Space Weather of the Heliosphere: Processes and Forecasts
Proceedings IAU Symposium No. 335, 2017
C. Foullon & O.E. Malandraki, eds.

doi:10.1017/S1743921317007244

Sandpile Models and Solar Flares: Eigenfunction Decomposition for Data Assimilation

Antoine Strugarek[1,2], Allan S. Brun[1], Paul Charbonneau[2] and Nicole Vilmer[3]

[1]Laboratoire AIM Paris-Saclay, CEA/Irfu Université Paris-Diderot CNRS/INSU, F- 91191 Gif-sur-Yvette
email: antoine.strugarek@cea.fr

[2]Département de physique, Université de Montréal, C.P. 6128 Succ. Centre-Ville, Montréal, QC H3C-3J7, Canada

[3]LESIA, Observatoire Paris, CNRS, UPMC, Universite Paris-Diderot, 5 place Jules Janssen, 92195 Meudon, France

Abstract. The largest solar flares, of class X and above, are often associated with strong energetic particle acceleration. Based on the self-similar distribution of solar flares, self-organized criticality models such as sandpiles can be used to successfully reproduce their statistics. However, predicting strong (and rare) solar flares turns out to be a significant challenge. We build here on an original idea based on the combination of minimalistic flare models (sandpiles) and modern data assimilation techniques (4DVar) to predict large solar flares. We discuss how to represent a sandpile model over a reduced set of eigenfunctions to improve the efficiency of the data assimilation technique. This improvement is model-independent and continues to pave the way towards efficient near real-time solutions for predicting solar flares.

Keywords. Sun: flares

1. Introduction

Solar eruptive phenomena have multiple impacts on the heliosphere and on the environment around our Earth (e.g. Schwenn 2006; Pulkkinen 2007). Along with Coronal Mass Ejections (CMEs), the solar flares are one of the most studied space weather phenomenon and can be accompanied by the acceleration of highly energetic particles. The energy and duration of solar flares have been shown to be distributed as power-laws (for a review, see Aschwanden *et al.* 2016), spanning more than eight orders of magnitude in energy. This striking self-organization (Charbonneau 2013) over the whole corona have led multiple authors to use basic, toy models such as sandpiles (e.g. Lu & Hamilton 1991) to model the statistical distribution of flares.

In the context of space weather, and also to better constrain the physical mechanisms behind flares, it is thus interesting to try to predict their occurence. Such a task was recently shown to be very difficult (Barnes *et al.* 2016), as all of the modern techniques developed to predict flares most often fail. Alternative methods and ideas are thus needed today to explore other paths to provide robust predictions of solar flares. We build here on an idea first developed by Bélanger *et al.* (2007); Strugarek & Charbonneau (2014) based on the combination of simple sandpile models reproducing the solar statistical distribution (Strugarek *et al.* 2014), and modern data assimilation techniques using the solar X-ray flux observed with the GOES satellites.

The aim of this proceeding is to report on an attempt to facilitate the data assimilation algorithm by reducing size of the state vector of the model. This is achieved by projecting

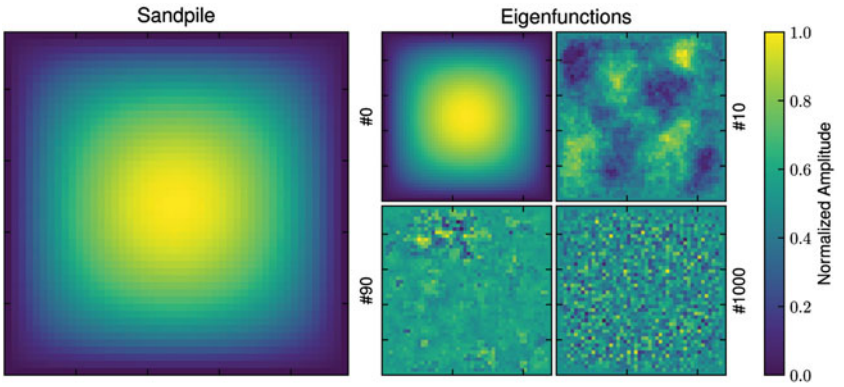

Figure 1. *Left.* 2D Sandpile model. *Right.* Four representative eigenfunctions of the sandpile.

the model itself on a basis of eigenfunctions (§ 2), which we couple to the data assimilation package (§ 3) that was originally developed in Strugarek & Charbonneau (2014). We conclude and lay out the perspective of this preliminary work in § 4.

2. Deterministically-driven sandpile and eigenvalue decomposition

We base our analysis on the deterministically-driven sandpile model developed by Strugarek *et al.* (2014). However, the procedure described here is independent of the model and could be applied to other types of modelling (e.g. Hung *et al.* 2017).

