SOMIM:

An open-source program code for the numerical
Search for **O**ptimal **M**easurements by an **I**terative **M**ethod

Kean Loon Lee,[1,2] Jiangwei Shang,[1] Wee Kang Chua,[4] Shiang Yong Looi,[5] and Berthold-Georg Englert[1,3]

[1] *Centre for Quantum Technologies*
National University of Singapore
3 Science Drive 2, Singapore 117543, Singapore

[2] *Graduate School for Integrative Sciences and Engineering*
National University of Singapore
28 Medical Drive, Singapore 117456, Singapore

[3] *Department of Physics*
National University of Singapore
2 Science Drive 3, Singapore 117542, Singapore

[4] *Centre for Asset Management Research and Investments*
NUS Business School, BIZ 2 Building Level 5
1 Business Link, Singapore 117592, Singapore

[5] *Department of Physics*
Carnegie Mellon University, Pittsburgh, PA 15213, USA

Abstract

SOMIM is an open-source program code that implements a Search for Optimal Measurements by using an Iterative Method. For a given set of statistical operators, SOMIM finds the POVMs that maximize the accessed information, and thus determines the accessible information and one or all of the POVMs that retrieve it. The maximization procedure is a steepest-ascent method that follows the gradient in the POVM space, and also uses conjugate gradients for speed-up.

This manual is for version 2.0.

Contents

1 License Agreement

2 What can SOMIM be used for?

Consider the following quantum communication scenario. Alice sends quantum states ρ_j to Bob, who wishes to perform a generalized measurement, specified by a positive-operator-valued measure (POVM), on the state he receives. Generally speaking, owing to the nature of quantum mechanics, it is impossible for Bob to obtain full knowledge about the states which he is receiving. Instead, he has to choose his measurement judiciously from all measurements permitted by quantum mechanics.

The set of input states $\mathcal{E} = \{\rho_j \mid j = 1, 2, ..., J\}$ with $\rho_j \geq 0$ in which Bob receives the objects, gives the over-all statistical operator ρ in accordance with

$$\rho = \sum_{j=1}^{J} \rho_j \quad \text{with } \mathrm{tr}\{\rho\} = 1\,. \tag{1}$$

The POVM with outcomes $\Pi_k (k = 1, 2, ..., K)$ decomposes the identity,

$$\sum_{k=1}^{K} \Pi_k = 1 \quad \text{with } \Pi_k \geq 0\,. \tag{2}$$

Then, the joint probability to receive the jth state and get the kth outcome is

$$p_{jk} = \mathrm{tr}\{\rho_j \Pi_k\}\,, \quad \sum_{j,k} p_{jk} = 1\,. \tag{3}$$

Bob's figure of merit is the *mutual information*

$$I(\mathcal{E}; \Pi) = \sum_{j=1}^{J} \sum_{k=1}^{K} p_{jk} \log_2 \frac{p_{jk}}{p_{j\cdot} p_{\cdot k}}\,, \tag{4}$$

where $p_{j\cdot}$ and $p_{\cdot k}$ are the marginal probabilities,

$$p_{j\cdot} = \sum_{k} p_{jk} = \mathrm{tr}\{\rho_j\}\,, \quad p_{\cdot k} = \sum_{j} p_{jk} = \mathrm{tr}\{\rho \Pi_k\}\,. \tag{5}$$

As stated, the ρ_js are normalized such that their traces equal the probabilities of receiving them.

Accessible Information(AI): The accessible information I_{acc} is the maximum of the mutual information $I(\mathcal{E}; \Pi)$ for all possible POVMs, that is

$$I_{\mathrm{acc}}(\mathcal{E}) = \max_{\Pi} I(\mathcal{E}; \Pi)\,. \tag{6}$$

For more about AI and related quantities, see Refs. [1] and [2].

Given a set $\mathcal{E}$ of statistical operators ρ_j, SOMIM calculates the accessible information associated with these statistical operators, and finds the POVM that retrieves the AI, or rather, it finds one of the optimal POVMs, as the optimum need not be unique. Repeated runs of SOMIM for the same input data but different random seeds may yield alternative optimal POVMs.

The calculation is performed using a combination of the steepest-ascent method (see Ref. [1] and Section 11.5 in Ref. [2]) and the conjugate-gradients (CG) method [3]. The percentage chance to calculate with one method or the other can be specified by the user (see Section 4 below). The implementation in SOMIM also makes use of the golden-section search method.

