On Super Numerical Simulation

Hitoshi IMAI (今井 仁司), Toshiki TAKEUCHI (竹内 敏己),
Hideo SAKAGUCHI (坂口 秀雄) and Tetsuya HISHINUMA (菱沼 哲也)†

Faculty of Engineering, University of Tokushima, Tokushima 770-8506, Japan.
(徳島大学工学部)

†Graduate School of Engineering, University of Tokushima, Tokushima 770-8506, Japan.
(徳島大学大学院工学研究科)

1 Introduction

Recently, we can easily use powerful computers. This year high-end PCs with 16GB memory(chipset:E7500) or 64GB memory(chipset:GC-HE) are available. In next thirty years computers will have more a million times capability. What will be the future numerical simulation ? Taking into consideration such development of computers, we propose a new terminology: super numerical simulation. It means numerical simulation which highly overstrides usual one. The similar terminology is seen in super computer. In what sense super numarical simulation should be super ? We propose five points: ultimate precision(ultimate reliability), ultimate visualization, ultimate fast-computing, ultimate applicability, and hardware implementation(cf. `http://isam.pm.tokushima-u.ac.jp/~imai/`). Hardware implementation means for example the flow simulator where the nural network simulating flow is implemented as circuitries. Among these five points ultimate precision, ultimate visualization and ultimate applicability are shown in the paper.

2 Ultimate precision

In numerical simulation reliability of numerical solutions is checked by comparing those obtained in different precision. If numerical solutions converge, then it is recognized reliable numerical solutions are obtained. However, sometimes numerical solutions do not converge. This is because they are not prepared abundantly. Usual numerical methods sometimes do not give abundant series of numerical solutions in different accuracy.

On the other hand, numerical simulation sometimes fails due to instability caused by numerical errors. Especially, the rounding error plays a fatal role. If the truncation error is as small as the rounding error, the rounding error spoils mathematical validity of the numerical method.

To overcome these difficulties we proposed IPNS(Infinite-Precision Numerical Simulation) [6]. IPNS consists of the arbitrary order approximation method and the multiple precision arithmetic. The former is used for arbitrary reduction of the truncation error. The latter is used for arbitrary reduction of the rounding error[9]. As the arbitrary order approximation method, spectral methods are available from the pratical view point[1]. Among spectral methods SCM (Spectral Collocation Method) is very useful. This is because the way of its application is similar as FDM. It is easily applicable to nonlinear problems including free boundary problems[11]. The multiple precision arithmetic is easily carried out by using free subroutine libraries on the net, e.g. `http://www.lmu.edu/acad/personal/faculty/dmsmith2/FMLIB.html`[10].

In the paper, SCM using Chebyshev Polynomials and Chebyshev-Gauss-Lobatto collocation points is used. A function $u(x)$ in $[-1,1]$ is approximated by the N-th order Chebyshev Polynomials as follows:

$$u(x) = \sum_{k=0}^{N} \tilde{u}_k \, T_k(x), \quad T_k(x) = \cos(k \ \arccos x). \tag{1}$$

There is an inversion formula

$$u_j = \sum_{k=0}^{N} \tilde{u}_k T_k(x_j), \qquad \tilde{u}_k = \frac{2}{N\bar{c}_k} \sum_{j=0}^{N} \frac{1}{\bar{c}_j} u_j T_k(x_j) \tag{2}$$

where

$$\bar{c}_j = \begin{cases} 2, & j = 0, N, \\ 1, & \text{otherwise}, \end{cases} \qquad x_j = \cos\frac{j\pi}{N}, \ j = 0, 1, \cdots, N. \tag{3}$$

$\{x_j\}$ are called Chebyshev-Gauss-Lobatto collocation points. Derivatives at the collocation points are easily computed from $\{u_j\}$. In SCM it is easy to increase the order of the approximation by increasing the number of collocation points. This feature is quite remarkable and different from other discretization methods. Of course SCM is applicable both in space and in time.

To see IPNS is realized, the following simple one-dimensional boundary value problem is solved.

Problem 1. Find $u(x)$ s.t.

$$u_{xx} = -\frac{\pi^2}{16}\sin\frac{(x+1)\pi}{4}, \qquad -1 < x < 1, \tag{4}$$

$$u(-1) = 0, \qquad u_x(1) = 0. \tag{5}$$

Remark 1. *The exact solution to* Problem 1 *is* $u(x) = \sin\dfrac{(x+1)\pi}{4}$.

