Skolkovo Institute of Science and Technology

Thesis Changes Log

Name of Candidate: Vladimir Fanaskov
PhD Program: Mathematics and Mechanics
Title of Thesis: Statistical inference and machine learning in numerical linear algebra
Supervisor: Associate Professor Aslan Kasimov

The thesis document includes the following changes in answer to the external review process.

We express gratitude to the reviewers for careful proofreading and helpful recommendations. In all but rare cases, we followed the suggestions and introduced appropriate changes.

Introduction

This text below includes rebuttal and description of changes introduced to the thesis in answer to the external review process.

Notation

When we need to provide a quote from the review, we use a bold font as follow:

> **Name of the reviewer: "This part represents a quote from the review."**

Horizontal lines as above are used to separate quotes.

Rebuttal

Nikolai Brilliantov: "Please make a careful proofreading of the thesis text. There exist misprints and errors. For instance, in Eq. (10.11) the factor $(1/h^2)$ is missed [...]. Besides, the meaning of Eq. (10.17) is questionable: for both cases, $i = 1, 2$ the same result is obtained.."

- The factor $1/h^2$ is not missing from Eq. (1.16) (former Eq. (10.11)). This equation is finite-element discretization, so the weak form is used, meaning in $D = 2$ there are two integrals $\int dx$, $\int dy$ and two first derivatives (after integration by parts). On the isotropic grid this gives h^2 from integrals and $1/h^2$ from derivatives so, overall, the factor is 1, not $1/h^2$.

- The meaning of Eq. (1.22) (former Eq. (10.17)) is quite clear, it certainly does not give the same results for $i = 1, 2$. Indeed, if one look at the equation

$$g_i(x, y) = \begin{cases} (2-i)\sigma + (i-1), & \text{if } (x-0.5)(y-0.5) \geq 0; \\ (i-1)\sigma + (2-i), & \text{if } (x-0.5)(y-0.5) < 0. \end{cases} \tag{14.4}$$

  One gets for $i = 1$

$$g_1(x, y) = \begin{cases} \sigma, & \text{if } (x-0.5)(y-0.5) \geq 0; \\ 1, & \text{if } (x-0.5)(y-0.5) < 0, \end{cases} \tag{14.5}$$

  and for $i = 2$ one has

$$g_2(x, y) = \begin{cases} 1, & \text{if } (x-0.5)(y-0.5) \geq 0; \\ \sigma, & \text{if } (x-0.5)(y-0.5) < 0. \end{cases} \tag{14.6}$$

  In our applications $\sigma \gg 1$, so $g_1(x, y)$ is large in the first and the third quadrant whereas $g_2(x, y)$ is large in the second and fourth quadrants.

Anh-Huy Phan: "Page 135, Algorithm 11 What is outcome of Algorithm 11?"

The set of actions (that is, the policy) is the outcome. This outcome is implicit in a sense that it is not returned as some object.

To provide an example that the situation is normal in reinforcement learning, we include here the bandit algorithm from [SB98, Section 2.4]:

A simple bandit algorithm

Initialize, for $a = 1$ to k:
$Q(a) \leftarrow 0$
$N(a) \leftarrow 0$

Loop forever:
$$A \leftarrow \begin{cases} \arg\max_a Q(a) & \text{with probability } 1-\varepsilon \quad \text{(breaking ties randomly)} \\ \text{a random action} & \text{with probability } \varepsilon \end{cases}$$
$R \leftarrow bandit(A)$
$N(A) \leftarrow N(A) + 1$
$Q(A) \leftarrow Q(A) + \frac{1}{N(A)}\big[R - Q(A)\big]$

As we can see, the algorithm above return nothing, and the policy is implicitly specified in the same way as in our algorithm.

Anh-Huy Phan: "Why does the agent go with her on page 183?"