3 Download and Compile

The complete set of files, including this manual, are available at the SOMIM site: `http://www.quantumlah.org/publications/software/SOMIM/`. Download `http://www.quantumlah.org/publications/software/SOMIM/all.tar.gz`, if you want to have the complete collection of files. Just this manual is fetched from `http://www.quantumlah.org/publications/software/SOMIM/Manual.pdf`. The Windows executable file for SOMIM can be downloaded from `http://www.quantumlah.org/publications/software/SOMIM/somim.tar.gz`. If you intend to modify the code, you can download the source files from `http://www.quantumlah.org/publications/software/SOMIM/source.tar.gz`.

The program is written in C++ and the graphic user interface (GUI) is implemented using wxWidgets (`http://www.wxwidgets.org/`). Here are the instructions for compiling SOMIM:

1. Install wxWidgets from `http://www.wxwidgets.org/downloads/`.
2. If you are working in Windows, you need to install MinGW (`http://www.mingw.org/download.shtml`) and MYSY (`http://www.mingw.org/msys.shtml`) as well.
3. When wxWidgets and MinGW are configured, you can compile SOMIM by executing **g++ MI.cpp ‘wx-config –libs‘ ‘wx-config –cxxflags‘ -o YourProgramName** in MSYS shell.
4. If you face problems running the program in a Linux environment, try **export LD_LIBRARY_PATH=/usr/local/lib**.
5. The executable file is compiled under Windows XP Service Pack 3, with wxWidgets 2.8.10, MinGW 5.1.6 and MSYS 1.0.11.

4 How to use the program

The GUI of SOMIM is shown in Fig. 1. In the first box labeled as “Parameter Settings”, J is the number of statistical operators to be input. The current maximum possible value is $J = 30$. Parameter K is the initial number of POVM outcomes, with the largest possible value being $K = 30$. The third field is the dimension N of the statistical operators which has $N = 30$ as the highest possible value; that is, the J different statistical operators are represented by $N \times N$ matrices. You can change the maximum values of J, K and N by modifying the source code. The fourth field is the chance

