POLITECNICO DI TORINO

A Master thesis presented for the
Master in Physics of Complex Systems

Optimization of the gradient descent dynamics in simple mean field spin glasses.

Politecnico
di Torino

Candidate: Lombardo Yuri
Relator: Pagnani Andrea
Relator: Urbani Pierfrancesco

Academic Year 2020/2021

Contents

1 Introduction to Optimization Problems

In recent years, analyzing big data sets and systems with a high number of particles has become more and more important in several fields. These systems, presenting a high dimensionality, are difficult to tackle due to the large number of degrees of freedom we have to handle. For instance, a common operation we have to perform is the minimization of a function, like the energy, and in such situations, the landscape is usually rugged and reaching the global minimum may become a non-trivial procedure; there could be several local minima that prevent us from reaching the global one. Nevertheless, the simplest and most used algorithms are based on the gradient descent (GD), which was developed two centuries ago [1], and these have shown to be to be straightforward and successful in a various array of applications.

Some examples of fields where such algorithms are used are:

- Machine Learning (supervised learning): where one has to minimize the cost function of the model to estimate which are the best parameters that we will use to tune the model itself. For instance, in the case of neural networks, the degrees of freedom can rapidly reach values of order $10^6 - 10^7$ and therefore the landscape becomes difficult to explore [2, 3].

- Inference: where we have to deduce an underlying structure from a set of signals (possibly noisy) and depending on the particular characteristic of the signal vector we can end up in rugged landscapes where GD is showing promising results [4, 5, 6].

- Evolutionary Biology: here the fitness (the ability of passing on their genes to subsequent generations) of a species or individual is fundamental in understanding who has a higher likelihood to survive; the ones that show the maximum fitness will thrive in the environment, but this function is influenced by several variables leading to a complex landscape [7].

Due to their simplicity, vanilla algorithm are still the most used but, given that the problems are getting tougher to solve and that the computational power is expected to increase at a slower pace in the near future [8], interest is shifting more toward optimization and efficiency. For instance, when we have to deal with non-convex optimization problems, newer and updated versions of old algorithm showed promising results [9, 10].

2 Mean field spin glasses as Hard Optimization Problems

2.1 Disordered systems

A computer scientist may initially think that a theoretical physical system has no meaning for his studies; but one can soon realize that some of these are a fertile ground to develop and test new algorithms given the complex settings. In this framework, there is a class of systems that fits perfectly our interests: the disordered systems. We can be distinguish two two major sub-classes based on two different kind of disorder. The first one is the annealed disorder and it refers to systems where the random variables, defining the randomness, change over time. For example if we have a hot material composed of different elements that is slowly cooling down, the different particles will move to reach the equilibrium, thus changing their position in space and their interaction with the other particles. The second case is the *quenched disorder*, completely opposite to the first, the impurities do not equilibrate with the environment but instead are stuck in random fixed positions. One of the most studied model of that kind is the Edwards–Anderson model [11]:

$$\mathcal{H} = \sum_{<ij>} J_{ij}\sigma_i\sigma_j$$

where the variables σ_i are spins, and the couplings J_{ij} are Gaussian random variables and the summation is performed over nearest-neighbor sites. The Hamiltonian is clearly quenched because the Js are randomly chosen but have no time dependence. This randomness of the couplings leads to the so-called "frustration": a situation in which it's impossible to satisfy all couplings at the same time, as it would be in a ferromagnetic system. Formally, a system is considered to be frustrated if there exists a loop in which the product of all the couplings is negative. In a frustrated loop, if we fix an initial spin, and we try to fix the neighboring spins in the loop one after the other according to the sign of the couplings, at the end we will return to the initial spin and be forced to flip it. Which means that we are not able to find such a configuration that satisfies all couplings and thus it's impossible to minimize the energy of all couplings simultaneously and we have a proliferation of meta-stable states.

2.2 The pure *p*-spin

Another model, belonging to the class of glassy systems [12], that has been widely used in this framework is the pure spherical p-spin. Even though it presents a rugged energy landscape it is somewhat easy to solve, allowing us to obtain some insights by studying it.

2.2.1 Definition of the Model

The system is characterized by real spherical spins affected by long range p-body interactions, weighted by random interaction coefficients, thus the Hamiltonian $\mathcal{H}_p$ for a system of N spins reads as follows

$$\mathcal{H}_p = \sum_{i_1,\dots,i_p} J_{i_1,\dots,i_p}\, \sigma_{i_1}(t) \cdots \sigma_{i_p}(t) \qquad\qquad \sum_{i=1}^{N} \sigma_i^2 = N$$

Notice that we will always use the following shortcut notation: $\sum_{i_1,\dots,i_p} = \sum_{1\le i_1<\cdots<i_p\le N}$ and that we will drop the time-dependence of the variables most of the time to restore it only when necessary. The constraint on the right is the spherical constraint and we assume that the couplings J are given by independent Gaussian variables with mean and variance given by the relation below.

$$\langle J^2_{i_1,\dots,i_p} \rangle = \frac{p!}{2N^{p-1}} \qquad\qquad \langle J_{i_1,\dots,i_p} \rangle = 0$$

The value of the variance is chosen in such a way that the energy is finite and it grows linearly with the number of particles in the system.

2.2.2 The Energy Landscape: complexity of local minima

As we anticipated earlier, in high-dimension several local minima may appear and when this happens our descent gets severely impacted. In the limit of an infinitely large number of particles, this particular system can be studied analytically and, in particular, we can look directly at the stationary points of the Hamiltonian to understand what is going on. These points verify the following conditions

$$\frac{\partial \mathcal{H}}{\partial \sigma_i} + \mu\, \sigma_i = 0 \qquad\qquad \mu = -\frac{1}{N} \sum_i \sigma_i \frac{\partial \mathcal{H}}{\partial \sigma_i}$$

Where μ, the radial reaction force, is a Lagrange multiplier that imposes the spherical constraint on the spins and that takes the value on the right when the point is a stationary one. What emerges is that the number of stationary points $\mathcal{N}$ increases exponentially with the energy up to a certain value E_{th} [13] and in particular its functional form is the following $\mathcal{N}(E) \sim e^{N\,\Sigma(E)}$ [14]. Σ is called configurational entropy or complexity, and its behavior is shown schematically in Figure 1.

But what happens when we want to use a gradient descent algorithm to find the minimal energy achievable? As we try to make our way toward lower energies, we realize that the number of minima at E_{th} is so numerous that, in the macroscopic limit, the probability of

reaching lower energy values becomes vanishing and thus our descent gets stuck over this threshold value.

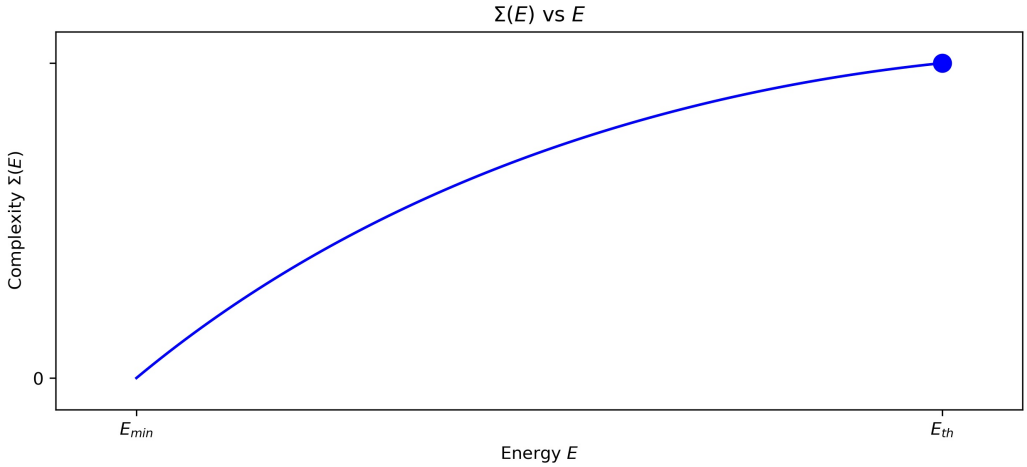

Figure 1: Relation between the complexity of the system and its energy in the case of the pure p-spin. The large circle highlights the energy value E_{th} for which we have stationary points with maximal complexity

2.2.3 Gradient Descent Dynamics Analysis

Even though the descent is not easy, the dynamics can be described analytically thanks to the Crisanti-Horner-Sommers-Cugliandolo-Kurchan equations in spin glass theory [15]. For this reason it appears to be a suitable field to analyze and compare the performance of GD and other techniques. Later on we will derive directly the integro-differential equations that allow us to describe the gradient descent evolution of the energy of such systems but, to put in the right frame the discussion, we will show preemptively some results.

Firstly, we should remind that when we use a gradient descent algorithm we have to choose an initial condition (or configuration) of our system and, if we do not have any insight that can help us with our decision, we are forced to choose a random one. In our case, and from the perspective of a physicist, this can be simply translated as choosing a random configuration taken from the system at an infinite temperature. Following the similarity with temperature, it may seem reasonable to choose a configuration associated to a lower temperature; indeed it makes sense, as we will start from a lower energy and our descent should be smoother and easier [16]. Therefore, we can initially equilibrate the system with a thermostat that is set at a temperature $T_{in} = 1/\beta_{in} < \infty$. But it is important to stress that we cannot choose a temperature that is too low, such as $0K$; it has to stay above a constant, the 'Mode Coupling Temperature' T_{MCT}, because at lower values we would not be able to

generate the configuration in a polynomial time and thus the results cannot be checked in real world experiments.

In the case where $p = 3$, following the theory of Cugliandolo-Kurchan, we are able to compute the threshold using the Formula 1 below as $E_{th} = -2/\sqrt{3} \simeq -1.1547$ [17, 18]. In our numerical simulations, where we consider multiple temperature values, we see that the system moves asymptotically toward E_{th}, confirming the prediction we've made previously. Note that each value of β is compared to $\beta_{MCT} = 1/T_{MCT} \simeq 1.24198$.

$$E_{th} = \frac{-f'(1)^2 + f(1)[f'(1) - f''(1)]}{f'(1)\sqrt{f''(1)}} \qquad \text{with} \qquad f(q) = \frac{1}{2}q^3 \tag{1}$$

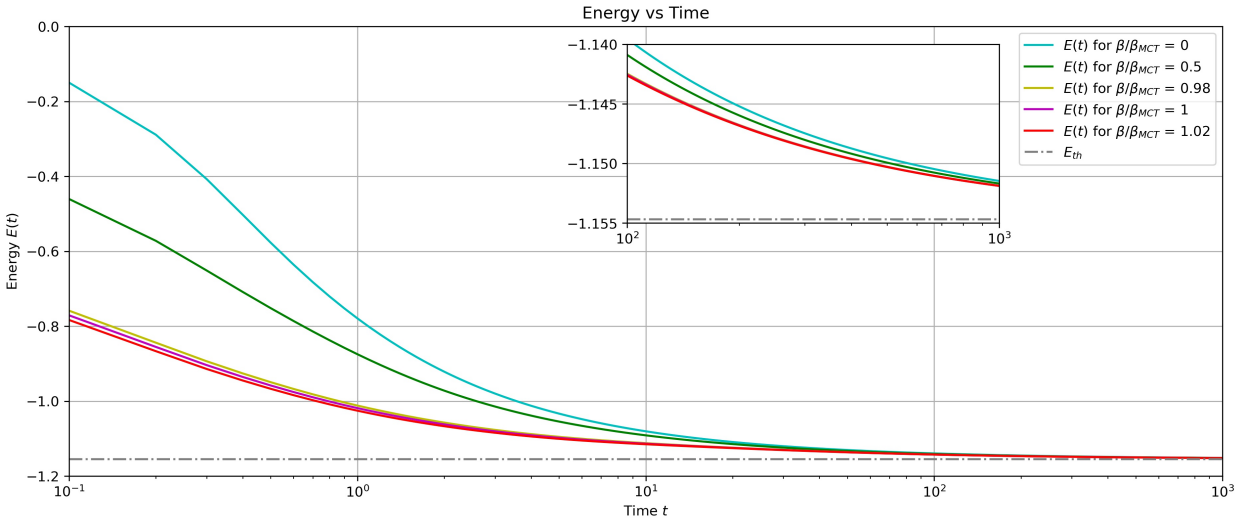

Figure 2: Gradient descent for the pure p-spin with p=3. The grid has been discretized with $N = 10000$ points and a time-step of $dt = 0.1$ was used. Different values of β have been chosen, but we can see that the long-time behavior is qualitatively the same.

3 The mixed p-spin model

The pure p-spin model has been been a topic of research for a long time and we know almost everything we are interested into; at the same time, we thought that its properties could be considered as universal and that its behavior would be shared by other similar problems, like the mixed p-spin but it has been shown that it's not always the case [18]. A mixed p-spin system is structurally identical to the pure one, the only difference is that in the former we have to consider interactions that can affect a different number of spins simultaneously, that means multiple p values, and hence we have:

$$\mathcal{H}_p = \sum_p \sum_{i_1,\dots,i_p} J_{i_1,\dots,i_p}\, \sigma_{i_1}(t) \cdots \sigma_{i_p}(t) \qquad\qquad \sum_i \sigma_i^2 = N$$

In these new settings, E_{th} is no longer the energy with the most numerous stationary states but instead it corresponds to the energy where the stationary states dominate the complexity. Interestingly, here we are able to cross the threshold barrier by modifying the initial conditions, and therefore by choosing wisely the temperature of the thermal bath. What we can witness is that for β small enough the descent shows a similar behavior to the one found beforehand: we still reach asymptotically the E_{th} of the system, whose value is $E_{th} = -\frac{71}{42}$ (which can be derived by using Formula 1 while considering $f(q) = \frac{1}{2}(q^3 + q^4)$). On the other hand, for β close to the mode-coupling one, the system does not forget its initial condition [19], even after a long time, and we are able to cross the threshold that was previously considered unsurpassable by the Cugliandolo-Kurchan argument. That behavior is clearly shown in the Figure 3; where the small inset, representing a vertical magnification at higher time t, allows us to see the crossing of the "barrier" which confirm the results recently found by Folena et al. [18].