The agent is a word that is not gender-specific. The same is true for most of the nouns in English (see the article https://dictionary.cambridge.org/grammar/british-grammar/nouns-and-gender). This mean we are at liberty to pick up a gender we prefer.

Anh-Huy Phan: "- Page 87 "instationary": should be "nonstationary"""

The term "instationary" is routinely used in the literature on iterative methods. We provide several examples from classical book [Hac16]:

1. Page 172 after equation (7.55): "For a fixed (or variable) step size Δt, recursion (7.55) describes the stationary (or **instationary**) Richardson method."

2. Page 204 after Exercise 8.43.: "A general convergence result of this kind (also for **instationary** ADI methods) is due to Alefeld [1]. Here, we call the method stationary if ω is constant during the iteration and **instationary** if it varies (as, e .g., it is assumed throughout the following section)."

3. Page 207 after Remark 8.47.: "Hence, the **instationary** ADI method permits not only halving of the order (for the case $m = 1$, compare also with Exercise 8.43b), but any arbitrarily small (and hence very favourable) order can be reached for sufficiently large m."

Anh-Huy Phan: "- Page 136 Provide applications where we need to solve the same linear system more than once."

The whole page was devoted to the examples of when one needs to solve the same linear system more than once. The quote is as follows:

> There are plenty of situations in scientific computing when precisely the same linear system (or a family of related linear systems) is solved repeatedly for different right-hand sides. The following list contains a few relevant examples:
>
> 1. Chorin's projection method [Cho67].
>
>   In this splitting scheme, the Poisson equation needs to be solved during each time step to recover pressure from auxiliary velocity.
>
> 2. Crank-Nicholson scheme for heat equation Section 1.3.10.
>
>   As in the case of any implicit scheme, one needs to solve a linear system to obtain an approximation for the next time step.
>
> 3. Quasi-geostrophic omega equation [BS10].
>
>   This anisotropic elliptic equation is solved during time marching when weather forecasts are performed.
>
> 4. PDE constrained optimization [Bie+03].
>
>   In this scenario, one requires to solve the same PDE for different input data (boundary conditions, initial conditions, right-hand-side, values of parameters, etc), which typically requires the solution of linear systems. See [RDW10] for a concrete example where the constraints are formed by elliptic PDE.
>
> 5. Inverse problems [Tar05].
>
>   The usual strategy to solve inverse problems includes Monte Carlo techniques [Tar05, Chapter 2], which require repeated solutions to forward problems. In the case of certain discretization of PDE, this, again, leads to large sparse linear systems.

6. More examples can be found in literature on Krylov subspace recycling techniques (see [Par+06] and references therein).

In addition, we introduced one more example from the present thesis. The added text reads

> There is also a simple example of this situation in the present thesis. Namely, the optimal heating problem discussed in Section 4.4.3 leads to the situation when the heat equation needs to be solved multiple times.

Anh-Huy Phan: "-Page 109, (6.14) What are the differences between matrices S_l, for l = 1,2,...? The same question is for matrices I_l."

The definitions of $\boldsymbol{S}_l$ and $\boldsymbol{I}_l$ read

$$(\boldsymbol{S}_l)_{ij} = \delta_{ij+1}, (\boldsymbol{I}_l)_{ij} = \delta_{ij},\ i,j = 1,\ \dots\ ,\ 2^l,$$

So l defines the number of elements in the matrix ($\boldsymbol{S}_l$ is of size $2^l \times 2^l$).

Anh-Huy Phan: "-Page 56 Why is the factor s common?"

The explanation can be found on the same page

> Common factor s appears in Lemma 4.3.2 because if we take $\boldsymbol{\Sigma}_0 = \boldsymbol{V}\boldsymbol{V}^T + s\boldsymbol{\Psi}$, posterior distribution for the scale $p\left(s|\boldsymbol{W}^T\boldsymbol{A}\boldsymbol{x} = \boldsymbol{W}^T\boldsymbol{b}\right)$ coincides with $\mathsf{IG}\left(s|\alpha,\beta\right)$, that is available information is insufficient to fix the scale.

