MONTE CARLO METHODS TO SOLVE SPARSE LINEAR SYSTEMS
Report 3

Massimiliano Lupo Pasini

Introduction

In this report we tackle the issue of different adaptive approaches that might be used in a Monte Carlo linear solver. The development of this topic has been raised by what shown in the previous reports. In fact the particular kind of problems of interest affects the minimal number of random walks to be used in order to compute an accurate solution. Moreover the number of steps considered for each random walk affects the accuracy of the computed solution as well.

This issue assumes a considerable role when we deal with an hybrid approach such as the Sequential Monte Carlo rather than the Monte Carlo Synthetic Acceleration. In fact the use of an adaptive algorithm may employ a varying number of random walks for each numerical iteration.

By following the introductory considerations made in this paragraph we aim at developing different criteria to select proper numerical and statistical settings.

In particular we focus at first in showing what already treated in other paperworks such as [Sla13]. Therefore we consider an adaptive procedure to detect a reasonable point where to cut each random walk without neglecting significant contributions.

Afterwards we take into account some methods that might be used to select the number of random walks to employ. We do this in order to maintain the uncertainty below a certain threshold.

In conclusion we combine all the previous approaches in order to exploit the efficiency of all of them in a greedy viewpoint.

All the model problems collected in the `GitHub` repository are used to test the efficiency of the algorithms introduces. They are "JPWH_991", "FS_680_1", "Marshak Wave", "1d shifted Laplacian" and "2d Laplacian" and "ifiss advection diffusion".

Chapter 1

An adaptive cut-off for the length of random walks

At first we focus on the definition of a weight cut-off in order to decide where to terminate the histories. It implies that we are looking for a quantity that tell us how many terms of the Neumann series need to be considered.

As already described in Section 1.1 of Report 1, the Forward method is characterized by the introduction of random variables such that

$$w_{i,j} = \frac{H_{i,j}}{P_{i,j}}.$$

This enables us to introduce new quantities

$$W_0 = \frac{h_{k_0}}{\tilde{p}_{k_0}}, \quad W_j = W_{j-1} w_{i,j}, \quad j = 1, \cdots, i.$$

representing the contribution given by the history for the estimation of the solution at each step. When W_i gets too small, then there is no point in keeping the random walk moving and we can stop it.

A very similar way of reasoning holds for the Adjoint method. The only difference is the way $w_{i,j}$ are defined:

$$w_{i,j} = \frac{H_{j,i}}{P_{i,j}}.$$

The formula for the estimator of the Forward Monte Carlo for a generic entry of the solution vector is

$$\theta_i = E\Big[\sum_{l=0}^{\infty} W_l b_{k_l}\Big] = x_i = \sum_{l=0}^{\infty} \sum_{k_1=1}^{n} \sum_{k_2=1}^{n} \cdots \sum_{k_l=1}^{n} P_{k_0,k_1} P_{k_1,k_2} \cdots P_{k_{l-1},k_l} w_{k_0,k_1} w_{k_1,k_2} \cdots w_{k_{l-1},k_l} b_{k_l}. \tag{1.1}$$

For the Adjoint method it is

$$\theta_j = E\Big[\sum_{l=0}^{\infty} W_l \delta_{k_l,j}\Big] = \sum_{l=0}^{\infty} \sum_{k_1}^{n} \sum_{k_2}^{n} \cdots \sum_{k_l}^{n} b_{k_0} P_{k_0,k_1} P_{k_1,k_2} \cdots P_{k_{l-1},K_l} w_{k_0,k_1} \cdots w_{k_{l-1},k_l} \delta_{k_l,j}. \tag{1.2}$$

The goal is to find an automatic way for the cutoff of the histories. Therefore we are looking for an m such that

$$\tilde{\theta}_i = E\Big[\sum_{l=0}^{m} W_l b_{k_l}\Big] = x_i = \sum_{l=0}^{m} \sum_{k_1=1}^{n} \sum_{k_2=1}^{n} \cdots \sum_{k_l=1}^{n} P_{k_0,k_1} P_{k_1,k_2} \cdots P_{k_{l-1},k_l} w_{k_0,k_1} w_{k_1,k_2} \cdots w_{k_{l-1},k_l} b_{k_l}.$$

and

$$\tilde{\theta}_j = E\Big[\sum_{l=0}^{m} W_l \delta_{k_l,j}\Big] = \sum_{l=0}^{m} \sum_{k_1}^{n} \sum_{k_2}^{n} \cdots \sum_{k_l}^{n} b_{k_0} P_{k_0,k_1} P_{k_1,k_2} \cdots P_{k_{l-1},K_l} w_{k_0,k_1} \cdots w_{k_{l-1},k_l} \delta_{k_l,j}.$$

are good approximations of 1.1 and 1.2 respectively.

In [Sla13] there is a criterion for the cutoff of the random walk that works either for the Forward or for the Adjoint method. It consists in defining a cutoff relative threshold W_c and looking for a step such that

$$W_f \leq W_c W_0. \tag{1.3}$$

W_0 is the value of the weight at the initial step of the random walk and W_f is the value of the weight after f steps.

When the inequality 1.3 holds, then it means that all the steps of the random walk that come after the f-th one are negligible, since the tally contributions become increasingly small. It is equivalent to say that all the powers of the iteration matrix that are bigger than f can be ignored in the Neumann series.

Of course the point where the random walks are cut does not have to be selected too early in the sequence of steps, otherwise it will compromise the accuracy of the computed solution.

In order to understand the efficiency of this criterion and what might be a reasonable choice for W_c, we tuned the weight cutoff to compute the solution to a set of problems.

1.1 Numerical results

In this section we consider the Jacobi preconditioning in order to rewrite the linear system $A\mathbf{x} = \mathbf{b}$ into a fixed point scheme.

The resulting iteration matrix H is such that

$$H = I - D^{-1}A, \quad D = diag(A).$$

1.1.1 Forward Monte Carlo

For now we focus on the Forward Method with an almost optimal probability to compute the solution to $A\mathbf{x} = \mathbf{b}$ with the standard Monte Carlo algorithm. Different values for the weight cutoff have been used.

In Figure 1.2 we have results for the the Marshak Wave problem (also called as Thermal Equation in this report). The matrix associated with this problem is such that $A \in \mathbb{R}^{1600\times1600}$. The spectral radius of H is $\rho(H) = 0.6$. Therefore we expect to need a few terms of the Neumann series in order to compute an accurate solution to the linear system. By setting the cutoff threshold to 1 we have a departure of the relative error from the Central Limit Theorem behavior. This happens because the threshold requires the random walks to be cut too early. Therefore the approximated Neumann series neglects powers of the iteration matrix that should be taken into account. It can be noticed that already for $W_c = 10^{-4}$ we succeed in reproducing pretty well the CLT nehavior. However, as the number of employed ranomd walks increases, the descent gets lower. This is due to the fact that, in order to get a moer precise solution, we cannot just increase the number of random walks. We have to increase the number of steps per history as well. This is a phenomeno that will be stressed out also by the error behaviour associated with other test cases.

Let us consider now the "JPWH_991" case. It is sits in $\mathbb{R}^{991\times991}$. Its iteration matrix has a spectral radius $\rho(H) = 0.979$. Therefore we expect to require a more demanding threshold in order to reproduce a behaviour similar to the one for the Marshak problem. By doing so, the behavior of this matrix is coherent with the expected trend. In fact for $W_c = 10^{-3}$, it can be noticed tht the Central Limit Theorem is respected.

More illustrative in terms of the usefulness of the tuning of W_c is the shifted Laplacian. This a 50 by 50 matrix and the spectral radius of the iteration matrix is $\rho(H) = 0.499$. For different values of the threshold, it is noticed a progressive improvement in terms of error descent, as shown in Figure 1.3.