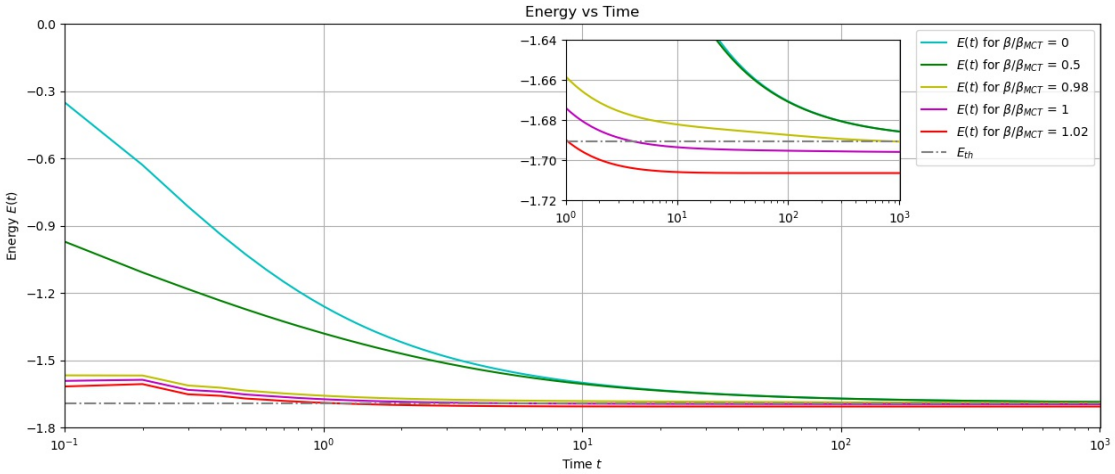

Figure 3: Gradient descent for a mixed p-spin model ($p = 3, 4$) with different values of β; where the gray dotted line represents the threshold. The grid has been discretized with $N = 10000$ points and a time step of $dt = 0.1$.

4 Integro-Differential Equations

4.1 Dynamical Mean-Field Equations

Considering that a wise choice of the initial conditions (in our case by changing the value of β) allows us to surpass the threshold energy, one may think if there are other ways to reach the same result. To do so we will take into consideration a particular mixed p-spin system with $p = 3, 4$ which presents a function v_t in front of the 4-spin term. This function will be exploited to reach the aforementioned aim in Section 5, in this propaedeutic part it won't play an important role.
Using the notation previously stated, its Hamiltonian is expressed as follows

$$\mathcal{H} = \sum_{i,j,k} J_{i,j,k}\, \sigma_i\sigma_j\sigma_k + v_t \sum_{i,j,k,l} J_{i,j,k,l}\, \sigma_i\sigma_j\sigma_k\sigma_l \qquad\qquad \sum_i \sigma_i^2 = N \tag{2}$$

To perform the standard gradient descent we consider the Langevin dynamic of each spin with no additive noise term:

$$\dot{\sigma_i} = -\mu_t\sigma_i - \frac{\partial\mathcal{H}}{\partial\sigma_i} = -\mu_t\sigma_i - \sum_{j,k\neq i} J_{i,j,k}\sigma_j\sigma_k + v_t \sum_{j,k,l\neq i} J_{i,j,k,l}\sigma_j\sigma_k\sigma_l$$

To solve this set of equations we will follow a perturbative approach, also called Dynamical Cavity Method [20]; we add to the system another spin, σ_0, and then consider the effect it has on the others.
First of all, we look at the new expression for the "initial" spins $\sigma_{i\neq 0}$:

$$\dot{\sigma_i} = -\mu_t\sigma_i - \sum_{j,k\neq i,0} J_{i,j,k}\sigma_j\sigma_k + v_t \sum_{j,k,l\neq i,0} J_{i,j,k,l}\sigma_j\sigma_k\sigma_l + H_i \tag{3}$$

$$\text{perturbation} \quad H_i = -\sum_{k\neq 0} J_{0,i,k}\sigma_i\sigma_k - v_t \sum_{k,l\neq 0} J_{0,i,k,l}\sigma_i\sigma_k\sigma_l \tag{4}$$

As we can see from Eq. (3), $\dot{\sigma_i}$ changes from the non-perturbed case only by an additive term H_i (that is exactly the perturbation we are adding).

On the other hand, σ_0's behavior is described by the following differential equation:

$$\dot{\sigma_0} = -\mu_t\sigma_0 - \sum_{j,k\neq 0} J_{0,j,k}\sigma_j\sigma_k + v_t \sum_{j,k,l\neq 0} J_{0,j,k,l}\sigma_j\sigma_k\sigma_l$$

Following our perturbative scheme it is natural to express each σ_i term as a sum of the non-perturbed value and the perturbation that the presence of σ_0 induces on these. Up to

linear order we can write:

$$\sigma_i = \sigma_i^0 + \delta\sigma_i^0 \qquad\qquad \delta\sigma_i^0 \simeq \int_0^t ds \frac{\delta\sigma_i^0(t)}{\delta H_i(s)} H_i(s) \tag{5}$$

If we now plug back the previous relations into Eq. (5), we will have multiple terms of the form $\sigma_k^0\ \delta\sigma_j^0$. We can notice that these terms will all have the same contribution due to symmetry properties. Thus we retrieve the following expression for the evolution of σ_0:

$$\dot{\sigma_0} = -\mu_t\sigma_0 - \underbrace{\Big[\sum_{j,k\neq 0} J_{0,j,k}\sigma_j^0\sigma_k^0 + v_t \sum_{j,k,l\neq 0} J_{0,j,k,l}\sigma_j^0\sigma_k^0\sigma_l^0\Big]}_{\eta_t} - \underbrace{2\sum_{j,k\neq 0} J_{0,j,k}\sigma_k^0\delta\sigma_j^0}_{A} - \underbrace{3v_t \sum_{j,k,l\neq 0} J_{0,j,k,l}\sigma_k^0\sigma_l^0\delta\sigma_j^0}_{B} \tag{6}$$

Where η_t can be thought as a noise term that contains the unperturbed variables $\sigma_{i\neq 0}$.

Hereby we will assume that we can neglect terms sub-leading in N: $\sum_{i_1,\dots,i_k} \approx \frac{1}{k!}\sum_{i_1}\cdots\sum_{i_k}$

Calculation of the 3-body term of Eq. (6):

$$\begin{aligned}
A &= \sum_{j,k\neq 0} J_{0,j,k}\sigma_k^0(t)\delta\sigma_j^0(t) \simeq \sum_{j,k\neq 0} J_{0,j,k}\sigma_k^0(t)\int ds\ \frac{\delta\sigma_j^0(t)}{\delta H_j(s)} H_j(s) = \\
&= -\sum_{j,k\neq 0} J_{0,j,k}\sigma_k^0(t)\int ds\ \frac{\delta\sigma_j(t)}{\delta H_j(s)} \sum_{l\neq 0} J_{0,j,l}\sigma_0(s)\sigma_l^0(s) = \\
&= -\sum_{j,k\neq 0}\sum_{l\neq 0} J_{0,j,k}J_{0,j,l}\int ds\ \sigma_k^0(t)\sigma_l^0(s)\frac{\delta\sigma_j(t)}{\delta H_j(s)}\sigma_0(s) = \\
&= -\sum_{j,k\neq 0} J_{0,j,k}^2\int ds\ \sigma_k^0(t)\sigma_k^0(s)\frac{\delta\sigma_j(t)}{\delta H_j(s)}\sigma_0(s) = \\
&\simeq -\frac{3!}{2N^2}\sum_{j,k\neq 0}\int ds\ \sigma_k^0(t)\sigma_k^0(s)\frac{\delta\sigma_j(t)}{\delta H_j(s)}\sigma_0(s) = \\
&\simeq -\frac{3!}{2\cdot 2!}\int ds\ \Big[\frac{1}{N}\sum_{k\neq 0}\sigma_k^0(t)\sigma_k^0(s)\Big]\Big[\frac{1}{N}\sum_{j\neq 0}\frac{\delta\sigma_j^0(t)}{\delta H_j(s)}\Big]\sigma_0(s) = \\
&= -\frac{3}{2}\int ds\ C(t,s)R(t,s)\sigma_0(s)
\end{aligned}$$

Calculation of the 4-body term of Eq. (6):

$$
\begin{aligned}
B &= v_t \sum_{j,k,l\neq 0} J_{0j,k,l}\sigma_k^0(t)\sigma_l^0(t)\delta\sigma_j^0(t) \simeq \sum_{j,k,l\neq 0} v_t J_{0j,k,l}\sigma_k^0(t)\sigma_l^0(t)\int ds\,\frac{\delta\sigma_j(t)}{\delta H_j(s)}H_j(s) = \\
&= -\sum_{j,k,l\neq 0} v_t J_{0j,k,l}\sigma_k^0(t)\sigma_l^0(t)\int ds\,\frac{\delta\sigma_j(t)}{\delta H_j(s)}\sum_{m,n\neq 0} v_s J_{0,j,m,n}\sigma_0(s)\sigma_m^0(s)\sigma_n^0(s) = \\
&= -\sum_{j,k,l\neq 0}\sum_{m,n\neq 0} J_{0j,k,l}J_{0,j,m,n}\int ds\, v_t v_s\sigma_k^0(t)\sigma_l^0(t)\sigma_m^0(s)\sigma_n^0(s)\frac{\delta\sigma_j(t)}{\delta H_j(s)}\sigma_0(s) = \\
&= -\sum_{j,k,l\neq 0} J_{0j,k,l}^2\int ds\, v_t v_s\sigma_k^0(t)\sigma_l^0(t)\sigma_m^0(s)\sigma_n^0(s)\frac{\delta\sigma_j(t)}{\delta H_j(s)}\sigma_0(s) = \\
&\simeq -\frac{4!}{2N^3}\sum_{j,k,l\neq 0}\int ds\, v_t v_s\sigma_k^0(t)\sigma_l^0(s)\sigma_m^0(s)\sigma_n^0(s)\frac{\delta\sigma_j(t)}{\delta H_j(s)}\sigma_0(s) = \\
&\simeq -\frac{4!}{2\cdot 3!}\int ds\, v_t v_s\Big[\frac{1}{N}\sum_{k\neq 0}\sigma_k^0(t)\sigma_k^0(s)\Big]\Big[\frac{1}{N}\sum_{l\neq 0}\sigma_l^0(t)\sigma_l^0(s)\Big]\Big[\frac{1}{N}\sum_{j\neq 0}\frac{\delta\sigma_j(t)}{\delta H_j(s)}\Big]\sigma_0(s) = \\
&= -6\int ds\, v_t v_s C(t,s)^2 R(t,s)\sigma_0(s)
\end{aligned}
$$

Where we have defined the functions C and R as follows:

$$
C(t,s) = \frac{1}{N}\sum_{i\neq 0}\sigma_i^0(t)\sigma_i^0(s) = \langle\sigma_i^0(t)\sigma_i^0(s)\rangle \qquad\qquad R(t,s) = \frac{1}{N}\sum_{i\neq 0}\frac{\delta\sigma_i(t)}{\delta H_i(s)} = \langle\frac{\delta\sigma_i(t)}{\delta H_i(s)}\rangle
$$

Finally, we can simplify the notation by defining a couple of two-variable functions: M_3^R and M_4^R

$$
A = \sum_{j,k\neq 0} J_{0j,k}\sigma_k^0(t)\delta\sigma_j^0(t) \simeq -\int_0^t ds\, M_3^R(t,s)\sigma_0(s) \qquad\qquad M_3^R(t,s) \coloneqq 3C(t,s)R(t,s)
$$

$$
B = v_t\sum_{j,k,l\neq 0} J_{0,j,k,l}\sigma_k^0(t)\sigma_l^0(t)\delta\sigma_j^0(t) = -\int_0^t ds M_4^R(t,s)\sigma_0(s) \qquad M_4^R(t,s) \coloneqq -6v_t v_s C(t,s)R(t,s)
$$

And in the end we are able to write Eq. (6) as follows:

$$
\dot{\sigma_0} = -\mu_t\sigma_0 - \eta_t + \int_0^t ds M^R(t,s)\sigma_0(s) \qquad\qquad M^R(t,s) \coloneqq M_3^R(t,s) + M_4^R(t,s)
$$

The last term we have to characterize is η_t, which can be considered as a bath produced by

the initial variables $\sigma_{i\neq 0}$ to the added one σ_0.

$$\eta_t = \sum_{j,k\neq 0} J_{0,j,k}\sigma_j^0\sigma_k^0 + v_t \sum_{j,k,l\neq 0} J_{0,j,k,l}\sigma_j^0\sigma_k^0\sigma_l^0$$

