

We are IntechOpen, the world's leading publisher of Open Access books Built by scientists, for scientists

6,900

Open access books available

185,000

International authors and editors

200M

Downloads

Our authors are among the

154

Countries delivered to

TOP 1%

most cited scientists

12.2%

Contributors from top 500 universities



WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index
in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

Numbers displayed above are based on latest data collected.
For more information visit www.intechopen.com



Hybrid Type Method of Numerical Solution Integral Equations and its Applications

D.G.Arsenjev¹, V.M.Ivanov¹ and N.A. Berkovskiy²

¹Professor, St.Petersburg State Polytechnical University,

²assoc. prof., St.Petersburg State Polytechnical University,
Russia

1. Introduction

The goal of current research is analysis of the effectiveness of application of semi-statistical method to the issues, which come up in computational and engineering practice.

The main advantages of this method are the possibility to optimize nodes on the domain of integration (which makes the work of calculator a lot easier), and also to control the accuracy of computations with the help of sample variance. Besides this, to improve the accuracy you can calculate an average solution by statistically independent estimations, acquired at a small number of integration nodes. A less attractive feature of this method is a low rate of convergence, which is relevant to all statistic methods.

The reason for this research has become a quite successful application of semi-statistical method to the test tasks [1, 2, 3]. The problem of plane lattice cascade flow with ideal incompressible fluid was chosen for simulation. With the help of semi-statistical method quite precise results have been achieved with the lattices, parameters of which were taken from engineering practice. These results were compared with the solutions from other computational methods.

Attempts to accelerate the rate of convergence brought to modernization of the method (deleting of spikes in average sum). As a result, in all the considered issues solutions with satisfactory precision were received, adaptive algorithm of lattice optimization was “putting” the nodes on the domain of integration in accordance with the theoretical considerations. However, in some cases the solution made by the semi-statistical method turned out to be longer, than when using deterministic methods, which is caused by imperfection of software implementation and also with the necessity to look for the new ways to accelerate rate of convergence for semi-statistical method, in particular, optimization mechanism.

2. Short scheme of semi-statical method

With semi-statistical method integral equations of the following kind can be solved:

$$\varphi(x) - \lambda \int_s K(x, y) \varphi(y) dy = f(x) \quad (1)$$

where S – smooth $(m-1)$ -dimensional surface in R^m ,

$$x \in S, y \in S, \lambda \in R,$$

K - kernel of equation, f - known function, φ - unknown function. This algorithm is described in detail in [1]. Let us shortly take a look at the scheme of its application in general case.

- With the help of random number generator on the surface S . N - number of independent points x_1, x_2, \dots, x_N (vectors) is created with a arbitrary probability density $p(x)$ (random integration grid).
- These points are placed one by one in (1), N equations of the kind given below are received:

$$\varphi(x_i) - \lambda \int_S K(x_i, y) \varphi(y) dy = f(x_i), \quad (i=1, 2, \dots, N) \quad (2)$$

- Integrals in (2) are substituted with the sums by the Monte-Carlo method [1, 2] and a system of linear algebraic equations appears

$$\varphi_i - \frac{\lambda}{N-1} \sum_{\substack{j=1 \\ j \neq i}}^N \frac{K(x_i, x_j)}{p(x_j)} \varphi_j = f(x_i) \quad (3)$$

Here $\{\varphi_i\}$ ($i=1, 2, \dots, N$) vector of unknown variables of the system (3). Having solved (3), φ_i take for approximated value $\varphi(x_i)$ of the solution of integral equation (1) correspondingly. Approximated value $\varphi(x) \forall x \in S$ is defined by “retracing” with the following algorithm:

$$\varphi(x) \approx f(x) + \frac{\lambda}{N} \sum_{i=1}^N \frac{K(x, x_i)}{p(x_i)} \varphi_i \quad (4)$$

The bigger is N the more precisely integrals in (2) are approximated by the finite sum in (3), which means that we can suppose, that by incrementing value of N is possible to minimize calculating error of approximation of φ_i from (3) and $\varphi(x)$ from (4) in a way that computation precision requires. As the number of thrown points is sometimes not enough to reach predefined precision (this number can't be enlarged infinitively as there is no possibility to solve to large equation systems), it is recommended to compute m times with N of thrown points, and then to average the results. This technique gives almost the same result if we would throw $N \times m$ points, because random points in different iterations are statistically independent.