In conclusion we present a very difficult test case, the "FS_680_1". The iteration matrix related to it has a spectral radius equal to 0.97. As it will be stressed out also later on in the Report, however, the iteration matrix has a particular sparsity pattern (see Figure 1.7). A lot of rows are full of zeros, causing the sudden death of many random walks in the Forward method. Indeed, even for a weight cut-off equal to 10^{-12}, the slope of the error curve is far from reproducing the expected $\frac{1}{\sqrt{N}}$.

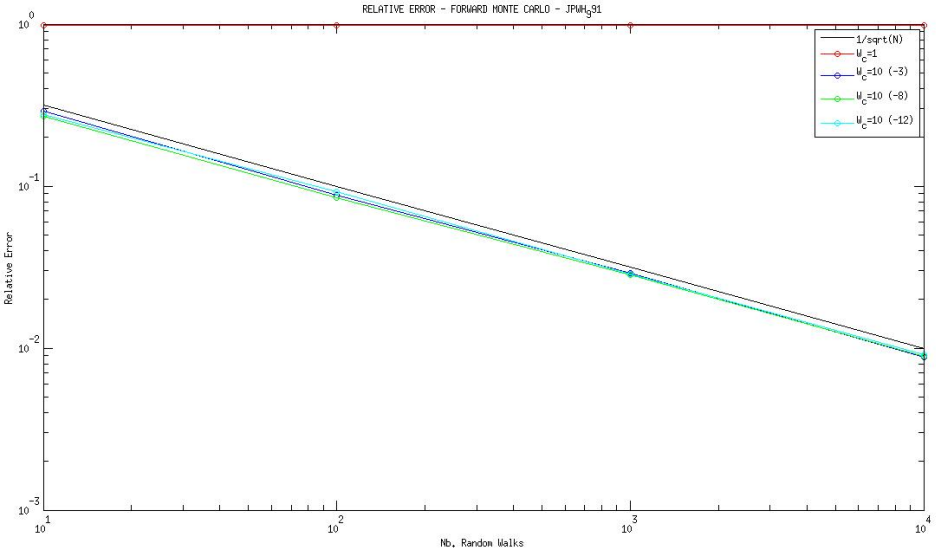

Figure 1.1: JPWH_991 - Relative error with a weight cutoff $W_c = 1$ (red), $W_c = 10^{-3}$ (green), $W_c = 10^{-8}$ (blue) and $W_c = 10^{-12}$ (cyan).

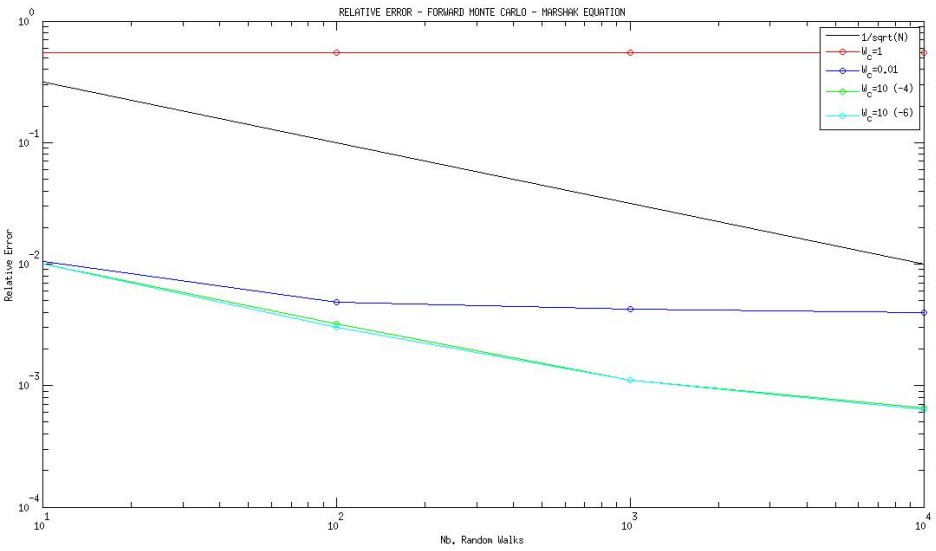

Figure 1.2: Thermal Equation - Relative error with a weight cutoff $W_c = 1$ (red), $W_c = 0.01$ (green), $W_c = 10^{-4}$ (blue) and $W_c = 10^{-6}$ (cyan).

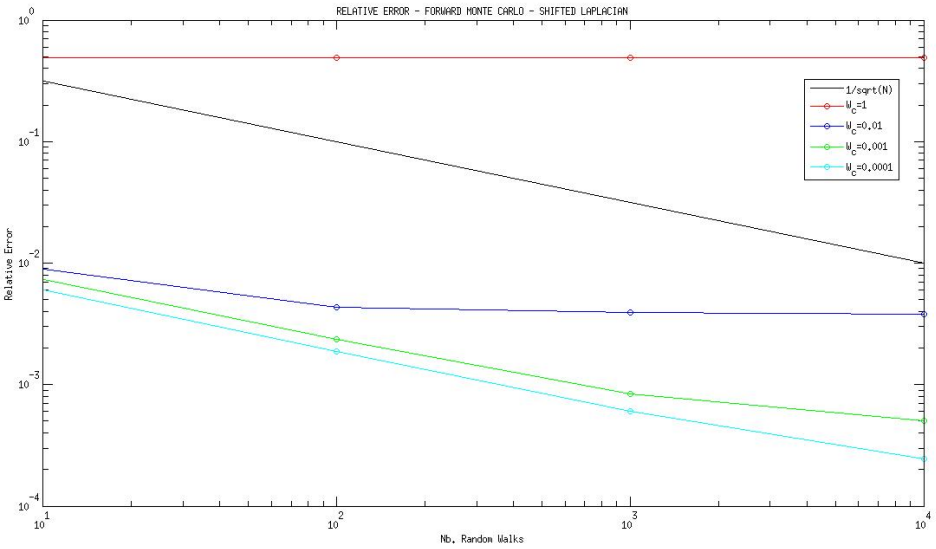

Figure 1.3: Forward Method. Shifted 1D Laplacian - Relative error with a weight cutoff $W_c = 1$ (red), $W_c = 0.01$ (green), $W_c = 0.001$ (blue) and $W_c = 0.0001$ (cyan).

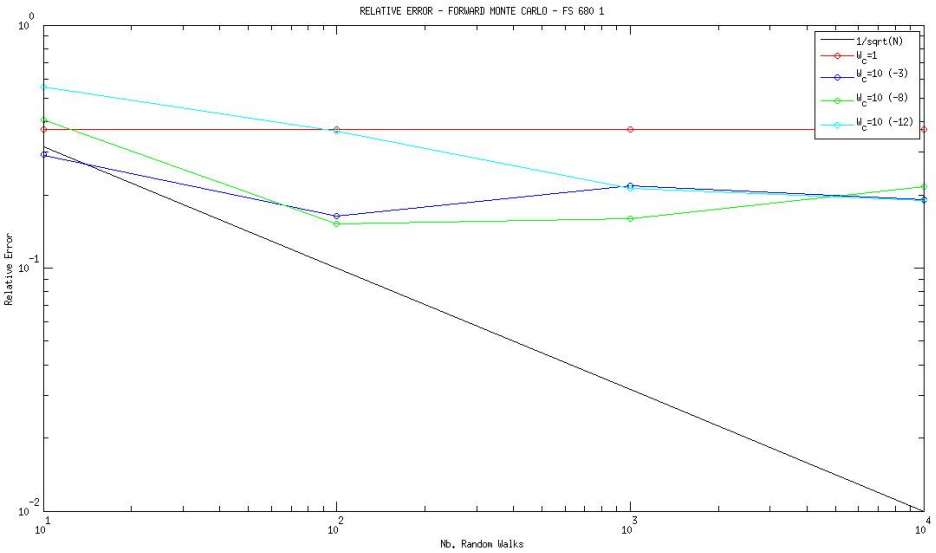

Figure 1.4: FS_680_1 - Relative error with a weight cutoff $W_c = 1$ (red), $W_c = 10^{-3}$ (green), $W_c = 10^{-8}$ (blue) and $W_c = 10^{-12}$ (cyan).