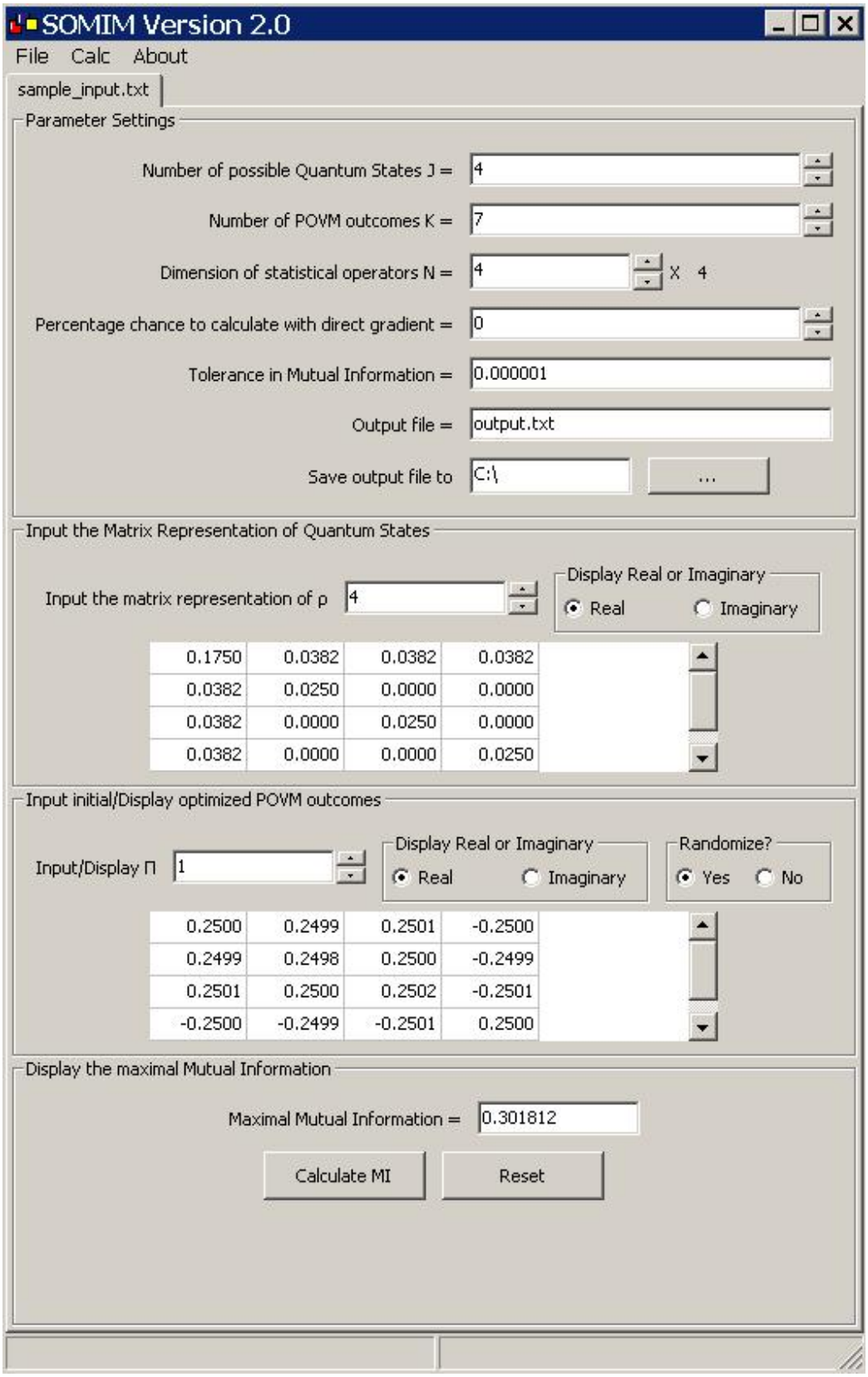

Figure 1: The graphical user interface (GUI) of the program.

of using the steepest-ascent gradient method to perform maximization in an iteration; this parameter controls the relative frequency of using the direct or the conjugate gradient. The fifth field gives the tolerance in the accessible information, the stopping criteria for the computation; the calculation stops when the difference in accessible information between the current iteration and the previous iteration is less than half of the sum multiplied by the tolerance plus the machine_epsilon ϵ_m (also termed as the machine accuracy, typical value for double precision is around 1.6×10^{-16}), i.e. when $2.0 \times (\text{current} - \text{previous}) \leq \text{tolerance} \times (\text{current} + \text{previous}) + \epsilon_m$. The sixth field is the name of the output file. By default, the output file will be located at hard disk C. You can change the output directory by clicking the ellipsis button "..." and choose your preferred location.

The next two boxes display the statistical operators $\{\rho_j\}_{j=1,\ldots,J}$ and the outcomes $\{\Pi_k\}_{k=1,\ldots,K}$ of the POVM. You will be given the option to either manually input the initial POVMs by clicking the "No" button or you may leave the program to choose a set of randomized initial POVMs. After the calculation, the optimal POVMs will be displayed. The spin buttons are used to switch between the various ρ_js and Π_ks, respectively, while the small box beside the spin button is used to choose to display the real or imaginary part of the chosen ρ_j or Π_k.

The maximum accessible information for the given set $\{\rho_j\}$ will be displayed in the last box after the "Calculate MI" button is pressed. All values will be reset to default when the "Reset" button is pressed.

Important note: The matrices for the ρ_js must have the correct dimension; they have to be hermitian with nonnegative eigenvalues; their traces are their statistical weights, which must add to unity: $\sum_j \text{tr}\{\rho_j\} = 1$.

5 How to import data

Data can be imported into SOMIM using a text file that is possibly generated by another program. An example is shown in Fig. 2. When importing, the numbers after the equal signs will be read into SOMIM. The first line gives the dimension N of the statistical operators. The second line gives the number J of statistical operators while the third line gives the number K of outcomes of the POVMs that SOMIM should start calculating with.

The subsequent lines give the input matrices for the statistical operators. Each line will give only one operator. For an operator represented in matrix form as