Numerical results are shown in Table 1. Here $(N+1)$ Chebyshev-Gauss-Lobatto points are used. Extremely high accuracy is obseved. It should be remarked that even to such a simple problem IPNS needs huge memory spaces and too much computational time now.

Table 1. Maximum error for Problem 1.(2500 digit numbers)

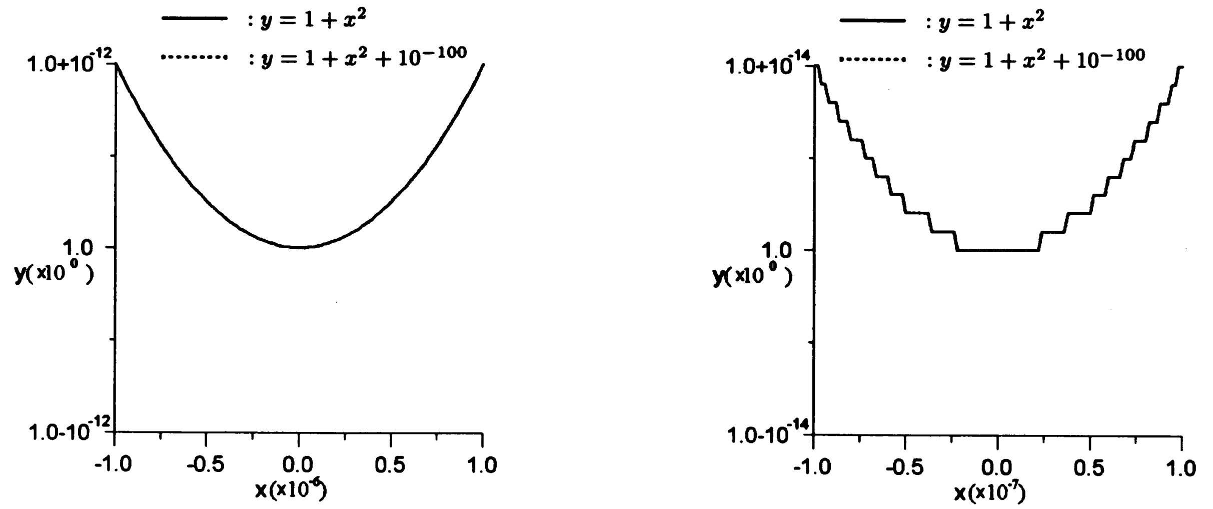
N	Maximum error	N	Maximum error	N	Maximum error
10	4.88×10^{-11}	80	1.03×10^{-151}	200	1.82×10^{-456}
20	6.64×10^{-27}	90	4.32×10^{-175}	300	1.20×10^{-736}
30	5.39×10^{-45}	100	6.01×10^{-199}	400	1.46×10^{-1031}
40	1.54×10^{-64}	120	6.38×10^{-248}	500	1.95×10^{-1337}
50	3.62×10^{-85}	140	2.42×10^{-298}	600	4.78×10^{-1652}
60	1.16×10^{-106}	160	5.25×10^{-350}	700	6.36×10^{-1974}
70	7.02×10^{-129}	180	9.38×10^{-403}	800	5.09×10^{-2302}

3 Ultimate visualization

In numerical simulation visualization is very important. However, in super numerical simulation normal visualization may not be available. For example IPNS gives numerical data in long digits. There is no visualization software which can deal with such data. Among highly accurate results by IPNS difference is extremely small. There is no visualization software which can show such small difference.

Here a new visualization for IPNS is presented. Multiple precision arithmetic is used in pre-processing which is programed in FORTRAN. Then suitable data whose regulation is given by the visualization software are created and delivered to the visualization software. Some examples obtained by the new visualization are shown below. It is interesting that basic facts in numerical analysis are visualized.

Fig. 1 shows two graphs of $y = 1 + x^2$ and $y = 1 + x^2 + 10^{-100}$ in different magnification. Graphs are generated by connecting $(x_j, y(x_j))_{j=0}^{100}$ in a straight line. $\{x_j\}$ are equally distributed. Data are prepared in double precision. Two graphs are not distinguished. This is because 10^{-100} is neglected compared with $1 + x^2$. Fig. 1(b) shows machine epsilon or distribution of floating-point numbers.