Anh-Huy Phan: "- Page 38, results shown in Figure 2.4. Why is GaBP robust? GaBP does not converge faster than Gauss-Seidel. GaBp(k) with different sweeps k = 1, 2, 3 do not have the same or comparable complexity to update x^m per iteration. Hence it is nonsense to compare converge of GaBP(k) with different k and Gauss-Seidel."

We can not agree with this comment.

First, it is explained in the text why GaBP is more robust. The relevant fragment reads

> First, from results in Fig. 3.4 we can see that GaBP is robust, unlike line Gauss-Seidel smoothers the performance of which depends on the direction of anisotropy.

To recapitulate, GaBP is considered more robust, because one does not need manually select the direction of anisotropy. For more general anisotropy patterns, GaBP will perform better that both line Gauss-Seidel smoothers.

Second, the claim that "GaBP does not converge faster than Gauss-Seidel" is not supported by the presented numerical experiment. On the contrary, we can see in both pictures that the one-directional variant of Gauss-Seidel smoother stagnates. It can be estimated from the pictures that one variant of Gauss-Seidel drops relative error by a factor of 10 after roughly 50 iterations. It means, one will need $\simeq 14 \times 50 = 700$ iterations to achieve error comparable to GaBP converged within 50 iterations.

Third, the claim that "GaBp(k) with different sweeps k = 1, 2, 3 do not have the same or comparable complexity to update x^m per iteration" is, again, not true. GaBP and Gauss-Seidel methods are comparable because both of them have the same asymptotic complexity $O(N)$. Moreover, we know exactly how to compare both solvers because we have a table with computational complexities (Table 3.1).

For example, consider GaBP(1). It has a complexity of $24N$ per iteration, whereas line Gauss-Seidel has a complexity of $9N$ per iteration. Now, we computed that Gauss-Seidel, if the wrong direction is chosen, needs $\simeq 700$ iterations, and GaBP(1) ≤ 50. So, if we take into account the computational budget, one iteration of GaBP(1) corresponds to three iterations of line Gauss-Seidel. This means we can divide 700 iterations by 3 and find that they correspond to $\simeq 230$ iterations of GaBP(1). Since $50 < 230$, line Gauss-Seidel is much less efficient in this case.

> **Anh-Huy Phan: "There is no sweep parameter k in Algorithms 1 and 2."**

The sweep parameter k is present in Algorithm 3 and the explanation in Section 3.5 suggests that k refers to the number of sweeps of GaBP as smoother. Besides, the number of iterations k is presented in mentioned algorithms implicitly in the part "while not converge do". The meaning of this phrase is also discussed after Algorithm 1.

Anh-Huy Phan: "- Summary on page 43 "Red-black has better convergence rate than other solvers" Again, since different solvers have different complexity per iteration, the number of iterations is not an appropriate measure for the convergence rate. For example GaBP(3) with k = 3 sweeps should demand lower number of "outer" iterations than GaBP(2), but it does not mean that GaBP(3) converges faster than GaBP(2)."

There are a few inaccurate statements in this comment.

First, the number of iterations (required to drop the norm of error by a certain amount) is the appropriate measure of the convergence rate. In fact, for symmetric positive definite matrices, the number of iterations required to drop the norm of error by a certain factor is directly related to the convergence rate. For more information see [Hac16, Section 2.2.6]. Presumably, what reviewers meant is that the convergence rate does not define the efficiency of the solver.

Second, it is common knowledge that it is pointless to compare the number of iterations required for convergence without considering the number of floating point operations required for each iteration. Precisely for that reason, we discussed in detail the effective amount of work at the beginning of Section 3.6 and list these metrics in all tables in the section with numerical results.