$$\begin{pmatrix} 0.1 & 0.3 + 0.5i \\ 0.3 - 0.5i & 0.6 \end{pmatrix}, \tag{7}$$

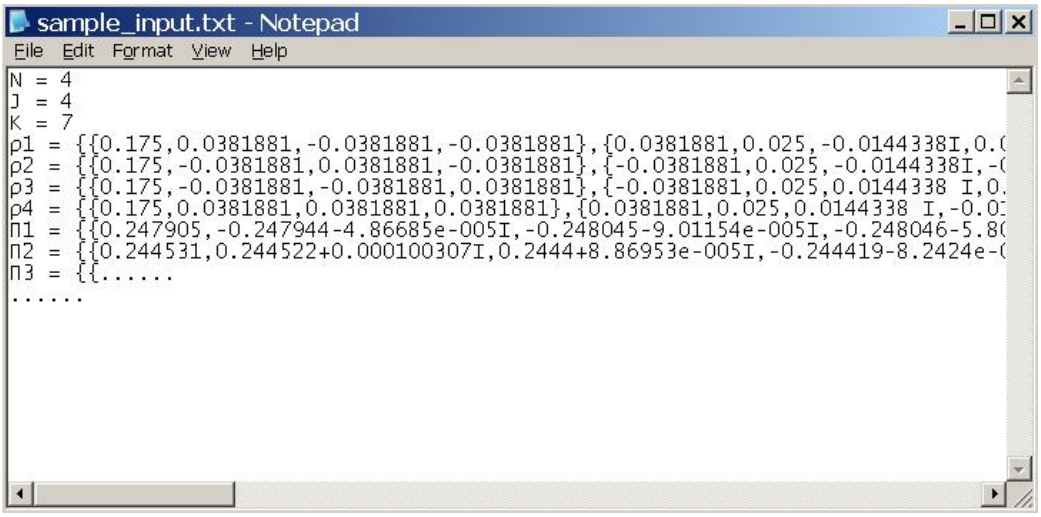

Figure 2: Example of an import file.

the input data should be formatted as $\{\{0.1, 0.3 + 0.5\mathtt{I}\}, \{0.3 - 0.5\mathtt{I}, 0.6\}\}$.

In most cases, randomly generated initial POVMs will lead to the maximum mutual information and it is safe to click the "Yes" button. However, there exist some pathological examples for which randomly generated initial POVMs almost never achieve the maximum and we need to manually set the initial POVMs. Therefore, additional matrices for the POVM outcomes, formatted in the same way as the statistical operators, could be entered as well.

Complex numbers are entered as RealPart + ImaginaryPart `I`, as illustrated by $-3.1 - 4.5\mathtt{I}$. Please note that the complex unit i must be entered in upper case `I` and it must be at the end of the entry.

6 Meaning of output data

A typical output file looks like Fig. 3. The first six lines give the following information: the number J of statistical operators, the initial number K of POVM outcomes, the dimension N of the statistical operators, the probability of maximizing information using steepest gradient method compared to conjugate gradient method, the tolerance in the calculated mutual information, and the seed for the random number generator.

The next block of lines gives the mutual information at the end of each iteration. In the example shown in Fig. 3, altogether 26 iterations have been performed with the final accessible information being $\mathrm{AI} = 0.301\,812$.