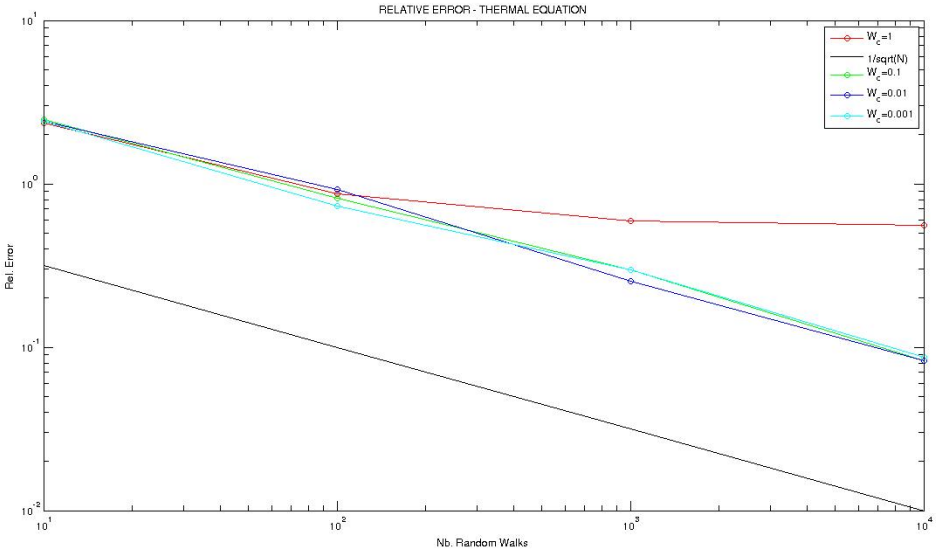

Figure 1.5: Thermal Equation - Relative error with a weight cutoff $W_c = 1$ (red), $W_c = 0.1$ (green), $W_c = 0.01$ (blue) and $W_c = 0.001$ (cyan).

1.1.2 Adjoint Monte Carlo

Now we focus on the Adjoint Method. As concerns the thermal equation, in Figure 1.5 the relative error with respect to the varying number of employed random walks is shown. $W_c = 0.1$ instead restores the CLT behavior, since a larger number of steps for each random walk is preserved by the cutoff.

We now consider the result of the Adjoint method applied on the problem "FS_680_1". By looking at Figure 1.6 it is discovered that the change of cutoff threshold does not affect much the change of slope of the relative error curve. It may be due to the particular pattern of the iteration matrix, characterized by half of the columns full of zeros (Figure 1.7).

A very interesting result is provided by the shifted 1D Laplacian, in terms of cutoff effectiveness and very similar to the Thermal Equation as well. For $W_c = 1$ it is found that the error behavior departs from what expected by the CLT. However even just for $W_c = 0.1$ the situation gets better and the $\frac{1}{\sqrt{N}}$ trend for the error is restored.

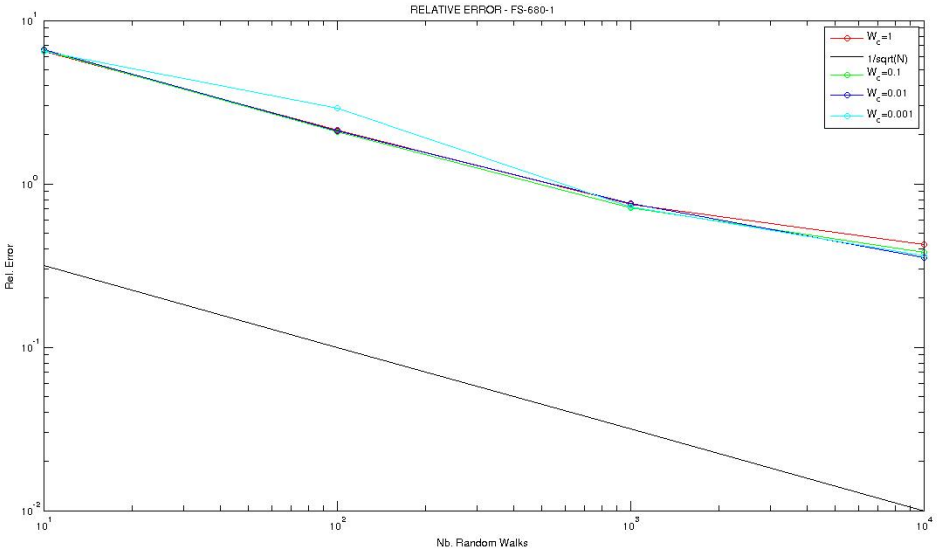

Figure 1.6: FS_680_1 - Relative error with a weight cutoff $W_c = 1$ (red), $W_c = 0.1$ (green), $W_c = 0.01$ (blue) and $W_c = 0.001$ (cyan).

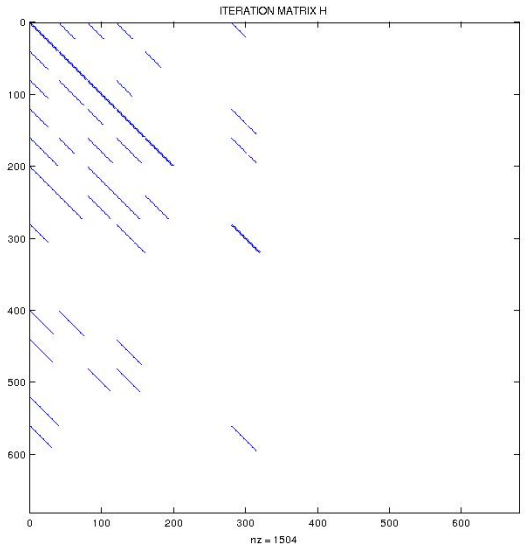

Figure 1.7: FS_680_1 - Sparsity pattern for iteration matrix H.

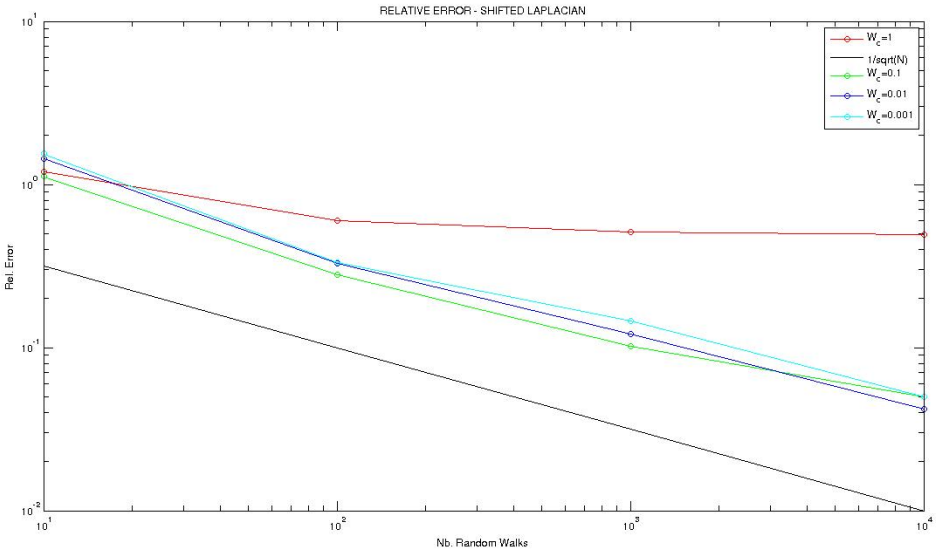

Figure 1.8: Shifted 1D Laplacian - Relative error with a weight cutoff $W_c = 1$ (red), $W_c = 0.1$ (green), $W_c = 0.01$ (blue) and $W_c = 0.001$ (cyan).