This Gaussian noise has zero mean $\langle\eta_t\rangle|_J = 0$ (average over the couplings J) and variance given by:

$$\begin{aligned}
\langle\eta_t\eta_s\rangle|_J &= \sum_{j,k\neq 0}\sum_{l,m\neq 0}\langle J_{0j,k}J_{0,l,m}\rangle\sigma_j^0(t)\sigma_k^0(t)\sigma_l^0(s)\sigma_m^0(s) + \\
&\quad + \sum_{j,k,l\neq 0}\sum_{m,n,p\neq 0}\langle J_{0j,k,l}J_{0m,n,p}\rangle\sigma_j^0(t)\sigma_k^0(t)\sigma_l^0(t)\sigma_m^0(s)\sigma_n^0(s)\sigma_p^0(s)v_tv_s = \\
&\simeq \frac{3!}{2N^2}\sum_{j,k\neq 0}\sigma_j^0(t)\sigma_k^0(t)\sigma_j^0(s)\sigma_k^0(s) + \frac{4!}{2N^3}\sum_{j,k,l\neq 0}\sigma_j^0(t)\sigma_k^0(t)\sigma_l^0(t)\sigma_j^0(s)\sigma_k^0(s)\sigma_l^0(s)v_tv_s = \\
&\simeq \frac{3}{2N^2}\sum_{j\neq 0}\sum_{k\neq 0}\sigma_j^0(t)\sigma_k^0(t)\sigma_j^0(s)\sigma_k^0(s) + \\
&+ \frac{4}{2N^3}\sum_{j\neq 0}\sum_{k\neq 0}\sum_{l\neq 0}\sigma_j^0(t)\sigma_k^0(t)\sigma_l^0(t)\sigma_j^0(s)\sigma_k^0(s)\sigma_l^0(s)v_tv_s = \\
&= \frac{3}{2}C(t,s)^2 + 2C(t,s)^3v_tv_s \coloneqq M^C(t,s)
\end{aligned}$$

Gathering all these results, we are finally able to write down the equations for the evolution of the correlation function $C(t,t')$ and the response function $R(t,t')$. Later we will provide also the energy as a function of C, R and μ.
If we assume that, in the macroscopic limit, the new variable $\sigma_0(t)$ behaves like the other ones, we are able to write down the following equations:

$$\begin{aligned}
C(t,t') &= \langle\sigma_0(t)\sigma_0(t')\rangle \\
\frac{\partial C(t,t')}{\partial t} &= -\mu(t)C(t,t') + \langle\eta(t)\sigma_0(t')\rangle + \int_0^t dsM^R(t,s)\langle\sigma_0(s)\sigma_0(t')\rangle = \\
&= -\mu(t)C(t,t') + \langle\eta(t)\sigma_0(t')\rangle + \int_0^t dsM^R(t,s)C(t',s)
\end{aligned}$$

The second term needs a careful treatment. After we explicit the average over the noise, that is assumed to be Gaussian and governed by the kernel M^C, we can apply the Novikov

theorem and an integration by parts to reach a compact result

$$\begin{aligned}\langle\eta(t)\sigma_0(t')\rangle &= \int D[\eta(t)]\, e^{-\frac{1}{2}\iint d\tau d\tau' \eta(\tau) M_C^{-1}(\tau,\tau')\eta(\tau')}\eta(t)\sigma_0(t') = \\ &= -\int_0^{t'} ds \int D[\eta(t)]\, \sigma_0(t')\frac{\delta}{\delta\eta(s)} e^{-\frac{1}{2}\iint d\tau d\tau' \eta(\tau) M_C^{-1}(\tau,\tau')\eta(\tau')} M^C(t,s) = \\ &= \int_0^{t'} ds\, \langle \frac{\delta\sigma_0(t')}{\delta\eta(s)} M^C(t,s)\rangle = \int_0^{t'} ds\, R(t',s) M^C(t,s)\end{aligned}$$

Where in the first step we have performed an integration by parts with respect to the noise η(s).

4.2 Final equations

$$\mu_t = \int_0^t ds\, M_C(t,s)R(t',s) + \int_0^t ds\, M_R(t,s)C(t',s) \tag{7}$$

$$\partial_t C(t,t') = -\mu_t C(t,t') + \int_0^{t'} ds\, M_C(t,s)R(t',s) + \int_0^t ds\, M_R(t,s)C(t',s) \tag{8}$$

$$\partial_t R(t,t') = \partial_t \langle \frac{\delta\sigma_o(t)}{\delta\eta(t')}\rangle = \delta(t,t') - \mu_t R(t,t') + \int_{t'}^t ds\, M_R(t,s)R(s,t') \tag{9}$$

Notice that the last integral goes from t' to t because the $R(s,t') = 0$ for $s < t'$, due to causality.

Where $C(t,t) = 1$, $R(t,t' \to t^+) = 1$ and $R(t,t') = 0$ for $t \leq t'$.

If we let $f_k(t,s) = \frac{1}{2}q^k$ we can write, in a easy and compact way, the two memory kernels M_C and M_R as:

$$\begin{aligned}M_C(t,s) &= \frac{3}{2}C^2(t,s) + \frac{4}{2}C^3(t,s)v_t v_s = {f'}_3(C(t,s)) + {f'}_4(C(t,s))v_t v_s \\ M_R(t,s) &= 3C(t,s)R(t,s) + 6C^2(t,s)R(t,s)v_t v_s = {f''}_3(C(t,s)) + {f''}_4(C(t,s))v_t v_s\end{aligned}$$

4.3 Energy

Up to now we have been able to characterize the evolution of the correlation and response function. Now we can apply the information gathered in order to express the energy of the system using these quantities. First of all we define a normalized energy function and then

we will compute the terms that will arise separately (but in the same manner):

$$e(t) = \frac{1}{N}\langle \mathcal{H} \rangle = \frac{1}{N}\langle \mathcal{H}_3 + \mathcal{H}_4 \rangle = e_3(t) + e_4(t) \tag{10}$$

$$\mathcal{H}_3 = \sum_{ijk} J_{ijk}\sigma_i\sigma_j\sigma_k; \qquad \mathcal{H}_4 = v_t \sum_{ijkl} J_{ijkl}\sigma_i\sigma_j\sigma_k\sigma_l;$$

Recall Eq. (3):

$$\dot{\sigma}_i = \frac{\partial \mathcal{H}}{\partial x} + \eta_t = -\mu_t\sigma_i - \sum_{j,k\neq i,0} J_{i,j,k}\sigma_j\sigma_k + v_t \sum_{j,k,l\neq i,0} J_{i,j,k,l}\sigma_j\sigma_k\sigma_l + H_i \tag{11}$$

Using the same technique followed by Castellani and Cavagna [21], we can beforehand use an odd representation of the number 1:

$$1 \coloneqq Z = \int D[\sigma]P(\sigma) = \iint D[\sigma]D[\hat{\sigma}]e^{S(\sigma,\hat{\sigma})}$$

Where:

$$S(\sigma,\hat{\sigma}) = -\frac{1}{2}\iint dtdt'\hat{\sigma}(t)D(t,t')\hat{\sigma}(t') + i\int dt\hat{\sigma}(t)\Big[\partial_t\sigma + \partial_\sigma\mathcal{H}\Big] \coloneqq -\frac{1}{2}\hat{\sigma}D\hat{\sigma} + i\hat{\sigma}\Big[\partial_t\sigma + \partial_\sigma\mathcal{H}\Big]$$

And by using the properties of the (Gaussian) distribution of the noise terms (in this case for $p = 3, 4$):

$$P(J_{i_1,\dots,i_p}) \sim e^{-{J_{i_1,\dots,i_p}}^2/2} \qquad J_{i_1,\dots,i_p}e^{-{J_{i_1,\dots,i_p}}^2/2} \sim -\frac{\partial}{\partial_{J_{i_1,\dots,i_p}}}e^{-{J_{i_1,\dots,i_p}}^2/2}$$

We are able to compute both energy terms of Eq. (10)

First term:

$$e_3(t) = \frac{1}{N}\langle \iint D[\sigma]D[\hat{\sigma}]e^{S(\sigma,\hat{\sigma})} \sum_{ijk} J_{ijk}\sigma_i\sigma_j\sigma_k \rangle =$$

$$= -\frac{1}{N}\langle \iint D[\sigma]D[\hat{\sigma}]e^{S(\sigma,\hat{\sigma})} \int dt' \sum_{i'} \sum_{j'k'} J_{i'j'k'}\hat{\sigma}_{i'}(t')\sigma_{j'}(t')\sigma_{k'}(t') \sum_{ijk} J_{ijk}\sigma_i(t)\sigma_j(t)\sigma_k(t) \rangle =$$

$$= -\frac{1}{N} \iint D[\sigma]D[\hat{\sigma}]e^{S(\sigma,\hat{\sigma})} \int dt' \sum_{i'} \sum_{j'k'} \sum_{ijk} \langle J_{i'j'k'}J_{ijk}\rangle \hat{\sigma}_{i'}(t')\sigma_{j'}(t')\sigma_{k'}(t')\sigma_i(t)\sigma_j(t)\sigma_k(t) =$$

$$= -\frac{1}{2N} \int dt' \frac{3!}{2N^2} \sum_i \hat{\sigma}_i(t')\sigma_i(t) \sum_j \sigma_j(t')\sigma_j(t) \sum_k \sigma_k(t')\sigma_k(t) =$$

$$= -\frac{3}{2} \int dt' \frac{1}{N} \sum_i \hat{\sigma}_i(t')\sigma_i(t) \frac{1}{N} \sum_j \sigma_j(t')\sigma_j(t) \frac{1}{N} \sum_k \sigma_k(t')\sigma_k(t) =$$

$$= -\frac{3}{2} \int dt' R(t,t')C(t,t')^2 = -\int dt' f_3'(C_{t,t'})R(t,t')$$

Second term:

$$e_4(t) = \frac{1}{N} \iint D[\sigma]D[\hat{\sigma}]e^{S(\sigma,\hat{\sigma})} \sum_{ijkl} J_{ijkl}\sigma_i\sigma_j\sigma_k\sigma_l =$$

$$= -\frac{1}{N}\langle \iint D[\sigma]D[\hat{\sigma}]e^{S(\sigma,\hat{\sigma})} \int dt' \sum_{i'} \sum_{j'k'l'} v_{t'} J_{i'j'k'l'}\hat{\sigma}_{i'}(t')\sigma_{j'}(t')\sigma_{k'}(t')\sigma_{l'}(t') \sum_{ijkl} J_{ijkl}\sigma_i(t)\sigma_j(t)\sigma_k(t)\sigma_l(t) \rangle =$$

$$= -\frac{1}{N} \iint D[\sigma]D[\hat{\sigma}]e^{S(\sigma,\hat{\sigma})} \int dt' v_{t'} \sum_{i'} \sum_{j'k'l'} \sum_{ijkl} \langle J_{i'j'k'l'}J_{ijkl}\rangle \hat{\sigma}_{i'}(t')\sigma_{j'}(t')\sigma_{k'}(t')\sigma_{l'}(t')\sigma_i(t)\sigma_j(t)\sigma_k(t)\sigma_l(t) =$$

$$= -\frac{1}{3!N} \int dt' v_{t'} \frac{4!}{2N^3} \sum_i \hat{\sigma}_i(t')\sigma_i(t) \sum_j \sigma_j(t')\sigma_j(t) \sum_k \sigma_k(t')\sigma_k(t) \sum_l \sigma_l(t')\sigma_l(t) =$$

$$= -6 \int dt' v_{t'} \frac{1}{N} \sum_i \hat{\sigma}_i(t')\sigma_i(t) \frac{1}{N} \sum_j \sigma_j(t')\sigma_j(t) \frac{1}{N} \sum_k \sigma_k(t')\sigma_k(t) \frac{1}{N} \sum_l \sigma_l(t')\sigma_l(t) =$$

$$= -6 \int dt' v_{t'} R(t,t')C(t,t')^3 = -\int dt' v_{t'} f_4'(C_{t,t'})R(t,t')$$

And finally:

$$e(t) = -\int_0^t ds \Big[f_3'(C_{t,s}) + v_s f_4'(C_{t,s}) \Big] R_{t,s} \tag{12}$$

4.4 Discrete Equations

Everything considered, we were able to write down a set of integro-differential equations for 7, 8, 9, 12; to tackle this problem at a numerical level we have to discretize all the functions we are handling and the calculations we have to perform. For that purpose we are going to consider the 2-variables function, C and R, as matrices of size N; similarly μ and the energy will be discretized as arrays of length N.
All the derivatives will be approximated using the finite difference approximation and the integrals as left-Riemann sums as shown below:

$$f(t+dt) = f(t) + dt \cdot f'(t) \qquad \int_0^t f(x)dx = \Delta x \sum_{i=0}^{N-1} f(x_i)$$

If we let dt $= \Delta t$, the four equations read:

$$\mu_i = \Delta t \sum_{s=0}^{i-1} \Big[M^C_{i,s} R_{i,s} + M^R_{i,s} C_{i,s} \Big] + \beta [f_3'(C_{i,0}) + v_i f_4'(C_{i,0})] C_{i,0}$$

$$C_{i+1,j} = C_{i,j} + \Delta t \Big[-\mu_i C_{i,j} + \Delta t \Big[\sum_{s=0}^{j-1} M^C_{i,s} R_{j,s} + \sum_{s=0}^{i-1} M^R_{i,s} C_{j,s} \Big] + \beta [f_3'(C_{i,0}) + v_i f_4'(C_{i,0})] C_{j,0} \Big]$$

$$R_{i+1,j} = R_{i,j} + \Delta t \Big[-\mu_i R_{i,j} + \delta_{i,j} + \Delta t \sum_{s=j}^{i-1} M^R_{i,s} R_{s,j} \Big]$$

$$e_i = -\Delta t \sum_{s=0}^{i-1} \Big[f_3'(C_{i,s}) + v_s f_4'(C_{i,s}) \Big] R_{i,s} - \beta \Big[f_3'(C_{i,0}) + v_i f_4'(C_{i,0}) \Big]$$

4.5 Pseudo-code

Upon initializing all the variables to a 0-value, we can summarize the algorithm we have used with the following pseudo-code. The results have been shown and described in Section 3.