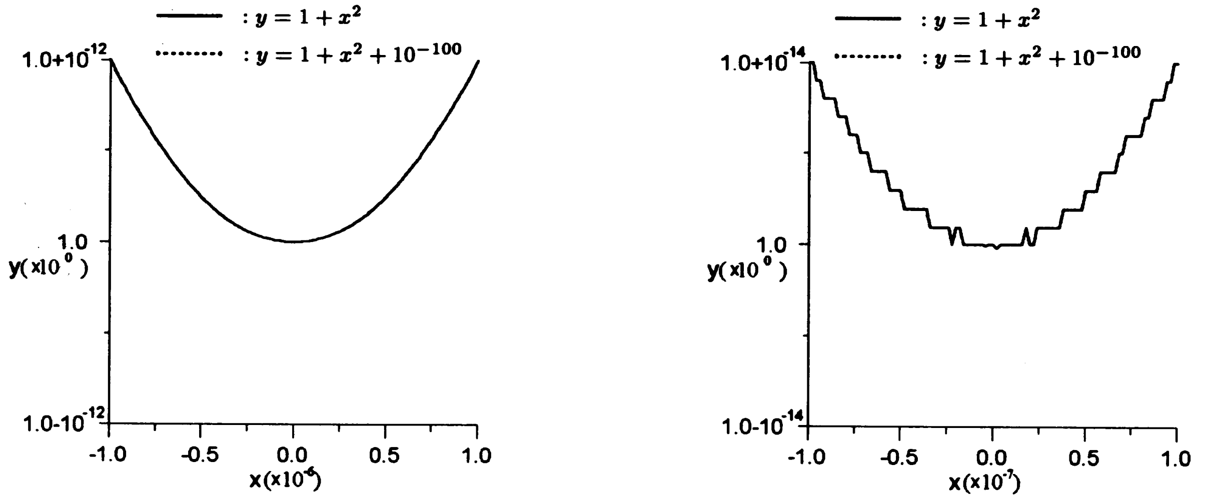


(a) Magnification: 10^6 in x, 10^{12} in y.

(b) Magnification: 10^7 in x, 10^{14} in y.

Fig. 1. Two graphs of $y = 1 + x^2$ and $y = 1 + x^2 + 10^{-100}$ (Double precision).

Fig. 2 shows two graphs of $y = 1 + x^2$ and $y = 1 + x^2 + 10^{-100}$ in different magnification. Graphs are generated by connecting $(x_j, \tilde{y}(x_j))_{j=0}^{100}$ in a straight line. $\{x_j\}$ are equally distributed. $\tilde{y}(x)$ is the interpolated function by SCM with $(x_i, y(x_i))_{i=0}^{10}$, $\{x_i\}$: Chebyshev-Gauss-Lobatto Collocation points in $[-1,\ 1]$. Data are prepared in double precision. Two graphs are not distinguished. This is because 10^{-100} is neglected compared with $1 + x^2$. In Fig. 2(b) oscillation to be induced from the rounding error is seen.