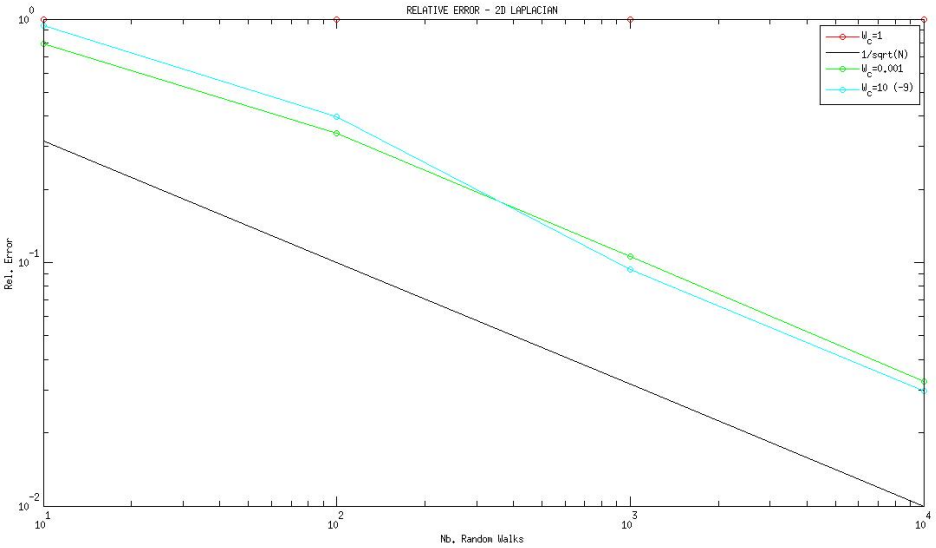

Figure 1.9: 2D Laplacian - Relative error with a weight cutoff $W_c = 1$ (red), $W_c = 0.001$ (green) and $W_c = 10^{-9}$ (cyan).

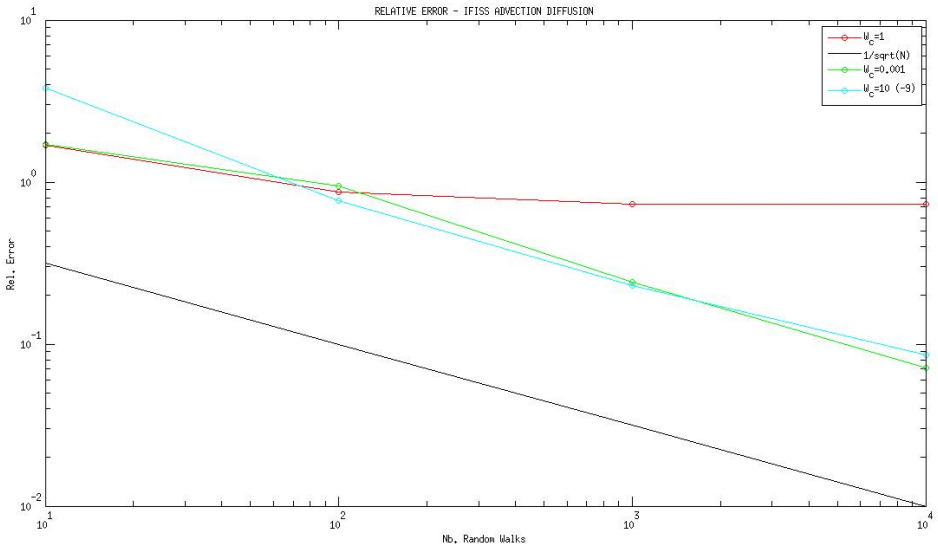

Figure 1.10: IFISS advection diffusion problem - Relative error with a weight cutoff $W_c = 1$ (red), $W_c = 0.001$ (green) and $W_c = 10^{-9}$ (cyan).

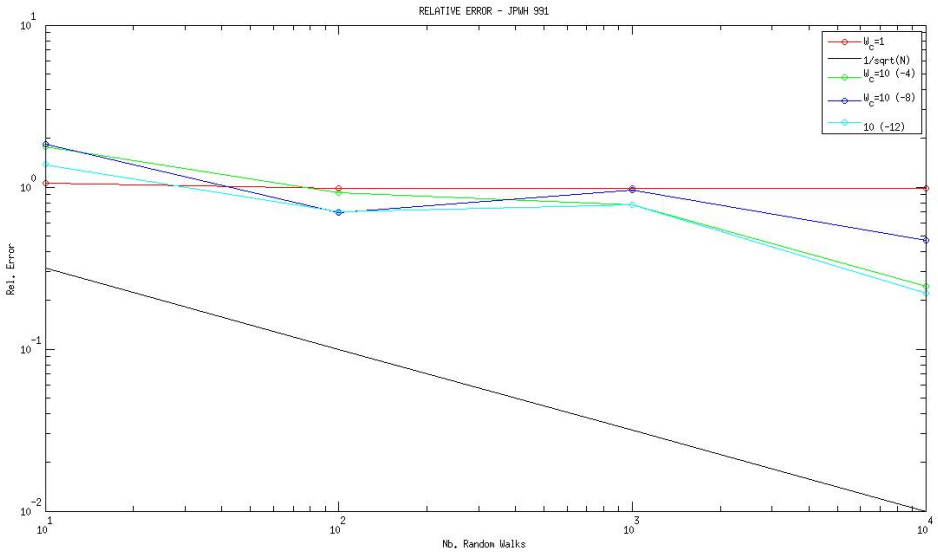

Figure 1.11: JPWH_991 - Relative error with a weight cutoff $W_c = 1$ (red), $W_c = 10^{-4}$, $W_c = 10^{-8}$ (green) and $W_c = 10^{-12}$ (cyan).

Chapter 2

An adaptive selection of the number of histories

In the previous chapter we have focused on the possibility to cut in an accurate way the random walks. This basing on a threshold that considers the magnitude of the weight at each step.

However the length of a truncated random walk is just one of the two degrees of freedom at hand. Indeed we can also pursue an adaptive way to select a reasonable number of histories to be taken. This is possible for controlling the uncertainty associated with the solution.

In order to do this, we can start from taking into account a statistical quantity that intrinsically quantify the uncertainty related to a random variable: the variance. Therefore the attempt in this case is to control the variance associated with the estimated solution. The goal is to reach a condition where this entails taking over the accuracy of the solution as well. In fact there is a strong correlation between uncertainty and accuracy of an estimation.

2.1 Variance-based adaptivity - Forward method

As regards the Forward method, we know that the expression for the variance related to each entry of the solution vector is

$$Var[\theta_i] = \sum_{k=0}^{\infty} \sum_{i_1}^{N} \sum_{i_2}^{N} \cdots \sum_{i_k}^{N} p_{i,i_1} p_{i_1,i_2} \cdots p_{i_{k-1},i_k} \Bigg[w_{i,i_1}^2 w_{i_1,i_2}^2 \cdots w_{i_{k-1},i_k}^2 b_{i_k}^2 + \\ + 2 \sum_{j=0}^{k-1} w_{i,i_1} w_{i_1,i_2} \cdots w_{i_{k-1},i_k} b_{i_k} w_{j,j_1} w_{j_1,j_2} \cdots w_{j_{k-2},j_{k-1}} b_{j_{k-1}} \Bigg] - x_i^2, \quad i = 1, \cdots, d.$$

d is referred to as the size of the problem.