Algorithm 1: Gradient Descent

Initialization:

$C(0,0) = 1$

$\mu(0) = \text{compute.}\mu(0)$

Loops:

for $i = 1$ **to** $N-1$ **do**

  foreach $j \in \{0, 1, \dots, i\}$ **do**

    $R(i,j) = R(i-1,j) + dt \cdot \partial_i R(i-1,j)$

    $C(i,j) = C(i-1,j) + dt \cdot \partial_i C(i-1,j)$

    $C(j,i) = C(i,j)$

  $\mu(i) = \text{compute.}\mu(i)$

for $i = 0$ **to** $N-1$ **do**

  energy(i) = compute.energy(i)

5 Optimization of gradient descent dynamics in the mixed p-spin

5.1 Lagrangian

What we would like to do now is to optimize the gradient descent by changing the shape of the function $v(t)$ in such a way that at the end of our simulation we reach a lower energy value compared to the standard algorithm. To do so, we add a constraint, and thus a cost, to the function v compared to its original (constant) value of 1.
Our goal can be summarized as follows:

$$\text{find } v_t = \underset{v_t}{\text{argmin}} \left[e(t^*) + \frac{\Delta}{2} \int_0^{t^*} dt \, (v_t - 1)^2 \right] \Big|_{\dot{C},\dot{R},\mu}$$

Where t^* is the final time that we want to reach in our simulation.

To complete this task we have to write down the complete Lagrangian of the dynamical evolution, were we have to add 3 more variables: P^C, P^R and λ; these 3 are variables

associated to the ones we have previously used in the forward-only protocol: C, R and μ.

$$\begin{aligned}\mathcal{L} =& \frac{\Delta}{2}\int_0^{t^*} dt\,(v_t-1)^2 - \int_0^{t^*} dt\Big[f_3'(C_{t^*,s}) + v_s f_4'(C_{t^*,s})\Big]R_{t^*,s} - \beta\Big[f_3(C_{t^*,0}) + f_4(C_{t^*,0})\Big] + \\ &+ \int_0^{t^*} dt \int_0^t dt' P^C_{t,t'}\Big[\partial_t C_{t,t'} + \mu_t C_{t,t'} - \int_0^t ds M^R_{t,s} C_{t',s} - \int_0^{t'} ds M^C_{t,s} R_{t',s}\Big] - \beta\hat{f}'(C_{t,0})C_{t',0}\Big] + \\ &+ \int_0^{t^*} dt \int_0^t dt' P^R_{t,t'}\Big[\partial_t R_{t,t'} + \mu_t R_{t,t'} - \delta(t-t') - \int_{t'}^t ds M^C_{t,s} R_{s,t'}\Big] + \\ &+ \int_0^{t^*} dt\lambda_t\Big[\mu_t - \int_0^t ds(M^C_{t,s}R_{t,s} + M^R_{t,s}C_{t,s}) - \beta\hat{f}'(C_{t,0})C_{t,0}\Big] \end{aligned} \tag{13}$$

To have a simpler notation, we have defined: $\hat{f}(C_{i,j}) = f_3(C_{i,j}) + v_i f_4(C_{i,j})$ and the partial derivative is made with respect to the argument of $\hat{f}$: $C_{i,j}$.

To minimize the Lagrangian, and therefore the energy reached at the end of the simulation, $\mathcal{L}$ has to satisfy the following conditions:

$$\begin{aligned} &\frac{\delta\mathcal{L}}{\delta P^C_{i,j}} = 0 \qquad &&\frac{\delta\mathcal{L}}{\delta P^R_{i,j}} = 0 \qquad &&\frac{\delta\mathcal{L}}{\delta\lambda_i} = 0 \\ &\frac{\delta\mathcal{L}}{\delta C_{i,j}} = 0 \qquad &&\frac{\delta\mathcal{L}}{\delta R_{i,j}} = 0 \qquad &&\frac{\delta\mathcal{L}}{\delta\mu_i} = 0 \qquad &&\frac{\delta\mathcal{L}}{\delta v_i} = 0 \end{aligned}$$

The first three equations (the ones in the upper part) can be ignored as they would simply give us back the original equations obtained in the previous sections; those we will focus on are the ones on the bottom part. First of all, we remind that $C_{i,i} = 1, R_{i,i} = 0$, thus there is no need to perform the functional derivatives over these parameter (being constants in fact). Moreover the two-variable function C is symmetric if we swap the arguments and the function $R_{i,j}$ admits non-zero values only for $i > j$. This simplifies greatly our analysis given that we can consider only the case where $\boldsymbol{i > j}$ and get information for all possible couples (i, j)

5.2 Final Equations

The direct computation is simple but lengthy and for this reason these has been performed directly in the Appendix A; here we present the final results.

5.2.1 Initial conditions

Initial condition for P^C:

$$P^C_{t^*-1,i} = \Big[f_3''(C_{t^*,i}) + v_i f_4''(C_{t^*,i})\Big] R_{t^*,i} + \beta\Big[f_3'(C_{t^*,0}) + f_4'(C_{t^*,0})\Big]\delta_{i,0}/\Delta t$$

Initial condition for P^R:

$$P^R_{t^*-1,i} = \Big[f_3'(C_{t^*,i}) + v_i f_4'(C_{t^*,i})\Big]$$

5.2.2 Backward Equations

Equation for λ:

$$\lambda_i = -\Delta t \sum_{t'=0}^{i} \Big[P^C_{i,t'} C_{i,t'} + P^R_{i,t'} R_{i,t'}\Big]$$

Evolution for P^R:

$$\begin{aligned} P^R_{i-1,j} = P^R_{i,j} - \Delta t \Bigg[& P^R_{i,j}\mu_i - \Delta t \sum_{t'=j+1}^{i} P^C_{i,t'}(\partial_R M^R_{i,j}) C_{t',j} - \Delta t \sum_{t'=0}^{j} P^C_{i,t'}(\partial_R M^R_{i,j}) C_{j,t'} + \\ & - \Delta t \sum_{t=i}^{t^*-1} P^C_{t,i} M^C_{t,j} - \Delta t \sum_{t'=0}^{j} P^R_{i,t'}(\partial_R M^R_{i,j}) R_{j,t'} - \Delta t \sum_{t=i+1}^{t^*-1} P^R_{t,j} M^R_{t,i} - \lambda_i \Big[M^C_{i,j} + (\partial_R M^R_{i,j}) C_{i,j}\Big] \Bigg] \end{aligned}$$

Evolution for P^C:

$$\begin{aligned} P^C_{i-1,j} = P^C_{i,j} - \Delta t \Bigg[& P^C_{i,j}\mu_i - \Delta t \sum_{t'=j+1}^{i} P^C_{i,t'}(\partial_C M^R_{i,j}) C_{t',j} - \Delta t \sum_{t=i}^{t^*-1} P^C_{t,i} M^R_{t,j} + \\ & - \Delta t \sum_{t'=0}^{j} P^C_{i,t'}(\partial_C M^R_{i,j}) C_{j,t'} - \Delta t \sum_{t=i+1}^{t^*-1} P^C_{t,j} M^R_{t,i} - \Delta t \sum_{t'=j+1}^{i} P^C_{i,t'}(\partial_C M^C_{i,j}) R_{t',j} + \\ & - \beta\delta_{j,0}\Big[\sum_{t'=0}^{i} P^C_{i,t'}\hat{f}''(C_{i,0}) C_{t',0} + \sum_{t=i}^{t^*-1} P^C_{t,i}\hat{f}'(C_{t,0}) + \lambda_i\Big(\hat{f}''(C_{i,0}) C_{i,0} + \hat{f}'(C_{i,0})\Big)/\Delta t\Big] + \\ & - \Delta t \sum_{t'=0}^{j} P^R_{i,t'}(\partial_C M^R_{i,j}) R_{j,t'} - \lambda_i\Big[(\partial_C M^C_{i,j}) R_{i,j} + (\partial_C M^R_{i,j}) C_{i,j} + M^R_{i,j}\Big] \Bigg] \end{aligned}$$

Equation for v:

$$\begin{aligned}
\Delta(v_i - 1) = {} & f_4'(C_{t^*,i})R_{t^*,i} + \beta \sum_{t'=0}^{i} P^C_{i,t'} f_4'(C_{i,0})C_{t',0} + \beta\lambda_i f_4'(C_{i,0})C_{i,0} + \\
& + \Delta t^2 \sum_{t=i+1}^{t^*-1} \sum_{t'=i+1}^{t} P^C_{t,t'} f_4''(C_{t,i})R_{t,i}C_{t',i}v_t + \Delta t^2 \sum_{t'=0}^{i} \sum_{s=0}^{t'-1} P^C_{i,t'} f_4''(C_{i,s})R_{i,s}C_{t',s}v_s + \\
& + \Delta t^2 \sum_{t=i+1}^{t^*-1} \sum_{t'=0}^{i} P^C_{t,t'} f_4''(C_{t,i})R_{t,i}C_{i,t'}v_t + \Delta t^2 \sum_{t'=0}^{i} \sum_{s=t'}^{i-1} P^C_{i,t'} f_4''(C_{i,s})R_{i,s}C_{s,t'}v_s + \\
& + \Delta t^2 \sum_{t=i+1}^{t^*-1} \sum_{t'=i+1}^{t} P^C_{t,t'} f_4'(C_{t,i})R_{t',i}v_t + \Delta t^2 \sum_{t'=0}^{i} \sum_{s=0}^{t'-1} P^C_{t,t'} f_4'(C_{i,s})R_{t',s}v_s + \\
& + \Delta t^2 \sum_{t=i+1}^{t^*-1} \sum_{t'=0}^{i} P^R_{t,t'} f_4''(C_{t,i})R_{t,i}R_{i,t'}v_t + \Delta t^2 \sum_{t'=0}^{i} \sum_{s=t'}^{i-1} P^R_{i,t'} f_4''(C_{i,s})R_{i,s}R_{s,t'}v_s + \\
& + \Delta t \sum_{t=i+1}^{t^*-1} \lambda_t f_4'(C_{t,i})R_{t,i}v_t + \Delta t\lambda_i \sum_{s=0}^{i-1} f_4'(C_{i,s})R_{i,s}v_s + \\
& + \Delta t \sum_{t=i+1}^{t^*-1} \lambda_t f_4''(C_{t,i})R_{t,i}C_{t,i}v_t + \Delta t\lambda_i \sum_{s=0}^{i-1} f_4''(C_{i,s})R_{i,s}C_{i,s}v_s
\end{aligned}$$

6 Numerical implementation

The previous set of coupled equations allows us to compute the new functions P^C, P^R and μ and thereafter we can use that information to estimate the function v_t. To do so we initialize accordingly the different variables at the step $t = t^*$, and then proceed backwardly toward $t = 0$. We have to notice thought that the new v_t we are able to compute is generated using the initial v_t, which in general will be different; for this reason we have to iterate the algorithm till the initial v and the final v are the same, up to an error. To quantify this error, we impose that the Frobenius norm of the difference between these two vectors v has to be smaller than a given value ϵ. Moreover, In order to avoid oscillations and a poor numerical behavior, at each iteration we update the function using a learning rate α that limits the update.
The algorithm can be simplified in these 4 steps (which are summarized in Algorithm 3):

- Compute the functions μ, C and R using the $v(t)$ found in the previous iteration.
- Propose a better $v_{\text{new}}(t)$ using the updated λ, P_C and P_R.

- Check if the $v(t)$ and $v_{\text{new}}(t)$ are (almost) the same: $\sqrt{\frac{1}{t^*}\int_0^{t^*} dt(v(t) - v_{\text{new}}(t))^2} < \epsilon$
- Update $v(t) = v(t) \cdot (1 - \alpha) + v_{\text{new}}(t) \cdot \alpha$.

Algorithm 2: Backward Algorithm

Initialization:

$P^C(t^* - 1, :) = \text{initialize}.P^C(t^*)$

$P^R(t^* - 1, :) = \text{initialize}.P^R(t^*)$

$\lambda(t^*) = \text{compute}.\lambda(t^*)$

Loops:

for $i = t^* - 1$ **to** 0 **do**

  $\lambda(i) = \text{compute}.\lambda(i)$

  foreach j **do**

    $P^C(i-1, j) = P^C(i, j) - dt \cdot (\partial_i PC(i, j))$

    $P^R(i-1, j) = P^R(i, j) - dt \cdot (\partial_i PR(i, j))$

for $i = 0$ **to** $N - 1$ **do**

  $v(i) = \text{compute}.v(i)$

Algorithm 3: Optimization Algorithm

$v = 1$

while $Norm > \epsilon$ **do**

  $\text{compute}(\mu, C, R)$ using v

  $\text{compute}(\lambda, P_C, P_R)$ using v

  $v_2 = \text{find.new}.v(\lambda, P_C, P_R)$

  $Norm = \text{sqrt}(\frac{1}{t^*}\int_0^{t^*} dt[v(t) - v_2(t)]^2)$

  $v_{\text{new}} = v \cdot (1 - \alpha) + v_2 \cdot \alpha$

  $v = v_{\text{new}}$

$\text{compute}(\mu, C, R)$ with $v = v_{\text{opt}}$

for $i = 0$ **to** $N - 1$ **do**

  $\text{optimal\_energy}(i) = \text{compute.energy}(\mu, C, R, v = v_{\text{opt}}, i)$

7 Numerical results

The sets of equations have been numerically simulated using the algorithms depicted previously. To have a good enough understanding of the model we considered different possible values of t^* and for each one, 3 different lengths of the grid have been chosen ($N = 500, 1000, 2000$). This ensures that the simulation is stable when using different time-steps and that the results are consistent. In our case, we also decided to set $\epsilon = 10^{-9}$ and a learning rate $\alpha = 0.02$, the last parameter ensures that our system doesn't slip easily into dangerous oscillatory cycle. Obviously, the easiest way to reach a lower energy at the end of the simulation is to decrease the value of Δ; in this way we give little constraint to the function $v(t)$ but at the same time too much freedom can easily lead to bad outcomes and numerical errors.