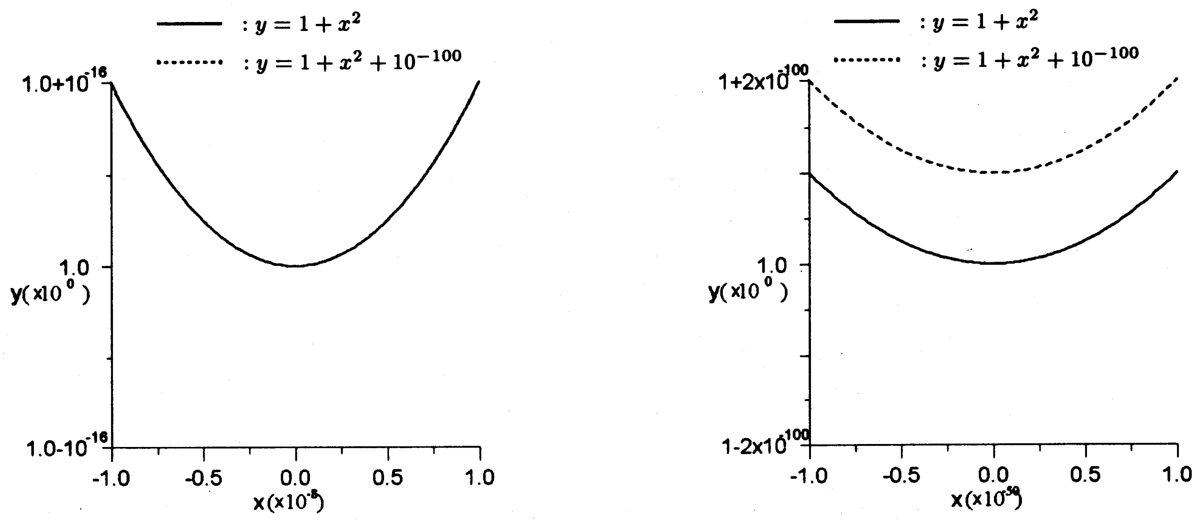


(a) Magnification: 10^6 in x, 10^{12} in y.

(b) Magnification: 10^7 in x, 10^{14} in y.

Fig. 2. Two graphs of interpolated functions created by SCM with data from $y = 1 + x^2$ and $y = 1 + x^2 + 10^{-100}$ (Double precision).

Fig. 3 shows the new visualization for two graphs of $y = 1 + x^2$ and $y = 1 + x^2 + 10^{-100}$ in different magnification. Graphs are generated by connecting $(x_j, \tilde{y}(x_j))_{j=0}^{100}$ in a straight line. $\{x_j\}$ are equally distributed. $\tilde{y}(x)$ is the interpolated function by SCM with $(x_i, y(x_i))_{i=0}^{10}$, $\{x_i\}$: Chebyshev-Gauss-Lobatto Collocation points in $[-1,\ 1]$. Data are prepared in 200 digit numbers. In Fig. 3(b) two graphs are very natural and distinguished from each other.



(a) Magnification: 10^8 in x, 10^{16} in y.

(b) Magnification: 10^{50} in x, 10^{100} in y.

Fig. 3. Two graphs of interpolated functions created by SCM with data from $y = 1 + x^2$ and $y = 1 + x^2 + 10^{-100}$ (200 digits).

4 Ultimate applicability

Inverse problems often arise from practical problems. It is well-known they are very difficult to be solved[3]. Their direct simulation has been a taboo due to easy corruption by the strong oscillation phenomenon. To avoid this oscillation phenomenon some additional methods are usually used together. To say roughly, these are regularization[12], the method of least squares and AI. Restriction of the dimension of the solution space is a sort of regularization. These methods are very useful but unfortunately not absolute. Especially, in numerical simulation the rounding error fails their theoretical usefulness.

On the other hand, direct simulation by IPNS was applied to several inverse problems governed by PDE systems[4, 5, 7, 8]. Numerical results were very satisfactory. This means numerical errors are extremely small, so they do not induce the oscillation phenomenon. In the paper the following integral equation of the first kind with an analytic kernel and an analytic solution is solved by IPNS without any additional methods like regularization[2, 5].

Problem 2. Find $u(y)$ s.t.

$$\int_{-1}^{1} e^{xy}u(y)dy = f(x), \quad f(x) = \frac{(2x-1)e^{x}+e^{-x}}{2x^2}, \quad f(0) = 1. \tag{6}$$

Remark 2. *The exact solution to* Problem 2 *is* $u(y) = \frac{y+1}{2}$.

SCM is applied to the integrand as follows:

$$e^{xy}u(y) \cong \sum_{k=0}^{N} \tilde{g}_k(x)T_k(y). \tag{7}$$

From the inversion formula

$$\tilde{g}_k(x) \cong \frac{2}{Nc_k}\sum_{j=0}^{N}\frac{1}{c_j}e^{xy_j}u_jT_k(y_j), \quad k = 0,\ 1, \cdots,\ N, \tag{8}$$

where

$$y_j = \cos\frac{j\pi}{N}, \quad u_j = u(y_j), \quad j = 0,\ 1, \cdots,\ N. \tag{9}$$

Thus

$$e^{xy}u(y) \cong \sum_{k=0}^{N}\frac{2}{Nc_k}\sum_{j=0}^{N}\frac{1}{c_j}e^{xy_j}u_jT_k(y_j)T_k(y). \tag{10}$$