Anh-Huy Phan: "-Parallel version (implementation) of the proposed GabP algorithm, pages 28. Do the two implementations converge to the same solution? Is the parallel version faster than Algorithm 1?"

The different meanings of parallel implementation are discussed after the Algorithms, right before Theorem 3.3.1, and in Section 3.6.1. The discussion section contains explanation on the speed of convergence, efficiency and degree of parallelism for GaBP solvers considered in the chapter.

Presented proofs for the convergence of GaBP ensure that all schedules that cover all edges of the graph converge to the same solution.

Changelog

Major changes

Evgeny Burnaev: "First, the belief propagation for nonsymmetrical systems in the second chapter lacks statistical interpretation. Surely, it is impossible to construct a normal Markov random field with a nonsymmetric matrix. I suggest to briefly discuss the interpretation or the lack of it."

We supplied Section 3.3.4 with the requested interpretation. In this section we briefly explain that GaBP for non-symmetric matrices is related to Monte Carlo inversion algorithm of matrix introduced by John von Neumann.

Evgeny Burnaev: "Second, Chapter 7 is a little light on considered architectures. It seems that the main advantage of the method is flexibility, so more experiments for architectures with more aggressive coarsening and larger interpolation/relaxation stencils would be helpful for understanding."

We introduced Section 8.6.4 where the requested experiments with different stencils are performed. Here we also discussed the results for more aggressive coarsening.

Evgeny Burnaev: "Third, Chapter 8 contains incomplete results and lacks theoretical justifications. I understand that in general, it is a difficult task to provide guarantees for online optimization of the solver for a sufficiently general linear system, but I suggest the author discuss possible lines of attack on the problem, or some back-of-the-envelope estimations, or heuristics explanations on why algorithms are going to converge."

We introduced Section 9.5 with a discussion of convergence of proposed algorithms.

Lei Zhang (comment): "I have a general question on the performance of probablistic and machine learning algorithms on more difficult and more practical problems, for example, elliptic problem with high contrast/highly oscillatory coefficients, Helmholtz equation with high wave numbers, and convection dominated flows, or even high Reynolds Navier-Stokes equations, to name a few. I can see there are some brief discussions

in the context of e.g., BPX preconditioners, but usually the condition number is not so large. It would be helpful to point out the difficulties and discuss the possible remedies in the thesis, for example, through one example, such as the dependence on the contrast of elliptic coefficient."

Nikolai Brilliantov: "It would be worth to add a special section where the discussion about the limitations of the applied methods should be given. I admit that such a discussion is distributed piecewise through the thesis, however I suggest accumulating them in one place, possibly in the conclusion."

We rewrote the conclusion with a specific emphasis on the limitations of the proposed approaches. We also included examples of difficulties ML approaches usually encounter when dealing with practical problems and possible remedies to those difficulties.

Several reviewers suggested rewriting the introduction and altering the structure of the thesis:

Anh-Huy Phan: "- Introduction section presents the goal of the research, e.g., study relations between statistical inference, machine learning and iterative algorithms, study probabilistic numerical algorithms for solving sparse linear systems. The goal and contribution of the thesis are clear. However, the motivation of the research and formulation of the studied problems are missing. Why are the studied problems important, e.g., probabilistic algorithms and generalization of BPX conditioners using machine learning methods, applying Bayesian analysis to solving linear systems? What are the benefits of the proposed algorithms, e.g., the probabilistic Richardson algorithm?"

Anh-Huy Phan: "- Appendices 9 and 10 should be presented in the Introduction or Preliminary section or a section for formulated problems. This will help the readers to understand an overview of the challenging problems considered in the thesis. Important concepts should be presented in the preliminary section, e.g., existing probabilistic algorithms, well and poorly calibrated system, "instationary" Richardson algorithm, multigrid method based on Deep NNs."