Interestingly, for several combinations of the parameters, the function $v(t)$ has always a similar shape: its value is greater than 1 for most of the time, starting from $t = 0$, but then it rapidly falls below 1, reaching even negative values as we get closer and closer to $t = t^*$. This is evident from the Figures 4, 5, in both images, in the central part (vertically), it is shown how the Frobenius Norm changes with respect to the iterative step. Being the vertical axis in log-scale, we can see that the algorithm is able to reach a considerably low precision ϵ quite fast. In the bottom part, where several shapes of $v(t)$ are displayed (one every 20 iteration, going from red at iteration 0 to blue in the last iteration) we reach a "controlled" shape in few dozens of iterations; this means that we are able to hit a satisfying result even without choosing such a low precision ϵ. When it comes to the behavior of the energy, we see that in Figures 4, when we compare the optimized and not-optimized, we are able to get a size-able gain: $E_{GD}(t^*) - E_{opt}(t^*) \simeq 0.039$. On the other hand in the second plot, we can notice that around $t^* = 3.9$ the aforementioned change in the derivative of $v(t)$ is huge; this leads to an unwelcome result and a simulation that has to be discarded. This unfortunate feature has been noticed in the cases where the $e(t)$ is flatter (we remind that the time axis is in log-scale), which means that the algorithm can become uncontrolled for choice of big t^* or β.

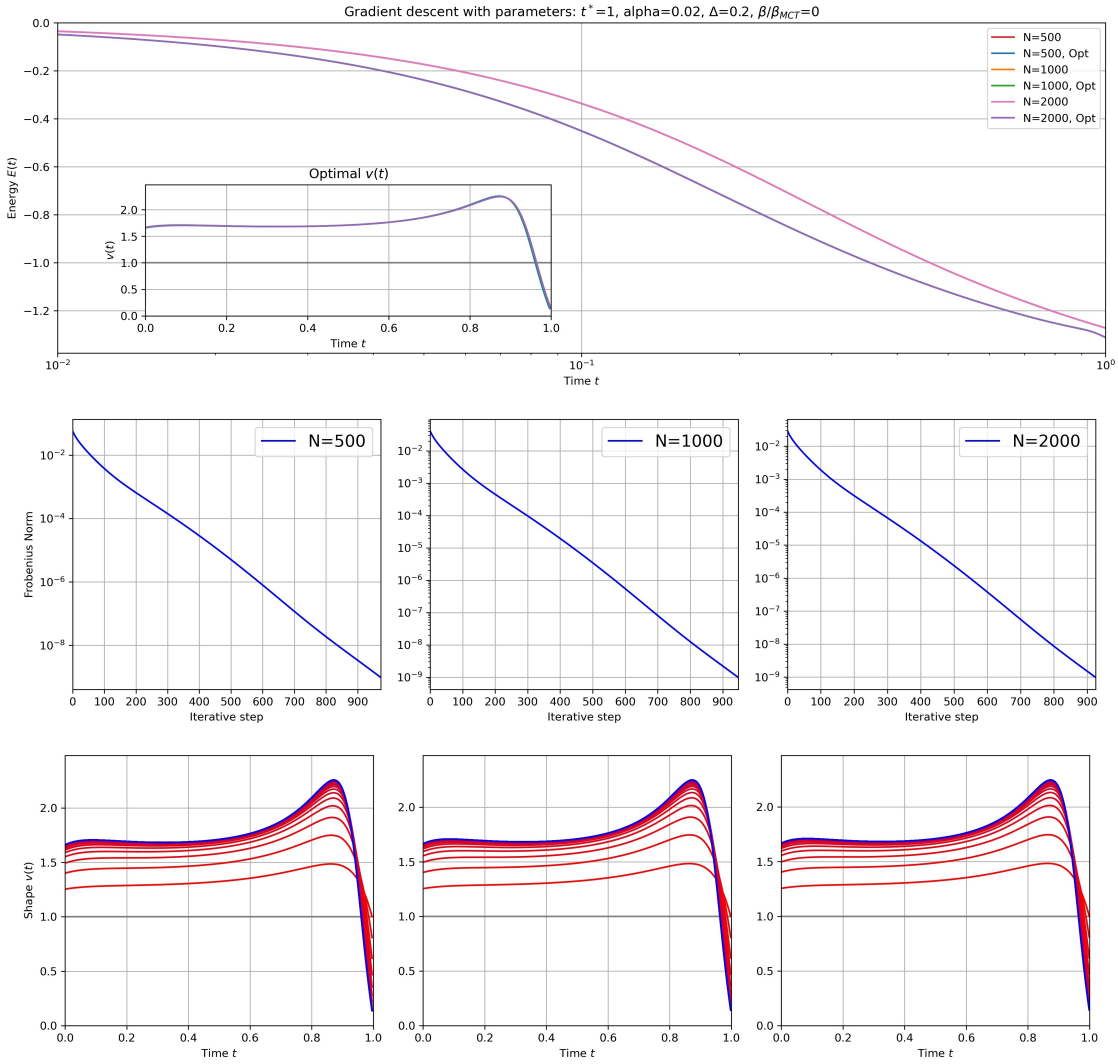

Figure 4: Simulation for: t*=1, Δ=0.2, α=0.02, dt=[0.002, 0.001, 0.0005], $\beta/\beta_{MCT} = 0$.
Top plot: evolution of the energy using the standard gradient descent vs the optimized one, we have considered 3 different values of N and thus we have 6 different curves that collapse into two.
Central plots: evolution of the Frobenius norm is shown for the different N values.
Bottom plots: evolution of the function $v(t)$ (every 20 steps), where the last step is blue and its colour 'linearly' transforms to red one has we approach the initial iteration.

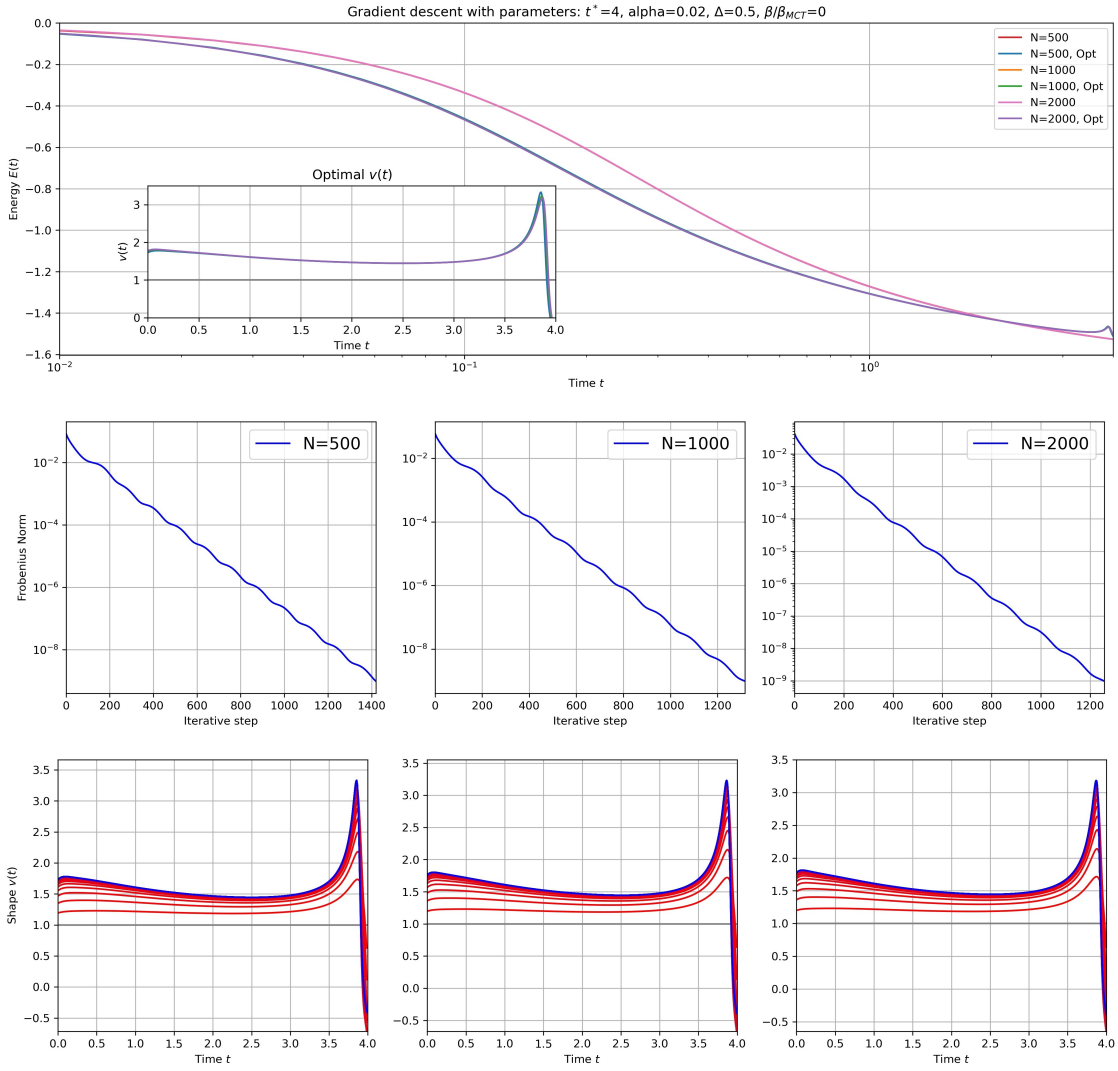

Figure 5: Simulation for: t*=4, Δ=0.5, α=0.02, dt=[0.08, 0.004, 0.002], $\beta/\beta_{MCT} = 0$.
Top plot: evolution of the energy using the standard gradient descent vs the optimized one, we have considered 3 different values of N and thus we have 6 different curves that collapse into two.
Central plots: evolution of the Frobenius norm is shown for the different N values.
Bottom plots: evolution of the function $v(t)$ (every 20 steps), where the last step is blue and its colour 'linearly' transforms to red one has we approach the initial iteration.

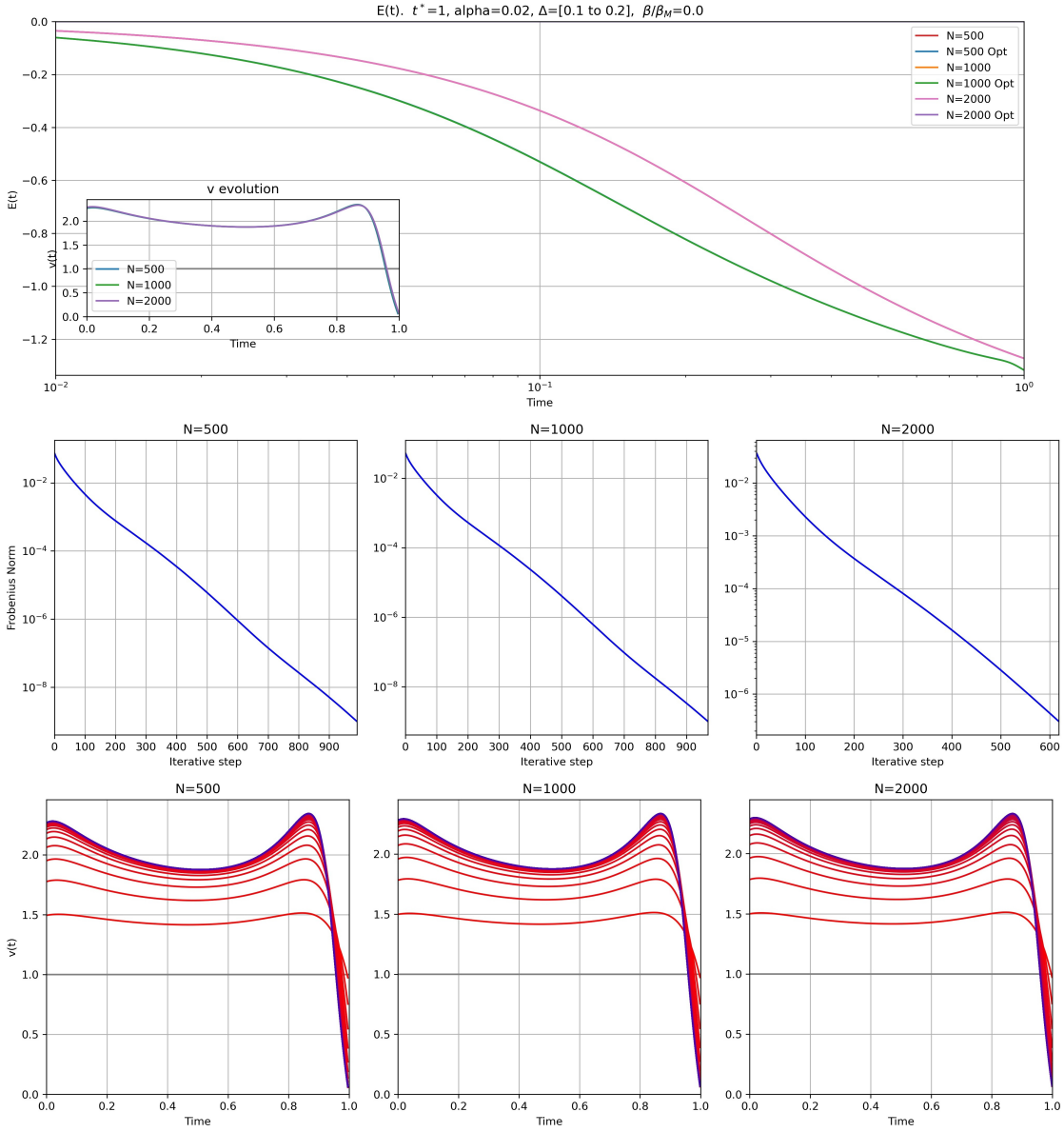

Figure 6: Simulation for: t*=1, Δ=[0.1 to 0.2], α=0.02, dt=[0.002, 0.001, 0.0005,], β/β_{MCT} = 0.