The aim of data assimilation is to adjust a set of control parameters (a 'state vector') of the model to get an optimal fit to observed (or synthetic) data. One straightforward strategy is to consider the initial condition of the model as this state vector. The advantage of this approach is that we ensure that we do not tune the model itself (*i.e.* the model retains its statistical properties no matter what we do to the set of control parameters), but we rather try to find a moment in the evolution of the model that fits the observational data.

One big disadvantage of this approach lies in the size of the set of control parameters. Indeed, the size of our set will be N^2 (N being the number of points in our 2D sandpile model), which will become quickly numerically prohibitive for most minimization techniques as N gets large. In order to mitigate this limitation, we explore here the possibility to project the model on a series of eigenfunctions to reduce the size of our state vector, e.g. by adjusting only the most relevant ones. We now describe in details our procedure.

We construct the eigenfunction basis of our model using its covariance matrix. Let us define the state of our sandpile model at time t by $\mathcal{S}(t)$, taken as a 1D vector of size N^2. We define $\mathcal{S}_0 = \mathcal{S}(t_0)$. A covariance matrix $\mathcal{C}$ of our model is a $N^2 \times N^2$ matrix that is defined as

$$\mathcal{C} = \overline{(\mathcal{S}(t) - \mathcal{S}_0)^\intercal \cdot (\mathcal{S}(t) - \mathcal{S}_0)}\,, \tag{2.1}$$

where the overline denotes a time average. We diagonalize the covariance matrix to obtain a set of N^2 eigenvectors of our model. This decomposition is illustrated in Figure 1 where we show the sandpile model itself along four representative eigenfunctions (out of 2304 in this case).

Now that we have a systematic method to derive a set of eigenfunctions of our model, we want to assess whether a subset of the eigenfunctions is enough to robustly control the large events triggered by the model in a given time window. We first choose one random state of our model $\mathcal{S}$, which we project on the eigenfunctions basis to obtain a

https://doi.org/10.1017/S1743921317007244 Published online by Cambridge University Press

set of eigenvalues. We run the sandpile model for 10^4 iterations to define our reference case. Then, we iteratively perturb each eigenvalue and compare the result of the sandpile model when using the perturbed initial state. We use a cost function $\mathcal{J}$ to estimate how much the perturbation affects the evolution of the sandpile, which we define as

$$\mathcal{J} = 1 - \left(\alpha \sum_{\text{match}} \frac{E_{\text{obs}}}{E_{\text{tot}}} - \beta \sum_{\text{miss}} \frac{E_{\text{obs}}}{E_{\text{tot}}} - \gamma \sum_{\text{FalseAlarm}} \frac{E_i}{E_{\text{tot}}} \right), \tag{2.2}$$

where E_{obs} is the energy of the events in the reference model, E_i the energy of events in the model being compared, $E_{\text{tot}} = \sum E_{\text{obs}}$ is the total energy of all the events in the reference model, and α, β and γ are adjustable coefficient that we set here to 1, 0.5 and 0.25. This cost function gives more weight to the large events that we most want to predict reliably. It also gives more weight to matching events, then missed events, and finally less weight to false alarms. If the model fits perfectly the reference model, $\mathcal{J} = 0$. Conversely, if none of the events fit the reference model, $\mathcal{J} \sim 1.75$ (no match, all events are considered as false alarms).

We repeat this procedure for 10 different random number sequences used in the sandpile model to ensure that the subset of eigenfunctions we are trying to isolate is robust and order the eigenfunctions in ascending order with their score $\mathcal{J}$. We now use this ordered basis of our model to perform data assimilation over synthetic data.

3. Data assimilation using the eigenvalue decomposition

The data assimilation procedure was briefly described in Strugarek & Charbonneau (2014) and can be summarized as follows.

We first construct synthetic data (upper right panel in Figure 2) by running our model over 4000 iterations and filtering events higher than 10^4 E_0 (E_0 being the energy normalization of our adimensional model, see Strugarek & Charbonneau 2014). This will be our target observation that we will try to match by finding a new initial state for our system. Then, we generate random states of our model that will be used as initial states for our data assimilation algorithm. We project these initial states on our ordered eigenfunction basis. After these initialization steps, we minimize the cost function $\mathcal{J}$ (Eq. 2.2) with a simulated annealing algorithm. The set of control parameters for this algorithm is defined by the amplitude of the first N_e eigenvalues, which we allow to evolve. The remaining $N - N_e$ eigenvalues are thus kept fixed during the assimilation procedure. We tested our algorithm for values of N_e ranging from 10 to $N_{\max}$ ($N_{\max} = 2206 < N_e$ due to the boundary conditions of the sandpile forcing it to zero on its edges). The resulting final state and execution time of the data assimilation procedure are shown in Figure 2.