In this context, a reasonable criterion to determine the number $\tilde{N}_i$ of random walks to be run is setting a threshold ε_1 and determine

$$\tilde{N}_i \quad s.t. \quad \frac{\sqrt{Var[\theta_i]}}{|E[\theta_i]|} < \varepsilon_1, \quad i = 1, \cdots, d. \tag{2.1}$$

The dependence of $Var[\theta_i]$ and $E[\theta_i]$ on $\tilde{N}_i$, which seems to be absent in the previous formula, is highlighted by the fact that θ_i is estimated by fixing a finite number of histories. Therefore we are controlling the relative standard deviation requiring it not to be too large. In other words we are pursuing a statistical setting where the uncertainty factor is not dominating over the expected value. This simple adaptive approach can be applied for the estimation of each component x_i. Therefore different number of histories may be employed to compute different entries of the solution vector.

2.2 Variance-based adaptivity - Adjoint method

As we already stressed out in the previous reports, in the Adjoint method each random walk gives contributions for more than one entry. Therefore in this case the selection of the number of random walks is global, since it involves the total number of histories to estimate the entire solution vector.

As concerns the Adjoint method, the estimation of the variance for each entry is

$$Var[\theta_i] = \sum_{k=0}^{\infty} \sum_{i_1}^{N} \sum_{i_2}^{N} \cdots \sum_{i_k}^{N} p_{i_k,i_{k-1}} \cdots p_{i_2,i_1} p_{i_1,i_0} \Big[w^2_{i_k,i_{k-1}} \cdots w^2_{i_2,i_1} w^2_{i_1,i_0} b^2_{i_0} \delta_{i_k,i} + \\ + 2 \sum_{l=0}^{k-1} w_{i_k,i_{k-1}} \cdots w_{i_2,i_1} w_{i_1,i_0} w_{l_{k-1},l_{k-2}} \cdots w_{l_1,l_0} \delta_{i_k,i} \delta_{l_{k-1},i} \Big] - x_i^2, \quad i = 1, \cdots, d. \tag{2.2}$$

A possible adaptive selection of $\tilde{N}$, in this situation, is

$$\tilde{N} \quad s.t. \quad \frac{\|\boldsymbol{\sigma}_{\tilde{N}}\|_1}{\|\mathbf{x}\|_1} < \varepsilon_1, \tag{2.3}$$

where $\boldsymbol{\sigma}$ is a vector whose entries are $\boldsymbol{\sigma}_{\tilde{N},i} = Var[\theta_i]$.

2.3 An a posteriori variance-based adaptivity

What introduced in the two previous section can be exploited in order to build an a posteriori adaptive algorithm, capable to identify the minimal value of $\tilde{N}$ that verifies 2.1 or 2.3 respectively.

Below there are the pseudo-codes associated with both Forward and Adjoint Monte Carlo.

Data: n, ε_1
Result: $\tilde{N}$, σ_i, x_i
For each entry of the solution vector ;
$\tilde{N}_i = n$;
compute θ_i;
while $\frac{\sigma_i}{|E[\theta_i]|} < \varepsilon_1$ **do**
  $\tilde{N} = \tilde{N} + n$;
  compute $E[\theta_i] = x_i$;
end
return θ_i, x_i, σ_i;

Algorithm 1: A posteriori adaptive Forward Monte Carlo

Data: n, ε_1
Result: $\tilde{N}$, $\boldsymbol{\sigma_i}$, $\mathbf{x}$
$\tilde{N} = n$;
compute θ;
while $\frac{\|\boldsymbol{\sigma}\|}{\|E[\theta]\|} < \varepsilon_1$ **do**
  $\tilde{N} = \tilde{N} + n$;
  compute $E[\theta] = \mathbf{x}$;
end
return θ, $\mathbf{x}$, $\boldsymbol{\sigma}$;

Algorithm 2: A posteriori adaptive Adjoint Monte Carlo

2.4 Numerical results

The adaptive threshold is set such that $\varepsilon_1 = 0.1$.

2.4.1 Forward Monte Carlo

As concerns the Forward method, the maximal admitted number of histories for each component is equal to 10 times the length of the solution vector. The number of steps per random walk is initially fixed to a constant number equal to 1000. At each adaptive iteration the number of random walks employed is increased by ten. Even if this is a very little number and it compromises the efficiency of the algorithms in terms of speed, we want to keep it small. In fact for now we want to figure out what is the minimal number of histories required for the fulfillment of the adaptive criterion. In Table 2.1 there are results for all the test cases.

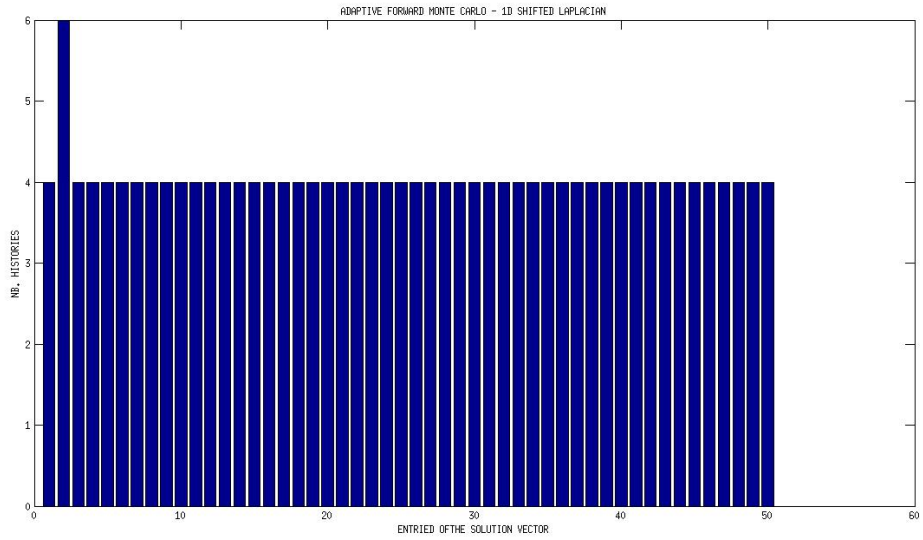

Figure 2.1: Adaptive Forward Monte Carlo - 1d shifted Laplacian - Number of random walks employed for each entry.

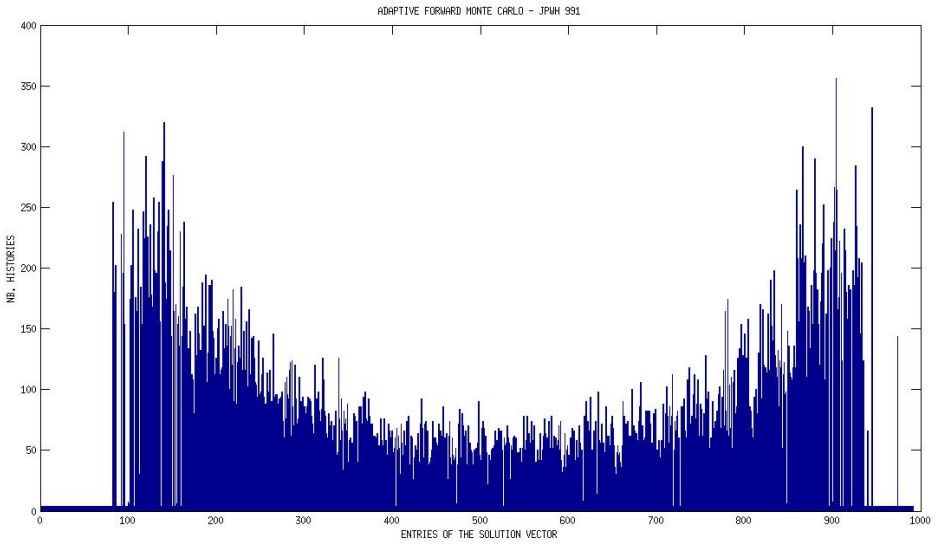

Figure 2.2: Adaptive Forward Monte Carlo - JPWH_991 - Number of random walks employed for each entry.