Note that here Δ increases linearly from 0.1 to 0.2 over the time [0, t*]

Top plot: evolution of the energy using the standard gradient descent vs the optimized one, we have considered 3 different values of N and thus we have 6 different curves that collapse into two.

Central plots: evolution of the Frobenius norm is shown for the different N values.

Bottom plots: evolution of the function $v(t)$ (every 20 steps), where the last step is blue and its colour 'linearly' transforms to red one has we approach the initial iteration.

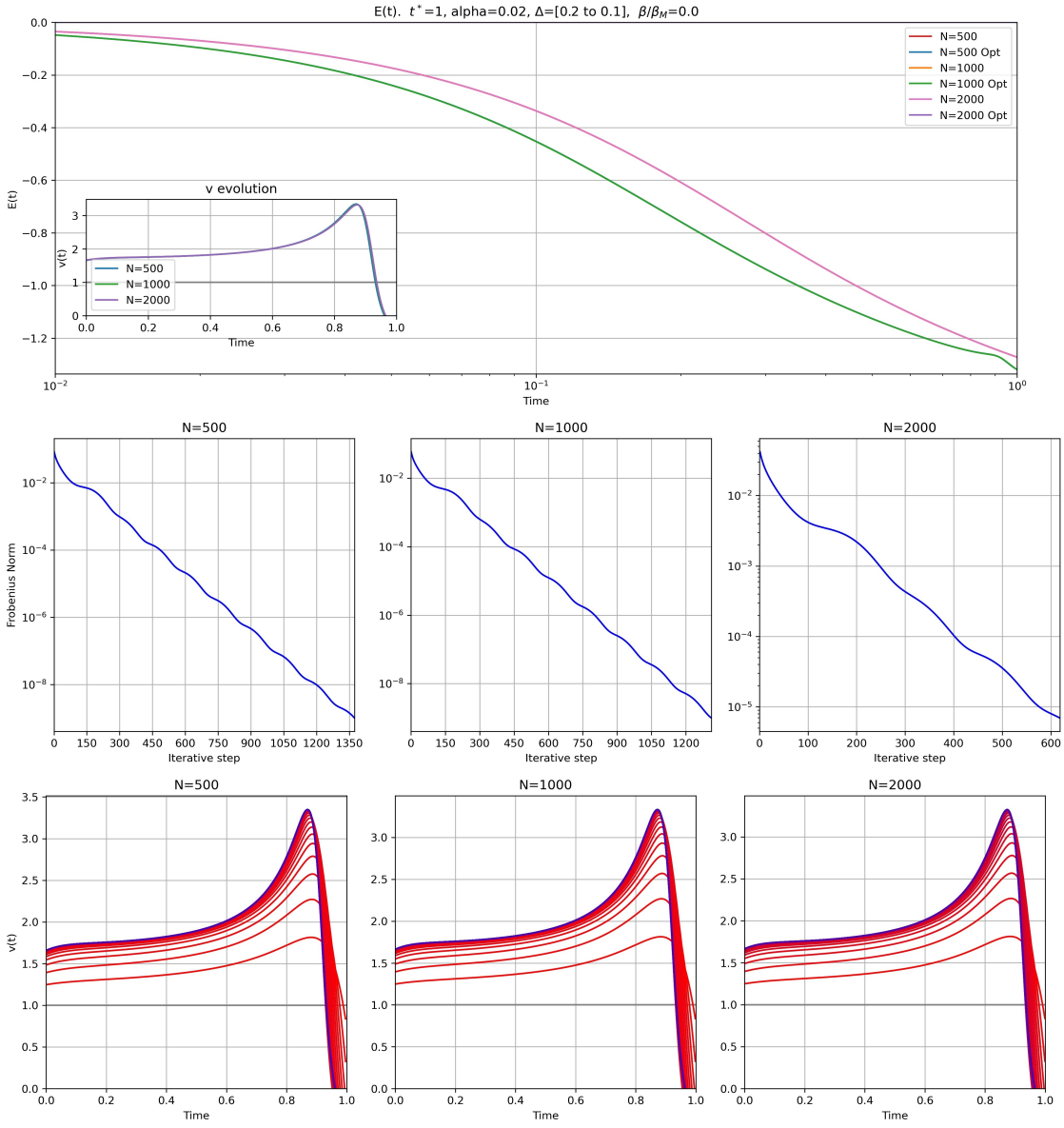

Figure 7: Simulation for: t*=1, Δ=[0.2 to 0.1], α=0.02, dt=[0.002, 0.001, 0.0005,], β/β_{MCT} = 0.

Note that here Δ decreases linearly from 0.2 to 0.1 over the time [0, t*]

Top plot: evolution of the energy using the standard gradient descent vs the optimized one, we have considered 3 different values of N and thus we have 6 different curves that collapse into two.

Central plots: evolution of the Frobenius norm is shown for the different N values.

Bottom plots: evolution of the function $v(t)$ (every 20 steps), where the last step is blue and its colour transforms "linearly" to red one has we approach the initial iteration.

As we have seen from the previous Figures 4, 5 for low values of the time we always have the same shape of the function $v(t)$. On the other hand, for value of the time close to t^*, $v(t)$ changes more sharply and abruptly. This can lead to instability in our simulation. For that reason we will constrain the freedom of the $v(t)$ function in a different way over time: for t close to 0 it will have more freedom (smaller Δ) and for t close to t* it will have less freedom (bigger Δ). We do so by increasing or decreasing Δ linearly over the time window. This last analysis is depicted in Figure 6 and 7 which has to be compared to the simplified case in Figures 4. In these new figures we tried to impose a linearly (or decreasing) Δ. Both simulation reached convergence, but the second one appears to be closer to instability. This is consistent with the fact that the shape of $v(t)$ at high times, where Δ is looser, leads to a sharper change. On the other hand, the first image is way more stable, and in the end reaches a gain of the order $E_{GD}(t^*) - E_{opt}(t^*) \simeq 0.047$, greater than the 0.039 we reached in the simplified case. For example, if we look at figure Figures 7, we notice that, compared to the plots we had for $\Delta = 0.05$ the gain is greatly improved and the numerical stability is preserved. At the same time comparing the first one to Figure 7, we notice that our assumption was correct: constraining the final part of $v(t)$ is more efficient compared to the other way around.

8 Conclusions and Perspectives

The method that we have developed on mixed p-spin glasses has allowed us to improve the performance of a simple gradient descent, although in limited situation. When the energy descent is steep, the algorithm shows to improve the GD quite a bit and we are able to reach lower energies than before. As the energy function reaches a flatter curvature, the algorithm shows some difficulties, particularly due to the shape of the function $v(t)$ that changes too rapidly in such a delicate numerical simulation. This is just a preliminary research in a context that has not been thoroughly studied and it can surely serve as a foundation for future studies. Even though we are not able to see what happens in the long time limit, we expect that further improvements on a similar fashion would allow us to cross the threshold E_{th}. In particular, there are different options that could be followed to improve such an algorithm. One would be to change the cost of adding the function $v(t)$ with respect to the initial constant value of 1; in our simulations we have used a parabolic shape centered around 1 but there are several families of functions that could be suitable for our needs. Moreover it would be interesting to look for an asymptotic solution to the problem, where the final time t^* goes to infinity.

This review has been written after completing a stage at the “Institut de Physique Theorique” funded by CEA (Commissariat à l’énergie atomique et aux énergies alternatives).

References

[1] A. Cauchy. Méthode générale pour la résolution des systèmes d'équations simultanées. *Comptes Rendus Hebd. Seances Acad. Sci. 25, 536–538*, 1847.

[2] I. Goodfellow and Y. Bengio and A. Courville. *Deep Learning*. MIT Press, 2016.

[3] J. Zhang. Gradient Descent based Optimization Algorithms for Deep Learning Models Training. 2019.

[4] S. Mannelli, G. Biroli, C. Cammarota, F. Krzakala, P. Urbani, and L. Zdeborova. Marvels and Pitfalls of the Langevin Algorithm in Noisy High-dimensional Inference. 2018.

[5] S. Mannelli, F. Krzakala, P. Urbani, and L. Zdeborova. Passed & Spurious: analysing descent algorithms and local minima in spiked matrix-tensor model. 2019.

[6] S. Mannelli and P. Urbani. Just a Momentum: Analytical Study of Momentum-Based Acceleration Methods Methods in Paradigmatic High-Dimensional Non-Convex Problem. 2021.

[7] R. Keesing and D. Stork. Evolution and Learning in Neural Networks: The Number and Distribution of Learning Trials Affect the Rate of Evolution. In R. P. Lippmann, J. Moody, and D. Touretzky, editors, *Advances in Neural Information Processing Systems*, volume 3. Morgan-Kaufmann, 1991.

[8] M. Waldrop. The chips are down for Moore's law. *Nature News*, 530:144, 2016.

[9] Giulio Biroli, Chiara Cammarota, and Federico Ricci-Tersenghi. How to iron out rough landscapes and get optimal performances: Averaged Gradient Descent and its application to tensor PCA. *Journal of Physics A: Mathematical and Theoretical*, 53, 2020.

[10] X. Yi, D. Park, Y. Chen, and C. Caramanis. Fast Algorithms for Robust PCA via Gradient Descent. 2016.

[11] Hidetoshi Nishimori. *Statistical Physics of Spin Glasses and Information Processing: an Introduction*. Oxford University Press, Oxford; New York, 2001.

[12] A. Crisanti and H.-J. Sommers. The spherical p-spin interaction spin glass model: the statics. *European Physical Journal B - EUR PHYS J B*, 87:341–354, 1992.

[13] G. Ben Arous, E. Subag, and O. Zeitouni. Geometry and Temperature Chaos in Mixed Spherical Spin Glasses at Low Temperature: The Perturbative Regime. *Communications on Pure and Applied Mathematics*, 73, 2019.

[14] A. Cavagna, I. Giardina, and G. Parisi. Stationary points of the Thouless-Anderson-Palmer free energy. *Physical Review B*, 57:1123612–11257, 1998.

[15] G. Ben Arous, A. Dembo, and A. Guionnet. Cugliandolo-Kurchan equations for dynamics of Spin-Glasses. *Probability Theory and Related Fields*, 136, 2004.

[16] G. Folena, S. Franz, and F. Ricci-Tersenghi. Gradient descent dynamics in the mixed p-spin spherical model: finite size simulation and comparison with mean-field integration. 2020.

[17] A. Barrat, S. Franz, and G. Parisi. Temperature evolution and bifurcations of metastable states in mean-field spin glasses, with connections with structural glasses. *Journal of Physics A General Physics*, 30, 1997.

[18] G. Folena, S. Franz, and F. Ricci-Tersenghi. Rethinking mean-field glassy dynamics and its relation with the energy landscape: The surprising case of the spherical mixed p -spin model. *Physical Review X*, 10, 2020.

[19] K. González-López and E. Lerner. An energy-landscape-based crossover temperature in glass-forming liquids. 2020.

[20] M Mezard, G Parisi, and M Virasoro. *Spin Glass Theory and Beyond*. world scientific, 1986.

[21] T. Castellani and A. Cavagna. Spin-glass theory for pedestrians. *J. Stat. Mech*, 2005, 2005.

A Appendix A: equations for μ, P^C, P^R and v

A.1 Discrete Lagrangian

Our optimization problem presents two different sets of equations that we have to solve, one is associated to μ, C and R which allows us to compute the energy evolution $e(t)$ of our descent, and the other one is associated to λ, P^C and P^R which lets us compute the shape of the function $v(t)$. The first set is characterized by its forward evolution, from an initial time $t = 0$ we reach a given time $t = t^* > 0$, and its integration is relatively easy as it was discussed in Section 4.4. When we consider the second one, we expect that it will follow a backward-protocol, going from $t = t^*$ to $t = 0$; this discrepancy has to be taken into consideration when integrating the equations. As a matter of facts, when we want to use a forward Euler method, we ignore the right-most point in our grid. On the other hand, if we want to use the same method in the reverse-protocol, we will ignore the left-most point in our grid, and we would not be able to conciliate the two sets of equations. For this reason, the best approach is to substitute all the integrals and derivatives in the Lagrangian (13) directly with their discrete counterpart and to perform the functional derivatives subsequently, and not the other way around. In this way we are sure that we won't have problems with the extremal points of our grids and therefore the initial conditions of our differential equations.