- You can get an estimated value of optimal density of integration nodes by formula [1] by means of approximated solution $\varphi(x)$.

$$p_{optm}(y) = C(N-1) \frac{\sum_{j=1}^N \frac{(K(x_j, y) \varphi(y))^2}{p(y)}}{\sum_{j=1}^N \left\{ \frac{K(x_j, y) \varphi(y)}{p(x_j)} \sum_{\substack{i=1 \\ i \neq j}}^N K(x_j, x_i) \varphi(x_i) \right\}} \quad (5)$$

Having generated the points with the density $p_{opt}(y)$, received from (5) by approximated values φ_i , we can get a more precise solution of the equation (1). After that with this equation and by means of (5) we can calculate again (more precise) value of optimal density. The process can be repeated till the density stops changing. This is the main sense of adaptive algorithm of choosing an optimal density.

3. Statement of the problem of blade cascade flow

A plane lattice with the increment t (Fig. 1) is given, on which from the infinity under the angle β_1 a potential flow of ideal fluid is leaking, coming out from a lattice under the angle β_2 . The task is to find an absolute value of a normed speed of the flow on the edging of the profiles.

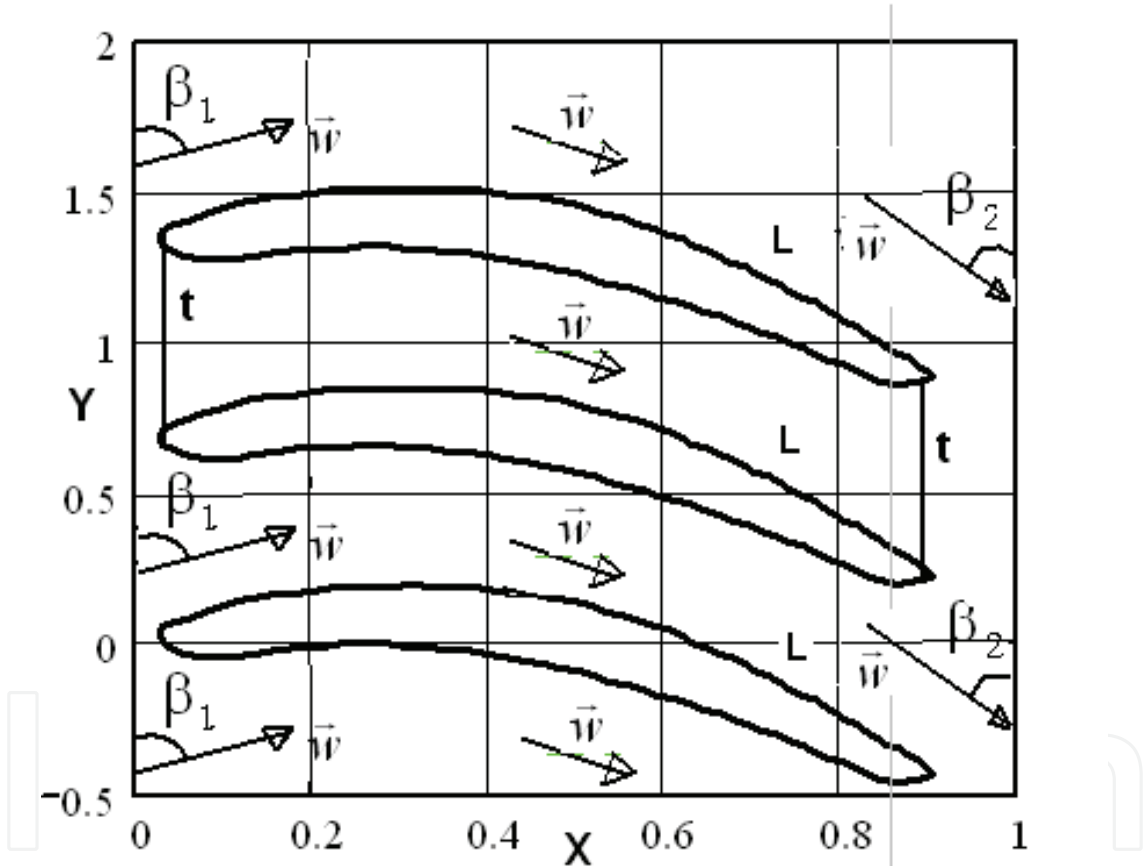


Fig. 1. Lattice of the profiles. \vec{w} is a vector of the flow speed, t is an increment of the lattice, β_1 is and input angle of the flow, β_2 is and output angle of the flow, L is a contour of the blade profile