Type of problem	Relative Error	CPU Time (s)	Nb. Histories
1d shifted Laplacian	0.0097	0.43	202
2d Laplacian	0.132	8995	255150
JPWH_991	0.15	459.3	75376
Marshak Equation	0.0173	6.77	6400 (4 per entry)

Table 2.1: Forward Monte Carlo. Adaptive criterion.

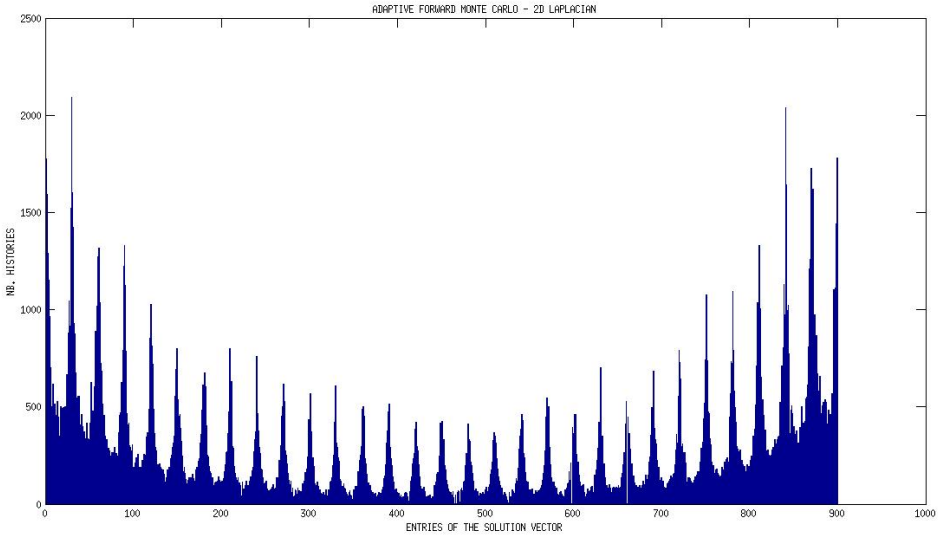

Figure 2.3: Adaptive Forward Monte Carlo - 2d Laplacian - Number of random walks employed for each entry.

2.4.2 Adjoint Monte Carlo

Now we get down to the Adjoint method. At each adaptive iteration the number of random walks employed is increased by two.

No restraints are put for a maximal number of random walks admitted. therefore we let the algorithm free to run until the constraint is verified.

Here below Table 2.2 shows results for the different test cases considered. As we can see, the times employed are very high. However we have to note that this is the standard Monte Carlo algorithm whose rate of convergence is the square root of the random walks computed. Thus we know it to be very slow a priori. Moreover the request of having a relative standard deviation smaller than 0.1 corresponds to a relative error of the same order. Therefore results are coherent with what expected a priori. The only cases to provide kind of poor results are "JPWH_991" and "FS_680_1". A likely explanation for this may be related to the particular structure of the sparsity pattern of the associated iteration matrices (see Figures 1.7 and **??**).

Type of problem	Relative Error	CPU Time (s)	Nb. Histories
1d shifted Laplacian	0.073	42.19	1826
2d Laplacian	0.087	246	1252
ifiss advection diffusion	0.074	1529	9934
Marshak equation	0.097	2008	8020
JPWH_991	0.1197	18704	51670
FS_680_1	0.265	46244	97816

Table 2.2: Adjoint Monte Carlo. Adaptive criterion.

The algorithms to detect the number of histories to run and the history's length cutoff can be combined in a unique adaptive approach. The threshold for the weight cut-off is set to 10^{-6}. Results are shown in Table 2.3. By comparing the time spent we can see that the new automatic

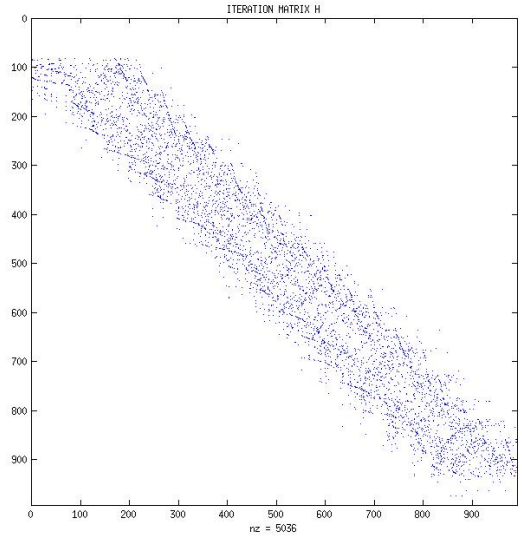

Figure 2.4: JPWH_991 - Sparsity pattern for iteration matrix H.

approach is highly competitive with the previous one. In fact the time spent is much less overall, without affecting the accuracy of the solution.

Type of problem	Relative Error	CPU Time (s)	Nb. Histories
1d shifted Laplacian	0.098	2.06	1838
2d Laplacian	0.089	198.5	1280
ifiss advection diffusion	0.083	933.9	9800
Marshak equation	0.094	907.86	8022
FS_680_1	0.264	76258	74976

Table 2.3: Adjoint Monte Carlo. Adaptive criterion.

By looking at the behavior of the residual, we verify what expected after the reasoning accomplished in the Appendix. In fact the residual decreases with respect to the number of histories with a slope of $\frac{1}{\sqrt{n}}$. For a graphical representation of this look at Figures 2.5, 2.6, 2.7 and 2.8.

2.5 A stationarity check for the a posteriori adaptivity

The a posteriori approach introduced before is very naive and it does not take into account whether the number of histories is enough to reach the asymptotic trend or not. In order to do this we need to look at the value of the variance with respect to two consecutive adaptive steps. For the Forward method it means that we need to set the number of histories $\tilde{N}_i$ such that

$$\frac{|\sigma_i^{n-1} - \sigma_i^n|}{|x_i|} < \varepsilon_2 \quad i = 1, \cdots, d \tag{2.4}$$

where d is the size of the problem, n is a generic step of the adaptive algorithm and ε_2 is a threshold set for the check of stationarity.

As concerns the Adjoint method instead we need to pursue $\tilde{N}$ such that

$$\frac{\|\boldsymbol{\sigma}^{n-1} - \boldsymbol{\sigma}^n\|_1}{\|\mathbf{x}\|_1}. \tag{2.5}$$

2.4 can be combined with 2.1and 2.5 can be combined with 2.3. In this way we can build up a more robust adaptive approach that takes into account the relative magnitude of the variance and the achievement of the asymptotic behavior as well. In fact taking few histories to determine the

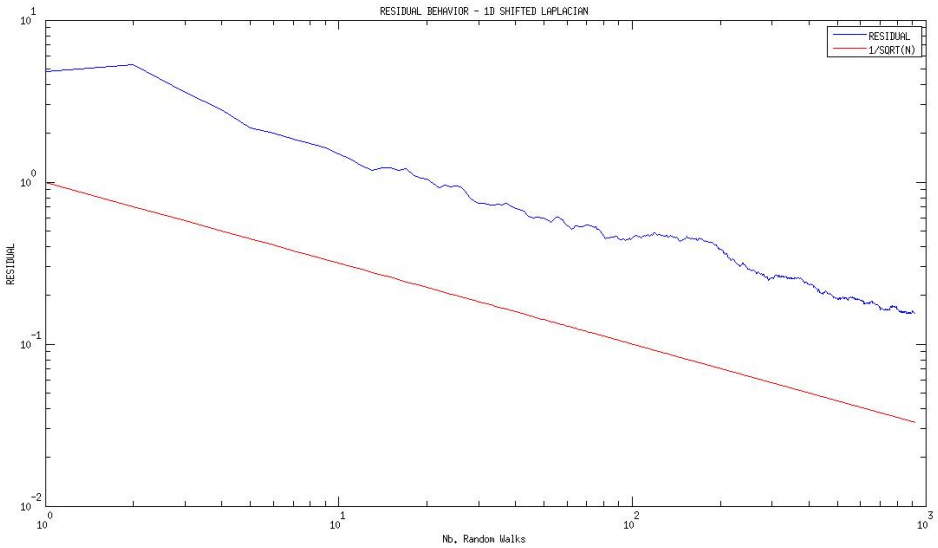

Figure 2.5: 1d shifted Laplacian - Residual behavior.