Then

$$\begin{aligned}\int_{-1}^{1} e^{xy}u(y)dy &\cong \int_{-1}^{1}\left\{\sum_{k=0}^{N}\frac{2}{Nc_k}\sum_{j=0}^{N}\frac{1}{c_j}e^{xy_j}u_jT_k(y_j)T_k(y)\right\}dy \\ &= \frac{2}{N}\sum_{k=0}^{N}\sum_{j=0}^{N}\frac{1}{c_kc_j}e^{xy_j}u_jT_k(y_j)\int_{-1}^{1}T_k(y)dy \\ &= \frac{2}{N}\sum_{\substack{k=0\\k\neq 1}}^{N}\sum_{j=0}^{N}\frac{1}{c_kc_j}e^{xy_j}u_j\cos\frac{jk\pi}{N}\cdot\frac{1+(-1)^k}{1-k^2}.\end{aligned} \tag{11}$$

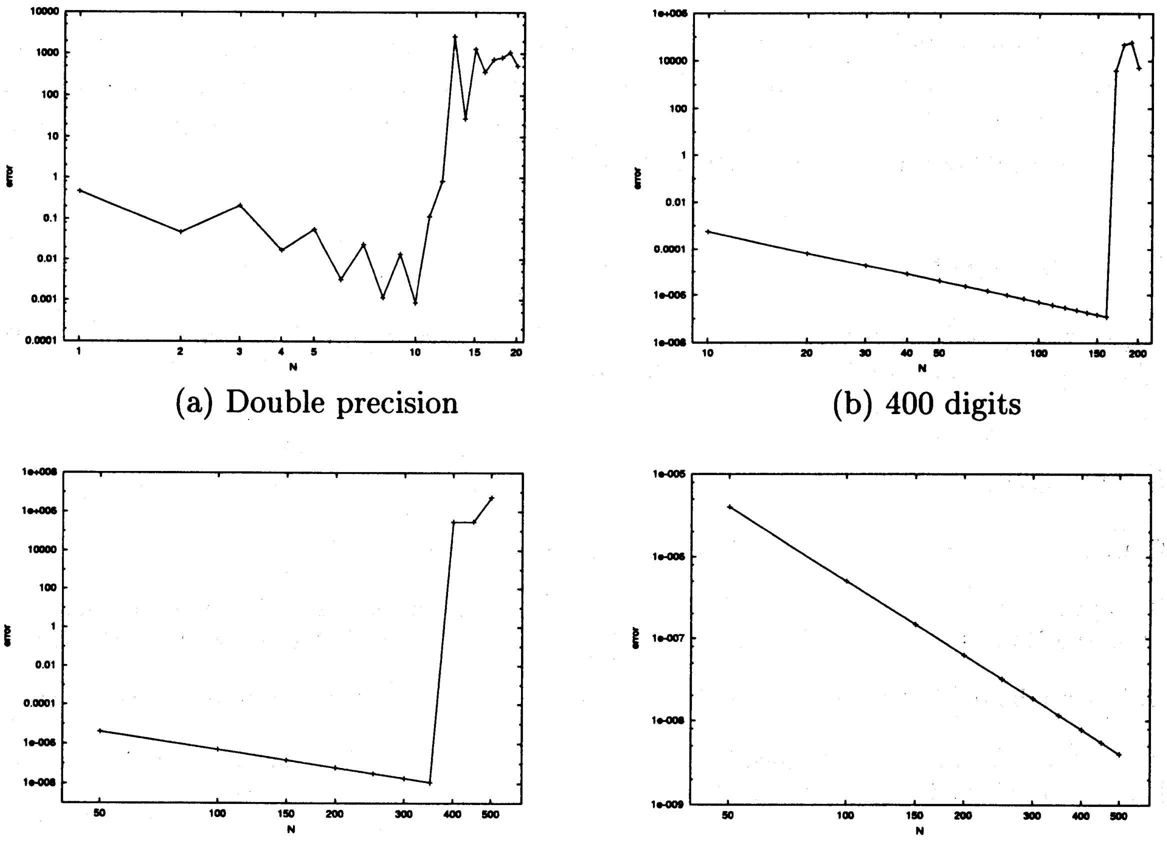
We choose properly points $\{x_l\}$, $l = 0,\ 1, \cdots,\ N$ on which the integral equation is satisfied. Set $f_l = f(x_l)$, then we have the following linear system:

$$\sum_{j=0}^{N} a_{lj}u_j = \frac{N}{2}f_l, \quad l = 0,\ 1, \cdots,\ N, \tag{12}$$

$$a_{lj} = \sum_{\substack{k=0 \\ k\neq 1}}^{N} \frac{1}{c_k c_j} e^{x_l y_j} \cos\frac{jk\pi}{N} \cdot \frac{1+(-1)^k}{1-k^2}. \tag{13}$$