This task comes [5] to the solution of integral equation of the following kind:

$$w(s)+\oint_L\left(K(s,l)-\frac{1}{L}\right)\cdot w(l)dl=b(s),$$

(6)

where $w(s)$ – normed speed of the flow;

$$K(s,l) = \frac{1}{t} \cdot \frac{\frac{\partial x}{\partial s} \cdot \sin\left(\frac{2\pi}{t} \cdot (y(s) - y(l))\right) - \frac{\partial y}{\partial s} \sinh\left(\frac{2\pi}{t} \cdot (x(s) - x(l))\right)}{\cos\left(\frac{2\pi}{t} \cdot (y(s) - y(l))\right) - \cosh\left(\frac{2\pi}{t} \cdot (x(s) - x(l))\right)};$$

$$b(s) = -2 \cdot \frac{\partial x}{\partial s} - \frac{\partial y}{\partial s} \cdot (\cot(\beta_1) + \cot(\beta_2)) + \frac{t}{L}.$$

Here s and l are values of the arch in different points of profile's edges, arches are counted from the middle of exiting border of the profile in the positive direction (counterclockwise); $x(l)$, $y(l)$ are the coordinates of the profile's point with the length of the arch l ; L is a contour of the blade profile; L is the length of the contour of the blade profile.

The direction of the unit tangent vector $\left\{ \frac{\partial x}{\partial s}, \frac{\partial y}{\partial s} \right\}$ is chosen in a way that the tracking of the contour would be made counterclockwise. As opposed to [5], in this research front side of the lattice is orientated not along the abscises axis, but along the ordinate axis. Besides that, in [5] the speed is normed so that the flow expense of the fluid on the output would be unitary, and in this research the speed is normed so that the absolute value of speed vector on the outcome of the lattice would be unitary. This is achieved by multiplication of the speed, received after solving equations (1) and $\sin(\beta_2)$. Exactly the second norm rate setting is applied in the computational program of the Ural Polytechnic Institute (UPI), where the computations were made with the method of rectangles with the optimal setting of the integration nodes [2, 6] The solutions, received in this program, have been chosen for the comparison in this research.

4. Scheme of application of semi-statical method to the problem of blade cascade flow

4.1 Main formulas

In this task contour L acts as a surface S , and an integral equation (6) with an unknown function $w(s)$ is solved. If by $w_k(s)$ we define an average solution after k iterations, and by $W_k(s)$ - value received on the integration with the number k after solving integral equation (1) on the N number of generated points, then we'll have

$$w_k(s) = \frac{1}{N} \cdot \sum_{m=1}^k W_k(s) \quad (7)$$

Selective standard deviation on the iteration with the number k is computed with the formula:

$$\delta_k(s) = \sqrt{\frac{1}{k^2} \cdot \sum_{l=1}^k D_l(s)}, \quad (8)$$

where D_l -selective dispersion in the end of one iteration with the number l ;

$$D_l(s)=\frac{1}{N(N-1)}\sum_{m=1}^N\left(\frac{K(s,l_m)-\frac{1}{L}}{p(l_m)}\cdot W_k(l_m)+b(s)-W_k(s)\right)^2$$

Here l_1, l_2, \dots, l_N are random points on the segment $[0, 2\pi]$, thrown on the iteration number k , N - a number of points in each iteration (in given below computational calculations is similar for all iterations), s -point of observation. Values of $w(l_m)$ are received as a result of approximated solution of integral equation (1) at the iteration number k - of the method. Computational practice has shown that deviation (calculating error) do not go behind the limits of standard deviation multiplied by three and, as usual, are within the boundaries of standard confidential (95%) interval.