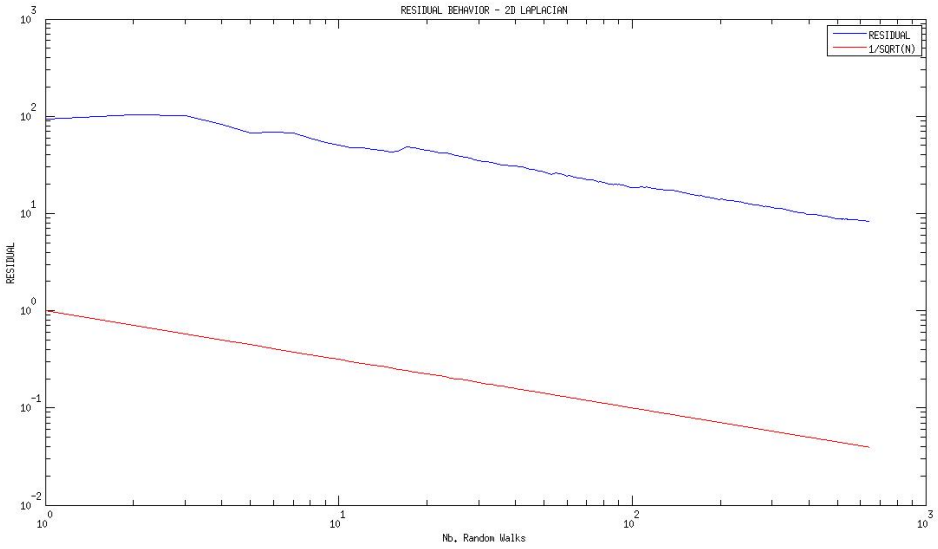

Figure 2.6: 2d Laplacian - Residual behavior.

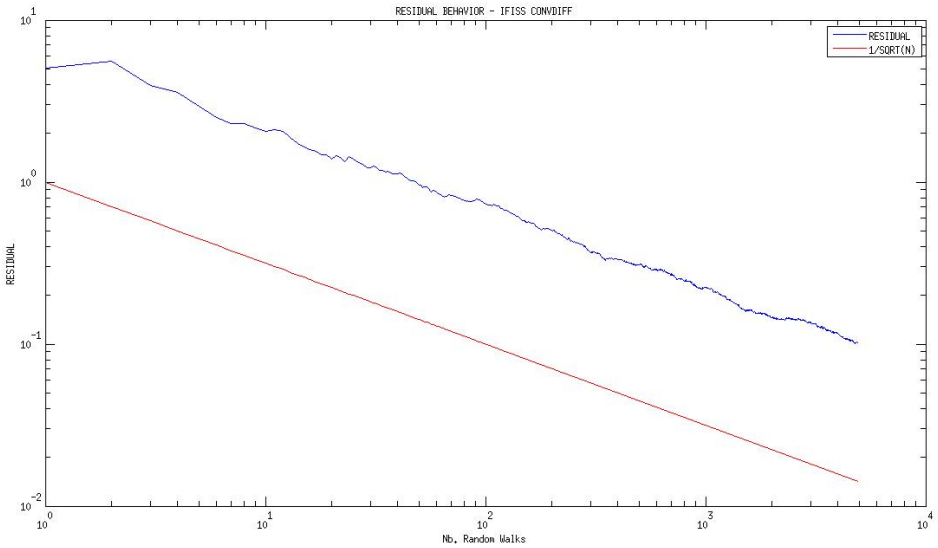

Figure 2.7: IFISS Advection Diffusion problem - Residual behavior.

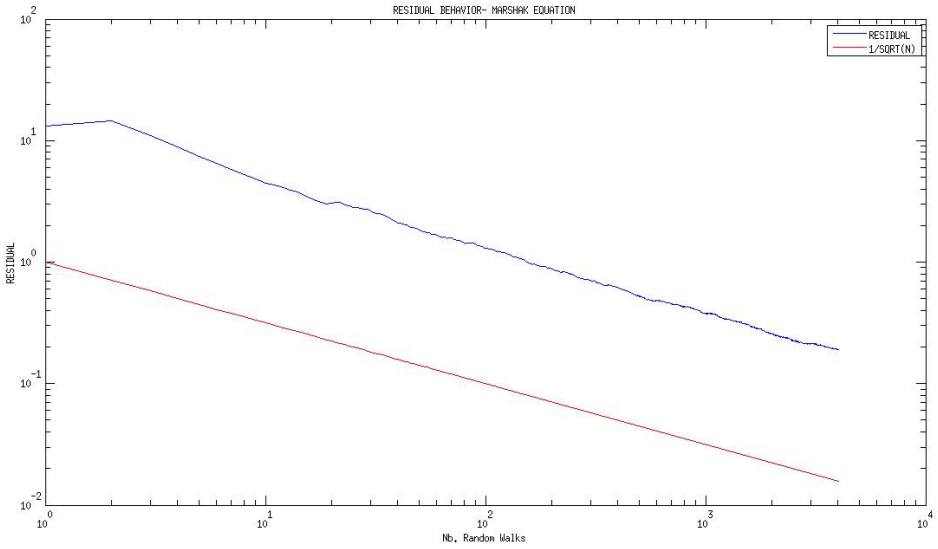

Figure 2.8: Marshak Equation - Residual behavior.

estimation of the solution may compromise severely the accuracy of the solution itself, especially in the viewpoint of applying this approach inside of a numerical scheme such as MCSA.

In order to verify the usefulness of using an additional stationarity check such as 2.4 and 2.5, we can tally the number of times the second criterion force the history to continue. For the sake of simplicity we call this phenomenon the rejection, since it represents the situation where the first criterion (2.1 or 2.3 respectively) would suggest to cut the random walk with the second criterion rejecting this proposal.

Below we show the results obtained by combining 2.3 and 2.5 for the Adjoint method. The test cases are the same as the ones we used in the previous section. The threshold for both the criterion is set to $\varepsilon_1 = \varepsilon_2 = 0.1$.

It is discovered that for the 1d shifted Laplacian and for the advection diffusion problem the check 2.5 forces the code to run even if 2.3 would suggest to cut the history. It means that the relative standard deviation is already lower than ε_1 but the slope of the standard deviation is not flat enough to catch the asymptotic behavior. Moreover, also in this case the request of having a relative standard deviation smaller than 0.1 forces the relative error to be of the same order. Thus it points out even more the correlation between uncertainty and accuracy associated with the solution in a stochastic linear solver.

By comparing the performance in 2.3 with the performance in 2.4 we can see that there is not much difference either in terms of number of random walks employed or in terms of time. Thus it looks like the stationarity check does not provide any improvement with respect to the only relative standard deviation.