Everything considered the new (discretized) Lagrangian can be written as follows:

$$
\begin{aligned}
\mathcal{L} = \Delta t \underbrace{\frac{\Delta}{2}\sum_{t=0}^{t^*-1}(v_t-1)^2}_{\text{v}} \underbrace{-\Delta t\sum_{t=0}^{t^*-1}\Big[f_3'(C_{t^*,t})+v_t f_4'(C_{t^*,t})\Big]R_{t^*,t}-\beta\Big[f_3(C_{t^*,0})+f_4(C_{t^*,0})\Big]}_{\text{Energy}}+ \\
+\sum_{t=0}^{t^*-1}\sum_{t'=0}^{t}\Delta t P^C_{t,t'}\Big[\underbrace{C_{t+1,t'}-C_{t,t'}}_{\alpha}+\Delta t\underbrace{\mu_t C_{t,t'}}_{\text{A}}-\Delta t^2\underbrace{\sum_{s=0}^{t-1}M^R_{t,s}C_{t',s}}_{1}-\Delta t^2\underbrace{\sum_{s=0}^{t'-1}M^C_{t,s}R_{t',s}}_{2}-\Delta t\underbrace{\beta\hat{f}'(C_{t,0})C_{t',0}}_{3}\Big]+ \\
+\sum_{t=0}^{t^*-1}\sum_{t'=0}^{t}\Delta t P^R_{t,t'}\Big[\underbrace{R_{t+1,t'}-R_{t,t'}}_{\beta}-\delta_{t,t'}+\Delta t\underbrace{\mu_t R_{t,t'}}_{\text{B}}-\Delta t^2\underbrace{\sum_{s=t'}^{t-1}M^R_{t,s}R_{t',s}}_{4}\Big]+ \\
+\sum_{t=0}^{t^*-1}\Delta t\lambda_t\Big[\mu_t-\Delta t\sum_{s=0}^{t-1}\underbrace{(M^C_{t,s}R_{t,s}}_{5}+\underbrace{M^R_{t,s}C_{t,s})}_{6}-\underbrace{\beta\hat{f}'(C_{t,0})C_{t,0}}_{7}\Big]
\end{aligned}
\tag{14}
$$

Where the different labels will be used to allow a simple follow-up of the calculations of the single terms.

A.2 Discrete equation for μ_i

$$\frac{\delta \mathcal{L}}{\delta \mu_i} = 0 \quad \Longrightarrow \quad \lambda_i = -\Delta t \sum_{t'=0}^{i} \Big[P^C_{i,t'} C_{i,t'} + P^R_{i,t'} R_{i,t'} \Big]$$

A.3 Discrete equation for $P^C_{i,j}$

A.3.1 Boundary term

The first thing we have to do is to consider what happens at the boundaries of our system, therefore we perform the functional derivative only with respect to $\delta C_{t^*,i}$. We consider the contribution of all the different parts we have in our Lagrangian and then combine them.

α contribution:

$$\frac{\delta}{\delta C_{t^*,i}} \sum_{t=0}^{t^*-1} \sum_{t'=0}^{t} P^C_{t,t'} \Big[C_{t+1,t'} - C_{t,t'} \Big] = \sum_{t=0}^{t^*-1} \sum_{t'=0}^{t} P^C_{t,t'} \Big[\delta_{t^*,t+1} \delta_{i,t'} - \delta_{t^*,t} \delta_{i,t'} \Big] =$$
$$= \sum_{t=i}^{t^*-1} P^C_{t,i} \Big[\delta_{t^*,t+1} - \delta_{t^*,t} \Big] = P^C_{t^*-1,i}$$

Energy contribution:

$$\frac{\delta}{\delta C_{t^*,i}} \Delta t \sum_{t=0}^{t^*-1} \Big[f_3'(C_{t^*,t}) + v_t f_4'(C_{t^*,t}) \Big] R_{t^*,t} + \frac{\delta}{\delta C_{t^*,i}} \beta \Big[f_3(C_{t^*,0}) + f_4(C_{t^*,0}) \Big] =$$
$$= \Delta t \Big[f_3''(C_{t^*,i}) + v_i f_4''(C_{t^*,i}) \Big] R_{t^*,i} + \beta \Big[f_3'(C_{t^*,0}) + f_4'(C_{t^*,0}) \Big] \delta_{i,0}$$

Summing the previous terms we are able to write down the initial condition for P^C:

$$P^C_{t^*-1,i} = \Big[f_3''(C_{t^*,i}) + v_i f_4''(C_{t^*,i}) \Big] R_{t^*,i} + \beta \Big[f_3'(C_{t^*,0}) + f_4'(C_{t^*,0}) \Big] \delta_{i,0} / \Delta t$$

A.3.2 Evolution

Now that we have found the initial condition, we have to compute the integro-differential equation governing the evolution of this Lagrange multiplier.

Term A:

$$\frac{\delta}{\delta C_{i,j}} \sum_{t=0}^{t^*-1} \sum_{t'=0}^{t} P^C_{t,t'} \mu_t C_{t,t'} = P^C_{i,j} \mu_i$$

Term α:

$$\frac{\delta}{\delta C_{i,j}} \sum_{t=0}^{t^*-1} \sum_{t'=0}^{t} P^C_{t,t'} \Big[C_{t+1,t'} - C_{t,t'} \Big] = \sum_{t=0}^{t^*-1} \sum_{t'=0}^{t} P^C_{t,t'} \Big[\delta_{i,t+1}\delta_{j,t'} - \delta_{i,t}\delta_{j,t'} \Big] =$$
$$= \sum_{t=j}^{t^*-1} P^C_{t,t'} \Big[\delta_{i,t+1} - \delta_{i,t} \Big] = P^C_{i-1,j} - P^C_{i,j}$$

The term labeled as (1) has to be treated a little bit differently. We want to consider functional derivatives with respect to $\delta C_{i,j}$ with $i > j$. For that reason we have to split the summation over s into two different summation, the first one (1A) where the condition is valid and the other one (1B) where it isn't valid.. In the former we apply the symmetric property of C and exchange the order of the indices to let us handle the term.

Term 1A:

$$\frac{\delta}{\delta C_{i,j}} \sum_{t=0}^{t^*-1} \sum_{t'=0}^{t} P^C_{t,t'} \sum_{s=0}^{t'-1} M^R_{t,s} C_{t',s} = \sum_{t=0}^{t^*-1} \sum_{t'=0}^{t} P^C_{t,t'} \sum_{s=0}^{t'-1} \Big[(\partial_C M^R_{t,s}) C_{t',s} \delta_{i,t}\delta_{j,s} + M^R_{t,s}\delta_{i,t'}\delta_{j,s} \Big] =$$
$$= \sum_{t=j+1}^{t^*-1} \sum_{t'=j+1}^{t} P^C_{t,t'} \Big[(\partial_C M^R_{t,j}) C_{t',j} \delta_{i,t} + M^R_{t,j}\delta_{i,t'} \Big] == \sum_{t'=j+1}^{i} P^C_{i,t'} (\partial_C M^R_{i,j}) C_{t',j} + \sum_{t=i}^{t^*-1} P^C_{t,i} M^R_{t,j}$$

Term 1B:

$$\frac{\delta}{\delta C_{i,j}} \sum_{t=0}^{t^*-1} \sum_{t'=0}^{t} P^C_{t,t'} \sum_{s=t'}^{t-1} M^R_{t,s} C_{s,t'} =$$
$$= \sum_{t=0}^{t^*-1} \sum_{t'=0}^{t} P^C_{t,t'} \sum_{s=t'}^{t-1} \Big[(\partial_C M^R_{t,s}) C_{s,t'} \delta_{i,t}\delta_{j,s} + M^R_{t,s}\delta_{i,s}\delta_{j,t'} \Big] =$$
$$= \sum_{t'=0}^{i} P^C_{i,t'} \sum_{s=t'}^{i-1} (\partial_C M^R_{i,s}) C_{s,t'} \delta_{j,s} + \sum_{t=i+1}^{t^*-1} \sum_{t'=0}^{t} P^C_{t,t'} M^R_{t,i} \delta_{j,t'} =$$
$$= \sum_{t'=0}^{j} P^C_{i,t'} (\partial_C M^R_{i,j}) C_{j,t'} + \sum_{t=i+1}^{t^*-1} P^C_{t,j} M^R_{t,i}$$

Term 2:

$$\frac{\delta}{\delta C_{i,j}}\sum_{t=0}^{t^*-1}\sum_{t'=0}^{t}P^C_{t,t'}\sum_{s=0}^{t'-1}M^C_{t,s}R_{t',s}=\sum_{t=0}^{t^*-1}\sum_{t'=0}^{t}P^C_{t,t'}\sum_{s=0}^{t'-1}(\partial_C M^C_{t,s})R_{t',s}\delta_{i,t}\delta_{j,s}=$$
$$=\sum_{t'=0}^{i}P^C_{i,t'}\sum_{s=0}^{t'-1}(\partial_C M^C_{i,s})R_{t',s}\delta_{j,s}=\sum_{t'=j+1}^{i}P^C_{i,t'}(\partial_C M^C_{i,j})R_{t',j}$$

Term 3:

$$\frac{\delta}{\delta C_{i,j}}\beta\sum_{t=0}^{t^*-1}\sum_{t'=0}^{t}P^C_{t,t'}\hat{f}'(C_{t,0})C_{t',0}=$$
$$=\beta\delta_{j,0}\Big[\sum_{t=0}^{t^*-1}\sum_{t'=0}^{t}P^C_{t,t'}\hat{f}''(C_{t,0})\delta_{i,t}C_{t',0}+\sum_{t=0}^{t^*-1}\sum_{t'=0}^{t}P^C_{t,t'}\hat{f}'(C_{t,0})\delta_{i,t'}\Big]=$$
$$=\beta\delta_{j,0}\Big[\sum_{t'=0}^{i}P^C_{i,t'}\hat{f}''(C_{i,0})C_{t',0}+\sum_{t=i}^{t^*-1}P^C_{t,i}\hat{f}'(C_{t,0})\Big]$$

Term 4

$$\frac{\delta}{\delta C_{i,j}}\sum_{t=0}^{t^*-1}\sum_{t'=0}^{t}P^R_{t,t'}\sum_{s=t'}^{t-1}M^R_{t,s}R_{t',s}=\sum_{t=0}^{t^*-1}\sum_{t'=0}^{t}P^R_{t,t'}\sum_{s=t'}^{t-1}(\partial_C M^R_{t,s})R_{s,t'}\delta_{i,t}\delta_{j,s}=$$
$$=\sum_{t'=0}^{i}P^R_{i,t'}\sum_{s=t'}^{i-1}(\partial_C M^R_{i,s})R_{s,t'}\delta_{j,s}=\sum_{t'=0}^{j}P^R_{i,t'}(\partial_C M^R_{i,j})R_{j,t'}$$

Term 5-6:

$$\frac{\delta}{\delta C_{i,j}}\sum_{t=0}^{t^*-1}\lambda_t\sum_{s=0}^{t-1}(M^C_{t,s}R_{t,s}+M^R_{t,s}C_{t,s})=$$
$$=\sum_{t=0}^{t^*-1}\lambda_t\sum_{s=0}^{t-1}\Big[(\partial_C M^C_{t,s})R_{t,s}+(\partial_C M^R_{t,s})C_{t,s}+M^R_{t,s}\Big]\delta_{i,t}\delta_{j,s}=$$
$$=\lambda_i\Big[(\partial_C M^C_{i,j})R_{i,j}+(\partial_C M^R_{i,j})C_{i,j}+M^R_{i,j}\Big]$$

Term 7:

$$\frac{\delta}{\delta C_{i,j}}\beta\sum_{t=0}^{t^*-1}\lambda_t\hat{f}'(C_{t,0})C_{t,0}=\beta\lambda_i\delta_{j,0}\Big[\hat{f}''(C_{i,0})C_{i,0}+\hat{f}'(C_{i,0})\Big]$$

A.4 Discrete equation for $P^R_{i,j}$

For $P^R_{i,j}$, the same considerations as $P^C_{i,j}$ apply.