After solving this linear system, $u(y)$ is reconstructed as follows :

$$u(y) = \sum_{k=0}^{N}\sum_{j=0}^{N} \frac{2}{Nc_k c_j} u_j T_k(y_j) T_k(y). \tag{14}$$



(a) Double precision

(b) 400 digits

(c) 1000 digits

(d) 2000 digits

Fig. 4. Behavior of maximum errors for Problem 2.

Fig. 4 shows errors for Problem 2 with C-G-L collocation points : $x_l = \cos\frac{l\pi}{N}$, $l =$

$0, 1, \cdots, N$. Here,

$$\text{error} = \max_{0 \leqq j \leqq N} |u_N(y_j) - u(y_j)|, \quad y_j = \cos\frac{j\pi}{N}, \quad j = 0, 1, \cdots, N \tag{15}$$

$u(y)$ is the exact solution, and $u_N(y)$ is the right-hand side of Eq. (14). If rounding error is not small enough, error grows explosively before obtaining good results. This shows the linear system (12) is very ill-conditioned. At the same time, if rounding error is small enough, error reduces successively as N becomes large. In Fig. 4(d) the regression line by the method of least squares is $\log(\text{error}) = -3.00 * \log N - 0.686$ with the correlation coefficient $\rho = -1.00$. This means IPNS works well.

5 Conclusion

In the paper a new terminology: super numerical simulation is proposed. Several examples of super numerical simulation are also presented. As computers become more powerful today's super numerical simulation becomes normal. However, super numerical simulation exists at any time, as super computer exists at any time.

Acknowledgements

This work is partially supported by Grants-in-Aids for Scientific Research (No. 13640119), from the Japan Society of Promotion of Science.

References

[1] C. Canuto et al., Spectral Methods in Fluid Dynamics, Springer, 1998.

[2] H. Fujiwara and Y. Iso, Numerical Challenge to Ill-posed Problems by Fast Multiple-Precision System, Proc. the 50th Japan National Congress on Theoretical and Applied Mechanics, 2001, 419-424.

[3] G. Hammerlin et al., Improperly Posed Problems and Their Numerical Treatment, Birkhäuser, 1983.

[4] H. Imai and T. Takeuchi, Application of the Infinite-Precision Numerical Simulation to an Inverse Problem, NIFS-PROC-**40**(1999), 38-47.

[5] H. Imai and T. Takeuchi, Some Advanced Applications of the Spectral Collocation Method, GAKUTO Int. Ser. Math. Sci. Appl., **17**(2001), 323-335.

[6] H. Imai, T. Takeuchi and M. Kushida, On Numerical Simulation of Partial Differential Equations in Infinite Precision, Advances in Mathematical Sciences and Applications, **9**(2)(1999), 1007-1016.

[7] H. Imai, T. Takeuchi, M. Nakamura and N. Ishimura, A DIRECT APPROACH TO AN INVERSE PROBLEM, GAKUTO Int. Ser. Math. Sci. Appl., **12**(1999), 223-232.

[8] H. Imai, T. Takeuchi and H. Sakaguchi, Infinite Precision Numerical Simulation for PDE Systems and Its Applications, RIMS Kokyuroku, Kyoto University, **1147**(2000), 42-50.

[9] D. E. Knuth, The Art of Computer Programming, Addison-Wesley, 1981.

[10] D. M. Smith, A FORTRAN Package For Floating-Point Multiple-Precision Arithmetic, Tran on Mathematical Software, **17**(1991), 273-283.

[11] Tarmizi, T. Takeuchi, H. Imai and M. Kushida, Numerical simulation of one-dimensional free b problems in infinite precision, Advances in Mathematical Sciences and Applications, **10**(2)(20C 672.

[12] A. N. Tikhonov and V. Y. Arsenin, Solution of Ill-Posed Problems, John Wiley & Sons, 1977.