Type of problem	**Relative Error**	**CPU Time (s)**	**Rejections**	**Nb. Histories**
1d shifted Laplacian	0.098	2.28	205	1836
2d Laplacian	0.089	164.48	0	1280
ifiss advection diffusion	0.083	630.18	2711	9800
Marshak equation	0.094	593.66	0	8018

Table 2.4: Adjoint Monte Carlo. Adaptive criterion with stationarity check.

Chapter 3

Monte Carlo Synthetic Acceleration with adaptivity

The approaches explored so far can be used inside of an accelerating scheme (such as MCSA), in order to compute automatically the number of required histories at each numerical iteration.

We focus now on the adaptive algorithm given by the combination of the history cutoff with 2.1 and 2.3 respectively.

3.1 Numerical results

The test cases used are the same as the ones used so far and the different thresholds involved for the adaptive algorithms are kept with the same values as well.

As concerns the MCSA with the Forward method, the stopping threshold on the residual is set to 10^{-3} (based on the relative residual), instead the adaptive threshold for the relative standard deviation is set to 0.5. The maximal number of numerical iterations is 300, the minimal number of random walks for each entry is 10, the cut-off threshold is 10^{-6} and the maximal number of steps admitted for each random walk is 1000.

3.1.1 Forward MCSA

In Table 3.1 there is a synthesis containing results associated with the different test cases. For the Thermal Equation the total number of numerical iterations employed is 2. At the first numerical iterations the distribution of required random walks is 10 for each entry. Instead at the second numerical iteration we have a distribution represented in Figure **??**.

Type of problem	**Relative Residual**	**Relative Error**	**Numerical Iter.**	**CPU Time (s)**
1d shifted Laplacian	$1.76 \cdot 10^{-4}$	$7.98 \cdot 10^{-5}$	2	2.73
2d Laplacian	-	-	-	-
ifiss advection diffusion	$5.39 \cdot 10^{-4}$	$3.18 \cdot 10^{-4}$	11	186550
JPWH_991	$5.91 \cdot 10^{-4}$	$2.88 \cdot 10^{-5}$	15	65956
FS_680_1	$8.24 \cdot 10^{-4}$	$8.36 \cdot 10^{-4}$	69	53369
Marshak Equation	$6.62 \cdot 10^{-4}$	$6.57 \cdot 10^{-4}$	2	250.78

Table 3.1: Forward MCSA.

3.1.2 Adjoint MCSA

Type of problem	**Relative Residual**	**Relative Error**	**Numerical Iter.**	**CPU Time (s)**
1d shifted Laplacian	$3.99 \cdot 10^{-4}$	$2.67 \cdot 10^{-5}$	6	0.94
2d Laplacian	-	-	-	-
JPWH_991	$9.56 \cdot 10^{-4}$	0.002	246	??
FS_680_1	-	-	-	-
ifiss advection diffusion	$9.68 \cdot 10^{-4}$	0.0323	166	41645
Marshak Equation	$7.69 \cdot 10^{-4}$	$8.94 \cdot 10^{-4}$	2	460

Table 3.2: Forward MCSA.

Appendix

Assume θ is an estimator for a quantity x which is unknown. It holds

$$Var[\theta] = E[\theta^2] - (E[\theta])^2.$$

The theoretical variance can be estimated with a sample quantity such as the sample variance, defied as

$$\hat{\sigma} = \frac{1}{(n-1)} \sum_{i=1}^{n} (\theta_i - \overline{\theta})^2, \quad \overline{\theta} = \frac{1}{n} \sum_{i=1}^{n} \theta_i. \tag{3.1}$$

Each θ_i is a random variable associated with a single random walk. The expected value of all the random walks return the estimator we are interested in. Therefore the sample size n is the total number of histories run. Since all the random walks are independent one of the other. Therefore all the realizations of the estimator θ_i are independent as well.

This means that

$$cov(\theta_i, \theta_j) = 0, \quad i \neq j.$$

This consideration is very important for our applications. In fact it implies that the standard deviation and the statistical error both associated with the standard Monte Carlo scales with the same order.

The statistical error is defined as

$$err = \sqrt{\frac{1}{n} \left[\sum_{i=1}^{n} (\theta_i - x) \right]^2}. \tag{3.2}$$

By comparing the second power of 3.2 with 3.1, we realize that they are not the same thing. In fact in 3.2 there are middle terms neglected in 3.1. However the independence between the θ_i makes this middle term approach to zero as the sample size $n \to \infty$. In fact the middle terms represent the sample covariances, which approach the value of the theoretical covariances for a reasonably high value of n. Since the mixed terms are almost null, 3.2 and 3.1 tend to be the same thing as $n \to \infty$.

Given a linear system

$$A\mathbf{x} = \mathbf{b}, \quad A \in \mathbb{R}^{d \times d}, \ \mathbf{x}, \mathbf{b} \in \mathbb{R}^d$$

and given $\hat{\mathbf{x}}$ the standard Monte Carlo solution, the residual is defined such as

$$res = \mathbf{b} - A\hat{\mathbf{x}} = A(\mathbf{x} - \hat{\mathbf{x}}).$$

It entails that the residual has the same behavior as the error. In fact the former is a linear transformation of the latter.

Bibliography

[AAB+05] V. Alessandrov, I. Atanassov, E. amd Dimov, S. Branford, A. Thandavan, and C. Weihrauch. Parallel hybrid Monte Carlo algorithms for matrix computations problems. *Lecture Notes in Computer Science*, 3516:752–759, 2005.

[BS97] M. Benzi and B. Szyld. Existence and uniqueness of splittings for stationary iterative methods with applications to alternating methods. *Numerische Matematik*, 76:309–321, 1997.

[BU] M. Benzi and B. Ucar. Product preconditioning for Markov chain problems.

[BU07] M. Benzi and B. Ucar. Block trianguar preconditioners for M-matrices and Markov chains. *Electronic Transaction on Numerical Analysis*, 26:209–227, 2007.

[DA98] I.T. Dimov and V.N. Alexandrov. A new highly convergent Monte Carlo method for matrix computations. *Mathematics and Computers in Simulation*, (47):165–181, 1998.

[DAK01] I. Dimov, V. Alexandrov, and A. Karaivanova. Parallel resolvent Monte Carlo algorithms for linear algebra problems. *Mathematics and Computers in Simulation*, 55(55):25–35, 2001.

[EMSH14] T.M Evans, S.W. Mosher, S.R. Slattery, and S.P. Hamilton. A Monte Carlo synthetic-acceleration method for solving the thermal radiation diffusion equation. *Journal of Computational Physics*, 258(November 2013):338–358, 2014.

[ESW02] T.M. Evans, S.R. Slattery, and P.P.H Wilson. Mixed Monte Carlo parallel algorithms for matrix computation. *International Conference on Computational Science 2002*, 2002.

[ESW13] T.M. Evans, S.R. Slattery, and P.P.H Wilson. A spectral analysis of the domain decomposed Monte Carlo method for linear systems. *International Conference on Mathematics and Computational Methods Applied to Nuclear Science & Engineering*, 2013.

[Hal62] J.H. Halton. Sequential Monte Carlo. *Mathematical Proceeding of the Cambridge Philosophical Society*, 58(1):57–58, 1962.

[Hal94] J.H. Halton. Sequential Monte Carlo techniques for the solution of linear systems. *Journal of Scientific Computing*, 9(2):213–257, 1994.

[Sla13] S. Slattery. *Parallel Monte Carlo Synthetic Acceleration Methods For Discrete Transport Problems*. PhD thesis, University of Wisconsin-Madison, 2013.

[Sri00] A. Srinivasan. Monte Carlo linear solvers with non-diagonal splitting. *Mathematics and Computer in Simulation*, 80:1133–1143, 2000.

[Vaj07] B. F. Vajargah. Different stochastic algorithms to obtain matrix inversion. *Applied Mathematics and Computation*, 189:1841–1846, 2007.