Nikolai Brilliantov: "I suggest expand ing the introduction adding a more detailed explanation, how different approaches used in the the sis a re related to each other and to the solution of linear problems in general. Mor eove r, the author shoul d invest for efforts to make the intro duction chapter more read able for non experts, probably adding more simple explanatory examples."

Luiz Faria: "Can the background material in chapter 9 be moved to the introduction, or earlier in the manuscript? I found it uncomfortable to have many forward references when reading the first sections, and I would have found it useful to read that review in e.g. the introduction."

In response to that, we completely rewrote the introduction. The main changes are as follows:

1. Material from appendices 9 and 10 now appears in the introduction.
2. We provided a general explanation of the goals of the research.
3. We introduced more clearly the motivation behind the proposed method and explained what these methods achieve.
4. We added a more accessible summary of the results of the thesis.

Minor changes

Nikolai Brilliantov: "in Eq. (10.17) there is a misprint "$y2$""

"$y2$" was changed on y

Alexey Zaytsev: "P. 1: the title has 2021 year instead of 2022"

Anh-Huy Phan: Frontpage "Moscow 2021" $->$ "2022"

The year on the frontpage was changed from 2021 to 2022.

Alexey Zaytsev: "P. 145: proposed algorithms we on a more serious examples − > proposed algorithms on more serious examples"

Anh-Huy Phan: page 145 "we on a more serious examples we perform a series of . . . "

We followed the suggestion of Alexey Zaytsev.

Alexey Zaytsev: "P. 188: prove it's form − > prove its form"

We followed the suggestion.

Nikolai Brilliantov: "I believe that the reference to the Boltzmann distribution in Eq. (2.6) is misleading (there are no important parameter temperature there). Hence it should be either explained in more detail or removed."

Eq. (3.6) (former Eq. (2.6)) explains the relation of graphical models to statistical physics. The temperature parameter is not of practical value for graphical models, but the relation itself is heavily exploited. We add relevant references in the footnote in the paragraph after Eq. (3.6).

Anh-Huy Phan: "we drawn 50 matrices" (page 138) (drew)

Anh-Huy Phan: "This parameters" (page 137)

Anh-Huy Phan: "are already present" on page 137

Anh-Huy Phan: "Both of this routes" page 14

Anh-Huy Phan: "Algorithm 4 simply perform an additional run"

We corrected these grammatical errors.

Anh-Huy Phan: "- Inconsistent notation for matrices, vectors [...]"

In general our notation for matrices is that we denote the matrix by boldface (e.g., $\boldsymbol{A}$), and matrix element goes with the light font (e.g., A_{ij}).

Anh-Huy Phan: "+ Boldface for matrices on page 16, but italic font on pages 14, 23, (2.32) on page 33, Algorithm 2, error propagation matrix M on page 37"

Suggested cases

- pages 14
- page 23
- Eq. (3.33) (former (2.32))
- Algorithm 2

are consistent with our rules so we have nothing to correct here. Note that in Eq. (3.33) the matrix consists of the matrix blocks, so $\boldsymbol{A}_{11}$ is itself a matrix and should come in boldface.

The "error propagation matrix M on page 37" was not in boldface, so we corrected this part.

There were other cases with the same problem:

Anh-Huy Phan: "- Page 50: $W^T AW$ A should be A."

Anh-Huy Phan: "- Page 65 G = I should be denoted as two matrices."

Anh-Huy Phan: "- Page 78 Check notation of v0 and matrices A, V"

Anh-Huy Phan: "- Page 87 N = I should be denoted as matrices"

Anh-Huy Phan: "- Page 88 Check notation for vector μ and matrix Σ in (4.48) and (4.49)"

Anh-Huy Phan: "- Page 105 Check notation for the matrices R and A in section 6.1.2: here R is an easy invertible approximation to A"

In these cases, we corrected fonts as the reviewer suggested.

Anh-Huy Phan: "- Contribution "Neural architecture is equivalent to the multigrid solver" and Section 7 "Neural architecture" is not a research topic, but neural architecture search or neural architecture design for a specific task."