On the top left panel we display the final cost function after the data assimilation procedure. The blue bars denote assimilation with varying N_e, while the orange bar corresponds to a standard assimilation run for which no eigenfunction decomposition was used (thus the set of control parameters corresponds to the N^2 points in the initial sandpile, see Strugarek & Charbonneau 2014). We immediately remark that if we use too few eigenfunctions in the data assimilation procedure, the system is too constrained and we are not able to reproduce satisfactorily the reference model. When $N_e \geqslant 50$, we empirically observe that we are able to match equally or better the reference model ($\mathcal{J}$ is smaller) than the standard assimilation case, and with a much faster execution time (lower left panel). The reference model used in the data assimilation procedure is shown in the right panels in black and grey. Three realisations of the model resulting from the assimilated initial states are shown in the three other panels.

https://doi.org/10.1017/S1743921317007244 Published online by Cambridge University Press

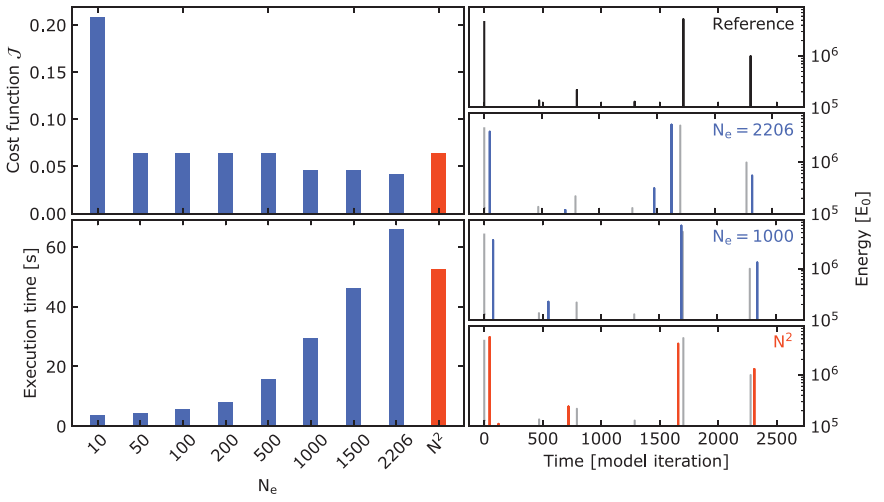

Figure 2. *Left.* Final cost function (top) and execution time (bottom) of the assimilation procedure, as a function of the number of eigenfunctions N_e involved in the assimilation. The orange bar corresponds to an asssimilation with no eigenvector projection, using all the $N^2 = 2304$ points in the sandpile. *Right.* Results from the sandpile models. The reference case being assimilated is at the top and reported in gray in all panels. The three lower panels are representative cases.

4. Conclusions

We have shown in this proceeding how to automatically decompose a sandpile model on a set of eigenfunctions. We found that the shape of the eigenfunctions described both the large-scale structure of the sandpile, as well as nested structures associated with locations in the sandpile likely to give rise to avalanches (see Figure 1).

In the context of data assimilation for predicting solar flares (Strugarek & Charbonneau 2014), this decomposition allows to reduce significantly the number of control parameters in the data assimilation procedure. Only a subset of eigenvalues can be used to control the subsequent evolution of a sandpile, keeping the other eigenvalues constant. This opens promising perspectives in designing a predictive tool based on sandpile models, even using expensive numerical minimization techniques such as simulated annealing.

We plan to make use of the eigenfunction decomposition to test our data assimilation procedure on real data in a future work. The speed up allowed by this decomposition will allow us to explore the parameter space of the minimization procedure on a large sample of flaring epochs for the Sun, and ultimately characterize the predictive capabilities of sandpiles when applied to a real system such as the flaring Sun.

References

Aschwanden, M. J., Crosby, N. B., Dimitropoulou, M., *et al.* 2016, *Space Sci Rev*, 198, 47
Barnes, G., Leka, K. D., Schrijver, C. J., *et al.* 2016, *ApJ*, 829, 89
Bélanger, E., Vincent, A. & Charbonneau, P. 2007, *Sol. Phys.*, 245, 141
Charbonneau, P. 2013, in Self-Organized Criticality Systems, ed. M. J. Aschwanden (Open Academic Press), 404
Hung, C. P., Brun, A. S., Fournier, A., *et al.* 2017, *ApJ* submitted
Lu, E. T. & Hamilton, R. J. 1991, *ApJL*, 380, L89
Pulkkinen, T. 2007, *LRSP*, 4, 1
Schwenn, R. 2006, *LRSP*, 3, 2
Strugarek, A. & Charbonneau, P. 2014, *Sol. Phys.*, 289, 4137
Strugarek, A., Charbonneau, P., Joseph, R. & Pirot, D. 2014, *Sol. Phys.*, 289, 2993

https://doi.org/10.1017/S1743921317007244 Published online by Cambridge University Press