The subsequent two blocks of lines give the J statistical operators and

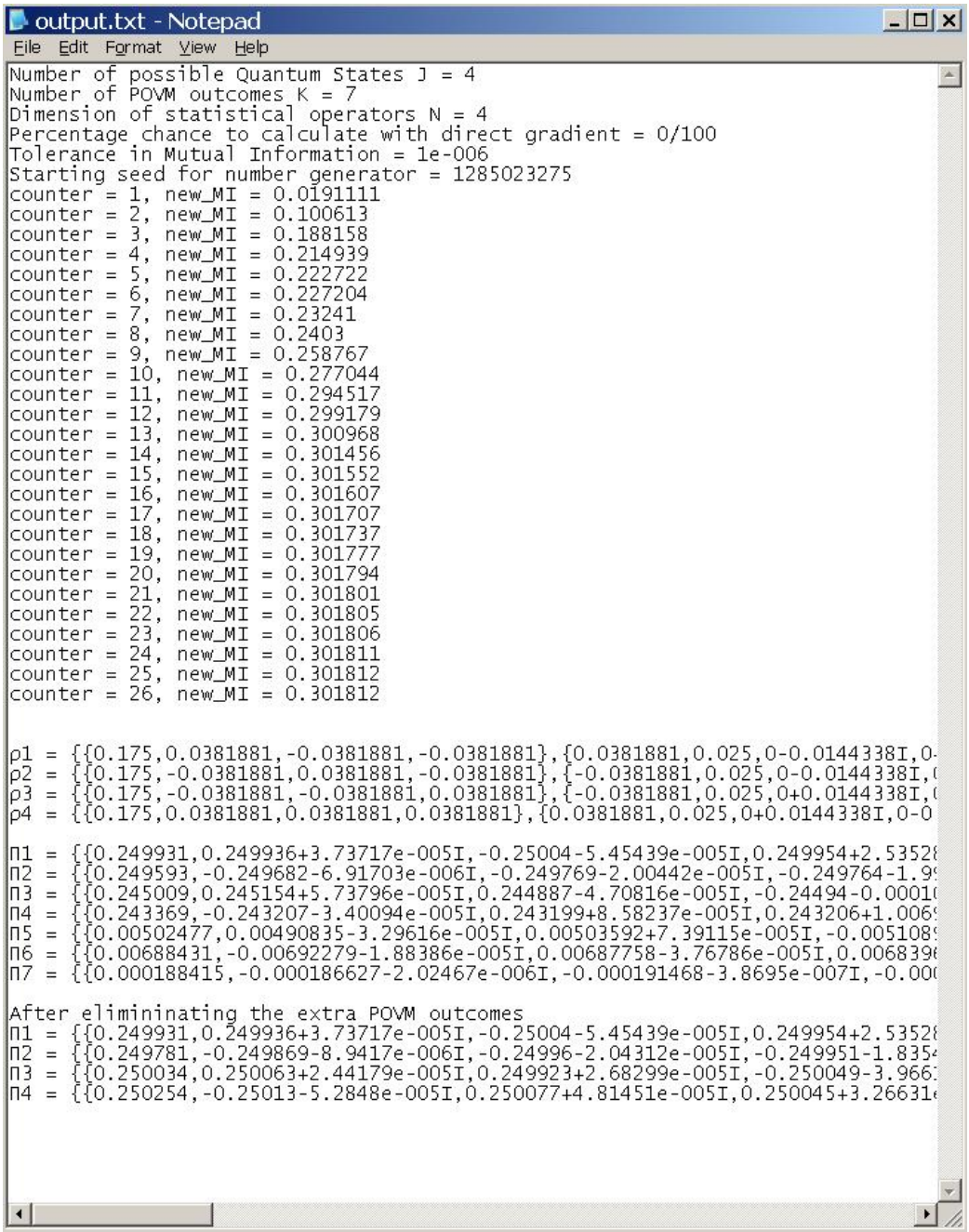

Figure 3: Example of output file.

the K outcomes of the optimal POVMs that correspond to the accessible information calculated in the final round of iteration. Each statistical operator/POVM outcome is given in a single line in matrix form, as explained in Section 5.

Among the K outcomes, if any two outcomes, say Π_{k_1} and Π_{k_2}, give equivalent probabilities, i.e. $p_{jk_1}p_{\cdot k_2} = p_{\cdot k_1}p_{jk_2}$ for all j, then these two POVM outcomes are replaced with one new POVM outcome, $\Pi_{k_1} + \Pi_{k_2}$, such that the new optimal POVM contains only $K - 1$ outcomes. The last block of data in the output file gives the POVM after this elimination process, i.e. the POVM is the optimal POVM with the least number of outcomes.

Caution: As is the case for all steepest-ascent methods, there is the possibility of convergence towards a local, rather than a global, maximum. There is no absolute protection against this danger, but in practice one can fight it efficiently by running the program many times for comparison, with different seeds. It also helps to start with a rather large K value.

7 Contact information

Please send your comments, suggestions, or bug reports to the following email account: `somim@quantumlah.org`

8 Acknowledgements

We acknowledge many valuable discussions with J. Řeháček and T. Karpiuk. This work was supported by NUS Grant WBS: R-144-000-109-112. Centre for Quantum Technologies (CQT) is a Research Centre of Excellence funded by Ministry of Education and National Research Foundation of Singapore.

References

[1] J. Řeháček, B.-G. Englert, and D. Kaszlikowski, *Iterative procedure for computing accessible information in quantum communication*, Phys. Rev. A **71**, 054303 (2005); eprint available at `http://arxiv.org/pdf/quant-ph/0408134`.

[2] J. Suzuki, S. M. Assad, and B.-G. Englert, "Accessible information about quantum states: An open optimization problem", Chapter 11 in *Mathematics of Quantum Computation and Quantum Technology*, edited by G. Chen, S. J. Lomonaco, and L. Kauffman (Chapman & Hall/CRC, Boca

Raton 2007), pp. 309–348; also available at `http://physics.nus.edu.sg/~phyebg/Papers/135.pdf`.

[3] W. H. Press, B. P. Flannery, S. A. Teukolsky, W. T. Vetterling, "Minimization or Maximazation of Functions", Chapter 10 in *Numerical Recipes in C: The Art of Scientific Computing*, (Cambridge University Press, 2nd edition 1992), pp. 394–455.