The part in the introduction that lists contributions of the thesis was there for the convenience of the reviewers:

> *To help the reviewers of this manuscript, we end the introduction with a list of main contributions of the present work.*

This part was removed for the final version of the thesis.

Current introduction better reflects the idea, that we study "neural architecture design".

Anh-Huy Phan: "+ Normal distribution on page 79 N(μ, Σ) and page 13 N(μ, Σ)."

We corrected the normal distribution on page 79.

Anh-Huy Phan: """Lanczost algorithm" shall be "Lanczos" algorithm."

We introduced requested changes.

Anh-Huy Phan: "-Page 58, definition of S-statistic in Lemma 3.3.5 The first term $(x - \widetilde{x})$ misses a transpose operator."

We introduced transpose operator.

Anh-Huy Phan: "- Page 139, Algorithms 12 and 13 $M(\theta)$ denotes the error propagation matrix, but in Algorithm 12, the same notation represents the iterative method."

We clarified that the error propagation matrices $\boldsymbol{M}(\theta)$ that define the family of iterative methods should be supplied to the algorithm.

Anh-Huy Phan: "Line 4 and Line 11, $\|x - y|$ should be $\|x - y\|$"

We introduced requested changes.

Anh-Huy Phan: "Page 135, Algorithm 11 The running index j changes inside the for loop at Line 6 or Line 8. This affects the output at line 11."

We fixed the outer index that represent the round to j.

Anh-Huy Phan: "Page 134, what are R1, ..., RN in (8.2)?"

These are rewards. We added the explanation under the equation.

Anh-Huy Phan: "MDS?"

MDS was changed to MDP.

Anh-Huy Phan: "- Page 103 What does $B_{\mathbf{BPX}}$ in (6.5) improve the pre-conditioner over that in (6.4)?"

We reformulated the sentence as follows:

"To introduce additional parameters to BPX preconditioner and later use them in optimization of condition number we replace tent function with empirical basis functions [. . .]"

This clarifies the end of the proposed parametrization.

Anh-Huy Phan: "- Page 140, Figure 8.1 Dashed and solid lines in Figure 8.1 and Figure 8.2 are not distinguished."

For each figure that contains dashed lines we clarified that "For each method, the mean execution time is specified with the dashed line in the legend". With this the visibility of dashed line becomes irrelevant.

Anh-Huy Phan: "Where is Figure 6.3a? There is a table with caption Figure 6.3 on page 114 but no figure."

Anh-Huy Phan: "Figures 6.3 6.4 and 6.5 are tables without caption. Check the captions or provide figures instead of tables."

Anh-Huy Phan: "Similar to Figures in Chapter 6, Figures 7.3, 7.4, 7.5 and 7.6 are tables. Move theses tables to the appropriate places. Captions of theses tables or figures should be self-contained"

We made sure tables are named appropriately in the text and the caption. We also supplied them with the caption that makes them self-contained:

"Comparison of three metrics for classical BPX preconditioners and optimized BPX preconditioners for selected equations. Spectral radius of error propagation matrix $\boldsymbol{I} - \boldsymbol{N}\boldsymbol{A}$ ($\boldsymbol{N}$ is BPX preconditioner $\boldsymbol{A}$ is a matrix of the original linear operators) denoted by ρ, condition number of preconditioned matrix κ, and number of iterations needed to drop and error by a factor of 10."

Anh-Huy Phan: "Move theses tables to the appropriate places."

We decided not to relocate tables because, currently, they are gathered on three separate pages in the middle of the section with the discussion of numerical results, which arguably is already an appropriate place.

Anh-Huy Phan: "- Part 2, page 101 "The rest part ... provides more examples on how machine learning can be used in numeral linear algebra"" What examples? please be more specific."