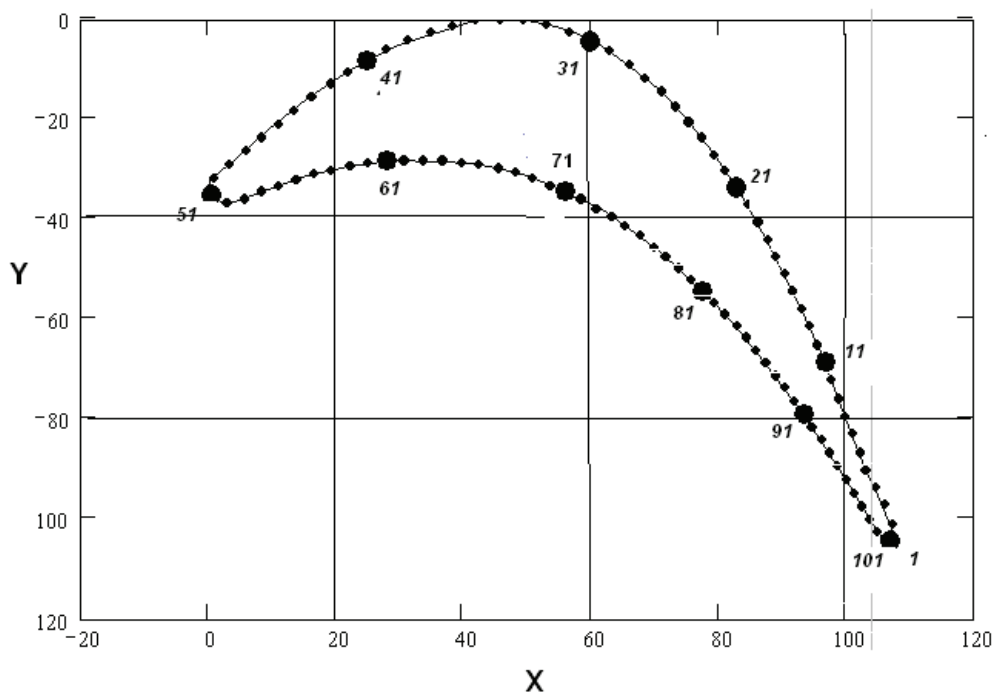


Fig. 2. Points, where the speed is calculated in computational examples – 50 equi-spaced points on the back and on the trough

Analytic definition of the blade contour

Integral equation (6) on the smooth contour L is of fredgolm type and has a unique solution [5]. Kernel of the equation (6) in case of two times differentiated contour can be considered continuous, as it has a removable singularity when $s=l$ [5]. In this research, however, contour is defined by a spline curve, first derivative of which has jumps in the finite set of points. This circumstance leads to the jumps of the kernel in the break points of the first derivative, which doesn't however influence the quality of computations. Besides that, the spline can always be approximated by a segment of Fourier row and the task can be solved on the infinitely derivated contour, as shown in [6]. Both approaches were tested, and the solutions made on a spline and on a segment of Fourier row were not considerably different. Semi-statical method was applied to the

equation (1) in accordance with the general scheme, which is described in detail in [3], without any additional preparation of regularization type. Values of the speed were calculated in 100 points of edging, 50 equal-spaced points on the trough and on the back correspondingly (Fig. 2). After that they were multiplied by $\sin(\beta_2)$, the result was compared to the solution, received by means of method of rectangles at the same contour. Both results were compared that to the one given by the UPI program. In the UPI program the contour is defined a little bit different, which causes insignificant divergences, which can be seen of the diagrams in the section of the results of computational modeling.

4.2 Computation algorithm and optimization.

The calculating was made iteratively. On each iteration a special number of random points on the segment $[0, 2\pi]$ was generated with the density, which was calculated by the results of previous iterations (adaptive algorithm). On the first iterations points were generated with uniform density on the segment $[0, 2\pi]$, which means approximately uniform distribution of the points on the contour of the blade, the results were defining more precisely by iterations, and approximated solution after iteration number i was considered to be arithmetic average of the solutions, received during previous iterations.

With the help of this approximated solution optimal density was calculated, using the method, described in [1]. Here algorithm is more economical, than described in [1], as it uses a more precise approximation to the right decision. As the computational practice has proved, on the strongly stretched contours on some iterations very strong spikes are possible, which are not smoothed by approximation even with the big number of iterations. However, it turned out that if the solution is very imprecise, than selective dispersions are also big in the check points, which are calculated during the work of the program.

We can introduce a constraint which will trace summands with a very big dispersion. In the current research the program is composed in a way that approximation is made not on all iterations, but only in those where relevant computational error, defined by the selective dispersion, is not bigger that 100 percent. Other solutions received on other iterations (usually not more than one percent from total number with the exclusion of the points close to edgings), are considered as spikes and are not included into the approximated finite sum. In case of much stretched working blade this improvement gives an undoubted advantage in the quality of computations.

With the help of semi-statistical method values of the speed were calculated in 150 points, distributed on the contour of the blade with an equal increment defined by the parameter u (which means practically equal increment on the arch length), and the values of the speed in checkpoints (which are distributed in the contour not evenly) were calculated with the help of interpolation. Selective dispersion is used as an index of precision of current approximation. It turned out that computer spends the most of time to calculating values of the kernel in the generated points, that's why the issue of decreasing number of generated points but saving precision of computation at the same time is important. In semi-statistical methods this can be achieved by optimization of the net of integration.