A.4.1 Boundary Term

β contribution:

$$\frac{\delta}{\delta R_{t^*,i}} \sum_{t=0}^{t^*-1} \sum_{t'=0}^{t} P^R_{t,t'} \Big[R_{t+1,t'} - R_{t,t'} \Big] = \sum_{t=0}^{t^*-1} \sum_{t'=0}^{t} P^R_{t,t'} \Big[\delta_{t^*,t+1}\delta_{i,t'} - \delta_{t^*,t}\delta_{i,t'} \Big] =$$
$$= \sum_{t=i}^{t^*-1} P^R_{t,i} \Big[\delta_{t^*,t+1} - \delta_{t^*,t} \Big] = P^R_{t^*-1,i}$$

Energy contribution:

$$\frac{\delta}{\delta R_{t^*,i}} \sum_{t=0}^{t^*-1} \Big[f_3'(C_{t^*,t}) + v_t f_4'(C_{t^*,t}) \Big] R_{t^*,t} = \Big[f_3'(C_{t^*,i}) + v_i f_4'(C_{t^*,i}) \Big]$$

Summing the previous two equations we are able to write down the initial condition for P^R:

$$P^R_{t^*-1,i} = \Big[f_3'(C_{t^*,i}) + v_i f_4'(C_{t^*,i}) \Big]$$

A.4.2 Evolution

Term B:

$$\frac{\delta}{\delta R_{i,j}} \sum_{t=0}^{t^*-1} \sum_{t'=0}^{t} P^R_{t,t'} \mu_t R_{t,t'} = P^R_{i,j} \mu_i$$

Term β:

$$\frac{\delta}{\delta R_{i,j}} \sum_{t=0}^{t^*-1} \sum_{t'=0}^{t} P^R_{t,t'} \Big[R_{t+1,t'} - R_{t,t'} \Big] = \sum_{t=0}^{t^*-1} \sum_{t'=0}^{t} P^R_{t,t'} \Big[\delta_{i,t+1}\delta_{j,t'} - \delta_{i,t}\delta_{j,t'} \Big] =$$
$$= \sum_{t=j}^{t^*-1} P^R_{t,j} \Big[\delta_{i,t+1} - \delta_{i,t} \Big] = P^R_{i-1,j} - P^R_{i,j}$$

Term 1A:

$$\frac{\delta}{\delta R_{i,j}} \sum_{t=0}^{t^*-1} \sum_{t'=0}^{t} P^C_{t,t'} \sum_{s=0}^{t'-1} M^R_{t,s} C_{t',s} = \sum_{t=0}^{t^*-1} \sum_{t'=0}^{t} P^C_{t,t'} \sum_{s=0}^{t'-1} \Big[(\partial_R M^R_{i,j}) C_{t',s} \delta_{i,t} \delta_{j,s} \Big] =$$
$$= \sum_{t'=0}^{i} P^C_{i,t'} \sum_{s=0}^{t'-1} \Big[(\partial_R M^R_{i,s}) C_{t',s} \delta_{j,s} \Big] = \sum_{t'=j+1}^{i} P^C_{i,t'} (\partial_R M^R_{i,j}) C_{t',j}$$

Term 1B:

$$\frac{\delta}{\delta R_{i,j}} \sum_{t=0}^{t^*-1} \sum_{t'=0}^{t} P^C_{t,t'} \sum_{s=t'}^{t^*-1} M^R_{t,s} C_{s,t'} = \sum_{t=0}^{t^*-1} \sum_{t'=0}^{t} P^C_{t,t'} \sum_{s=t'}^{t^*-1} \Big[(\partial_R M^R_{i,j}) C_{s,t'} \delta_{i,t} \delta_{j,s} \Big] =$$
$$= \sum_{t'=0}^{i} P^C_{i,t'} \sum_{s=t'}^{t^*-1} \Big[(\partial_R M^R_{i,s}) C_{s,t'} \delta_{j,s} \Big] = \sum_{t'=0}^{j} P^C_{i,t'} (\partial_R M^R_{i,j}) C_{j,t'}$$

Term 2:

$$\frac{\delta}{\delta R_{i,j}} \sum_{t=0}^{t^*-1} \sum_{t'=0}^{t} P^C_{t,t'} \sum_{s=0}^{t'-1} M^C_{t,s} R_{t',s} = \sum_{t=0}^{t^*-1} \sum_{t'=0}^{t} P^C_{t,t'} \sum_{s=0}^{t'-1} M^C_{t,s} \delta_{i,t'} \delta_{j,s} =$$
$$= \sum_{t=i}^{t^*-1} P^C_{t,i} \sum_{s=0}^{i-1} M^C_{t,s} \delta_{j,s} = \sum_{t=i}^{t^*-1} P^C_{t,i} M^C_{t,j}$$

Term 4:

$$\frac{\delta}{\delta R_{i,j}} \sum_{t=0}^{t^*-1} \sum_{t'=0}^{t} P^R_{t,t'} \sum_{s=t'}^{t-1} M^R_{t,s} R_{s,t'} =$$
$$= \sum_{t=0}^{t^*-1} \sum_{t'=0}^{t} P^R_{t,t'} \sum_{s=t'}^{t-1} \Big[(\partial_R M^R_{t,s}) R_{s,t'} \delta_{i,t} \delta_{j,s} + M^R_{t,s} \delta_{i,s} \delta_{j,t'} \Big] =$$
$$= \sum_{t'=0}^{i} P^R_{i,t'} \sum_{s=t'}^{i-1} (\partial_R M^R_{i,s}) R_{s,t'} \delta_{j,s} + \sum_{t=j}^{t^*-1} P^R_{t,j} \sum_{s=j}^{t-1} M^R_{t,s} \delta_{i,s} =$$
$$= \sum_{t'=0}^{j} P^R_{i,t'} (\partial_R M^R_{i,j}) R_{j,t'} + \sum_{t=i+1}^{t^*-1} P^R_{t,j} M^R_{t,i}$$

Term 5-6:

$$\frac{\delta}{\delta R_{i,j}} \sum_{t=0}^{t^*-1} \lambda_t \Big[\sum_{s=0}^{t-1} (M^C_{t,s} R_{t,s} + M^R_{t,s} C_{t,s}) \Big] = \sum_{t=0}^{t^*-1} \lambda_t \sum_{s=0}^{t-1} \Big[M^C_{ts} + (\partial_R M^R_{t,s}) C_{t,s} \Big] \delta_{i,t} \delta_{j,s} =$$
$$= \lambda_i \Big[M^C_{i,j} + (\partial_R M^R_{i,j}) C_{i,j} \Big]$$

A.5 Discrete equation for v_i

Term v:

$$\frac{\delta}{\delta v_i} \frac{\Delta}{2} \sum_{t=0}^{t^*-1} (v_t - 1)^2 = \Delta (v_i - 1)$$

Term Energy:

$$\frac{\delta}{\delta v_i} \sum_{t=0}^{t^*-1} (f_3'(C_{t^*,t}) + v_t f_4'(C_{t^*,t})) R_{t^*,t} = \sum_{t=0}^{t^*-1} f_4'(C_{t^*,t}) R_{t^*,t} \delta_{i,t} = f_4'(C_{t^*,i}) R_{t^*,i}$$

Term 1A:

$$\frac{\delta}{\delta v_i} \sum_{t=0}^{t^*-1} \sum_{t'=0}^{t} P^C_{t,t'} \sum_{s=0}^{t'-1} M^R_{t,s} C_{t',s} =$$
$$= \frac{\delta}{\delta v_i} \sum_{t=0}^{t^*-1} \sum_{t'=0}^{t} P^C_{t,t'} \sum_{s=0}^{t'-1} \Big[f_4''(C_{t,s}) v_t v_s R_{t,s} \Big] C_{t',s} =$$
$$= \sum_{t=0}^{t^*-1} \sum_{t'=0}^{t} P^C_{t,t'} \sum_{s=0}^{t'-1} f_4''(C_{t,s}) R_{t,s} C_{t',s} \Big[v_t \delta_{i,s} + v_s \delta_{i,t} \Big] =$$
$$= \sum_{t=i+1}^{t^*-1} \sum_{t'=i+1}^{t} P^C_{t,t'} f_4''(C_{t,i}) R_{t,i} C_{t',i} v_t + \sum_{t'=0}^{i} \sum_{s=0}^{t'-1} P^C_{i,t'} f_4''(C_{i,s}) R_{i,s} C_{t',s} v_s$$

Term 1B:

$$\frac{\delta}{\delta v_i}\sum_{t=0}^{t^*-1}\sum_{t'=0}^{t}P^C_{t,t'}\sum_{s=t'}^{t-1}M^R_{t,s}C_{s,t'} =$$
$$= \frac{\delta}{\delta v_i}\sum_{t=0}^{t^*-1}\sum_{t'=0}^{t}P^C_{t,t'}\sum_{s=t'}^{t-1}\Big[f_4''(C_{t,s})v_tv_sR_{t,s}\Big]C_{s,t'} =$$
$$= \sum_{t=0}^{t^*-1}\sum_{t'=0}^{t}P^C_{t,t'}\sum_{s=t'}^{t-1}f_4''(C_{t,s})R_{t,s}C_{s,t'}\Big[v_t\delta_{i,s}+v_s\delta_{i,t}\Big]$$
$$= \sum_{t=i+1}^{t^*-1}\sum_{t'=0}^{i}P^C_{t,t'}f_4''(C_{t,i})R_{t,i}C_{i,t'}v_t + \sum_{t'=0}^{i}\sum_{s=t'}^{i-1}P^C_{i,t'}f_4''(C_{i,s})R_{i,s}C_{s,t'}v_s$$

Term 2:

$$\frac{\delta}{\delta v_i}\sum_{t=0}^{t^*-1}\sum_{t'=0}^{t}P^C_{t,t'}\sum_{s=0}^{t'-1}\Big[f_4'(C_{t,s})v_tv_s\Big]R_{t',s} =$$
$$= \frac{\delta}{\delta v_i}\sum_{t=0}^{t^*-1}\sum_{t'=0}^{t}P^C_{t,t'}\sum_{s=0}^{t'-1}\Big[f_4''(C_{t,s})v_tv_sR_{t,s}\Big]C_{s,t'} =$$
$$= \sum_{t=0}^{t^*-1}\sum_{t'=0}^{t}P^C_{t,t'}\sum_{s=0}^{t'-1}f_4'(C_{t,s})R_{t',s}\Big[v_t\delta_{i,s}+v_s\delta_{i,t}\Big] =$$
$$= \sum_{t=i+1}^{t^*-1}\sum_{t'=i+1}^{t}P^C_{t,t'}f_4'(C_{t,i})R_{t',i}v_t + \sum_{t'=0}^{i}\sum_{s=0}^{t'-1}P^C_{i,t'}f_4'(C_{i,s})R_{t',s}v_s$$

Term 3:

$$\frac{\delta}{\delta v_i}\beta\sum_{t=0}^{t^*-1}\sum_{t'=0}^{t}P^C_{t,t'}\hat{f}'(C_{t,0})C_{t',0} = \beta\sum_{t=0}^{t^*-1}\sum_{t'=0}^{t}P^C_{t,t'}f_4'(C_{t,0})C_{t',0}\delta_{i,t} = \beta\sum_{t'=0}^{i}P^C_{i,t'}f_4'(C_{i,0})C_{t',0}$$

Term 4:

$$\frac{\delta}{\delta v_i}\sum_{t=0}^{t^*}\sum_{t'=0}^{t}P^R_{t,t'}\sum_{s=t'}^{t-1}M^R_{t,s}R_{s,t'} =$$
$$=\frac{\delta}{\delta v_i}\sum_{t=0}^{t^*}\sum_{t'=0}^{t}P^R_{t,t'}\sum_{s=t'}^{t-1}\Big[f_4''(C_{t,s})v_tv_sR_{t,s}\Big]R_{s,t'} =$$
$$=\sum_{t=0}^{t^*}\sum_{t'=0}^{t}P^R_{t,t'}\sum_{s=t'}^{t-1}f_4''(C_{t,s})R_{t,s}R_{s,t'}\Big[v_t\delta_{i,s}+v_s\delta_{i,t}\Big] =$$
$$=\sum_{t=i+1}^{t^*-1}\sum_{t'=0}^{i}P^R_{t,t'}f_4''(C_{t,i})R_{t,i}R_{i,t'}v_t+\sum_{t'=0}^{i}\sum_{s=t'}^{i-1}P^R_{i,t'}f_4''(C_{i,s})R_{i,s}R_{s,t'}v_s$$

Term 5:

$$\frac{\delta}{\delta v_i}\sum_{t=0}^{t^*-1}\lambda_t\sum_{s=0}^{t-1}M^C_{t,s}R_{t,s}=\frac{\delta}{\delta v_i}\sum_{t=0}^{t^*-1}\lambda_t\sum_{s=0}^{t-1}f_4'(C_{t,s})v_tv_sR_{t,s} =$$
$$=\sum_{t=i+1}^{t^*-1}\lambda_tf_4'(C_{t,i})R_{t,i}v_t+\lambda_i\sum_{s=0}^{i-1}f_4'(C_{i,s})R_{i,s}v_s$$

Term 6:

$$\frac{\delta}{\delta v_i}\sum_{t=0}^{t^*-1}\lambda_t\sum_{s=0}^{t-1}M^R_{t,s}C_{t,s}=\frac{\delta}{\delta v_i}\sum_{t=0}^{t^*-1}\lambda_t\sum_{s=0}^{t-1}f_4''(C_{t,s})v_tv_sR_{t,s}C_{t,s} =$$
$$=\sum_{t=0}^{t^*-1}\lambda_t\sum_{s=0}^{t-1}f_4''(C_{t,s})R_{t,s}C_{t,s}\Big[v_t\delta_{i,s}+v_s\delta_{i,t}\Big] =$$
$$=\sum_{t=i+1}^{t^*-1}\lambda_tf_4''(C_{t,i})R_{t,i}C_{t,i}v_t+\lambda_i\sum_{s=0}^{i-1}f_4''(C_{i,s})R_{i,s}C_{i,s}v_s$$

Term 7:

$$\frac{\delta}{\delta v_i}\beta\sum_{t=0}^{t^*-1}\lambda_t\hat{f}'(C_{t,0})C_{t,0}=\beta\sum_{t=0}^{t^*-1}\lambda_tf_4'(C_{t,0})C_{t,0}\delta_{i,t}=\beta\lambda_if_4'(C_{i,0})C_{i,0}$$