These examples were presented in the initial version. Now we reformulated the text as follows:

> The rest of the present part provides more examples of how machine learning can be used in numerical linear algebra. Namely, we start in Chapter 7 with a general explanation of unsupervised training suitable for the construction of solvers and preconditioners for numerical linear algebra. The whole endeavor is based on the introduction of appropriate stochastic losses and the minimization of them with gradient-based methods. That is, we apply classical approaches from machine learning to the construction of preconditioners. The models for preconditioners under the study are given by parametric families of generalized BPC multilevel preconditioners with overall architectures resembling U-Net. Next, in Chapter 8 we study the connection between multigrid and neural networks. In this chapter, we apply techniques from Chapter 7 to access the generalization capabilities of proposed approaches. That is, we show that one can train (perform optimization) on problems with a small number of unknowns and later use the same method on problems with a large number of unknowns without the loss in performance. Finally, in Chapter 9 we turns to the online setup. Namely, we show that it is possible to improve the iterative method on the flight using auxiliary information available as a byproduct of iterations. This is done with a help of k-armed bandits and Bayesian optimization.

We hope that the text clarifies what examples of machine learning approaches we have in mind.

> **Anh-Huy Phan: "- Numerical examples and the whole Chapter 6, it is not clear how Machine learning methods are applied to find the optimal preconditioner. Does the author mean that minimization of spectral conditional number uses ML methods?"**

The same fragment quoted above also answers this question. The techniques used to find optimal preconditioners are ML-based since we define stochastic loss, architecture (modified BPX preconditioner) and use stochastic optimization to find optimal parameters of the model (in this context, parameters that lead to the small condition number of the preconditioned linear system).

Besides, we provide additional comments on the matter in the introduction.

> **Anh-Huy Phan: "- Table 2.3: What are the numbers shown in each cell in Table 2.3?"**

We agree that this was not clarified enough. Section 3.6 starts with the description of measures we use to compare iterative methods. The one measure is the spectral radius of the error propagation matrix. The other is the effective amount of work that takes into account the number of floating point operations performed by solver or preconditioner.

To clarify that we provide the effective amount of work is the mentioned tables we introduced a phrase in brackets:

> Table 3.3, Table 3.4, Table 3.5, Table 3.6, Table 3.7 contain results (spectral radius $\rho \in [0, 1)$, which is located in the upper half of each cell, and the effective amount of work which is situated in the lower half of each cell) for multigrid used as a stand-alone solver (the left part of the table) as well as a preconditioner (the right part of the table). Table 3.8 contains results for stand-alone solvers.

> **Anh-Huy Phan: "- Confusing notation: $\left|\widetilde{R}\right|$ in Theorem 2.3.2 can be understood as determinant of the matrix $\widetilde{R}$."**

We introduced a footnote that clarifies this notation.

> **Anh-Huy Phan: "- Chapter 8 The over-relaxation parameter ω is not defined yet on page 132. What is the role of ω in the SOR? Section 8.1 should formulate the problem with SOR before introducing the task of finding accurate approximation to ω."**

SOR is now defined in Eq. (1.4) in Chapter 1.

> **Anh-Huy Phan: "- Part 1 misses a section for conclusion. What is the benefit of the probabilistic algorithm over deterministic algorithms? When can the proposed algorithm be applied?"**

Part I and Part II share the same conclusion. The end of the probabilistic algorithms is emphasized in a novel version of the introduction.

Anh-Huy Phan: "- Page 48: What is the problem of probabilistic reconstruction?"

In the current version the meaning of probabilstic reconstruction is also clarified in the introduction.

Anh-Huy Phan: "- Theorem 28. Does δ_{ij} denote the Kronecker delta? The notation is not defined in Chapter 2."

We clarified the notation in the footnote.

Anh-Huy Phan: "- Page 88 What does the symbol $\otimes$ stand for?"

There was a reference there [GN18, Definition 2.2.1] but we also clarified that $\otimes$ stands for the Kronecker product.