaussian width may be calculated. The unfolded points are

$$X_j = \frac{1}{N} \sum_{k=1}^N \frac{1 + \operatorname{erf}\left(\frac{(x_j - x_k)}{\sqrt{2}\sigma}\right)}{2} , \quad (3.5)$$

where the Gaussian unfolding has been used. The Kolmogorov distance is evaluated at the known points X_j . The uniform density has a linear function as cumulative function, so the coordinate set $\{X_j, X_j\}$. The cumulative function of the unfolded spectrum is the counting function, so the set $\{X_j, \frac{j}{N}\}$. The Kolmogorov distance is therefore

$$D_{\text{KS}} = \max_j \left| \frac{1}{N} \sum_{k=1}^N \frac{1 + \operatorname{erf}\left(\frac{(x_j - x_k)}{\sqrt{2}\sigma}\right)}{2} - \frac{j}{N} \right| , \quad (3.6)$$

which has the minimum $\sigma = 0$. This also makes sense intuitively, as the most uniform distribution must be the one where the points are spread out equidistantly.

4 Buzzard Nests

This concerns the file `Buzzard_Read.m`, the results of which are published in [3]. It analyses the spacing of buzzard nests in the Teutoburger forest north-west of Bielefeld. For conservation

reasons, the real data has not been included, but a generated dataset of the same format has been included for illustration.

The analysis requires the appropriate Coulomb gasses to have been generated and their distances calculated with `CoulombGasSim.m` and `CoulombGasLoad.m`, respectively. The unfolded distances of the data points are found, grouped, and compared to the Coulomb gasses. The uncertainties of the fit found with `betaUncertainty.m` are also implemented for a fit of the groups. The correlation between different groups is found by averaging as described in Section 2.1 with `CovCon.m`.

The analysis in `BuzzardRead.m` also includes the option of cutting away the edge of the data points with a mask and look only at the bulk. This is done with `CutOutBuzz.m`, but seems to introduce a bias for the buzzard data. It may be useful for other dataset and is therefore included.

5 References

- [1] A. Mielke, *On the Role of Zero Modes and Spacing Distributions in Random Matrix Theory and its Applications*, (2020), PhD thesis. Available at <https://pub.uni-bielefeld.de/record/2940010>.
- [2] G. Akemann, M. Kieburg, A. Mielke, T. Prosen, *Universal Signature from Integrability to Chaos in Dissipative Open Quantum Systems*, Phys. Rev. Lett. 123, 254101, 1–6 (2019) [arXiv:1910.03520].
- [3] G. Akemann, M. Baake, N. Chakarov, O. Krüger, A. Mielke, M. Ottensmann, and R. Werdehausen, *Territorial Behaviour of Buzzards Versus Random Matrix Spacing Distributions*, In preparation.
- [4] W. K. Hastings, *Monte Carlo sampling methods using Markov chains and their applications*, Biometrika 57, 97–109 (1970).
- [5] D. Chafaï and G. Ferré, *Simulating Coulomb gases and log-gases with hybrid Monte Carlo algorithms*, J. Stat. Phys. 174, 692–714 (2019) [arXiv:1806.05985].
- [6] V. M. Strutinsky, *Shell Effects in Nuclear Masses and Deformation Energies*, Nuclear Physics **A95**, 420–442 (1967).