5. Results of computational modeling

To continue, let us introduce some denominations. On all the figures from 3 to 5 variable m stands for the number of point of observation, w_m is the speed in the point number m ,

calculated by means of semi-statistical method, w_{1m} is the speed in point number m , calculated by means of method of rectangles, w_{2m} is the speed in point number m , calculated by means of UPI program, $|w_m - w_{1m}|$ is absolute deviation (calculation error) of calculation of the speed in point number m by means of semi-statistical method in comparison to the method of rectangles. Phrase “speed, calculated by means of semi-statistical method (4*400)” will mean that for calculation 4 iterations of semi-statistical method were made, with 400 points generated in every iteration.

Next (Fig. 3 – Fig. 5) the results of computational modeling are shown.

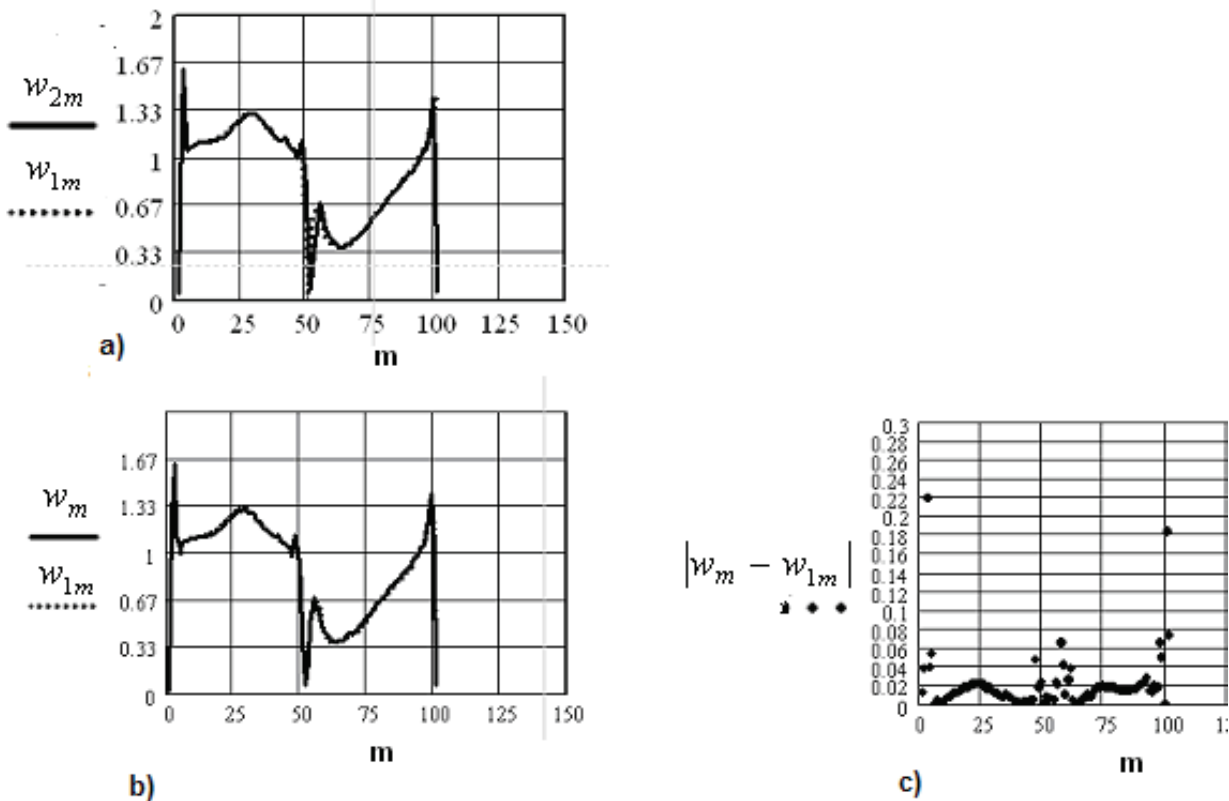


Fig. 3. Results of computational modeling on blade:

- a) Speed graph, calculated by means of method of rectangles and speed graph calculated by means of UPI program
- b) Speed graph, calculated by means of semi-statistical method (150*400) and speed graph, calculated by means of method of rectangles
- c) Absolute deviation graph of calculation of the speed by means of semi- statistical method (150*400) in comparison to the method of rectangles

From given above examples (Fig.3) it is evident, that semi-statistical method commutated the speed with a good precision in all the points of contour, except for some points in the edgings, which are not important for practical issues.

6. Analysis of effectiveness of density adaptation

It was very interesting to investigate, how adaptive algorithm works when choosing optimal density. It appears that the points become denser on the edgings and on the back, which

means exactly the same places of profile, where the quality of computation is very bad during first iterations. This is illustrated by the Fig. 4.

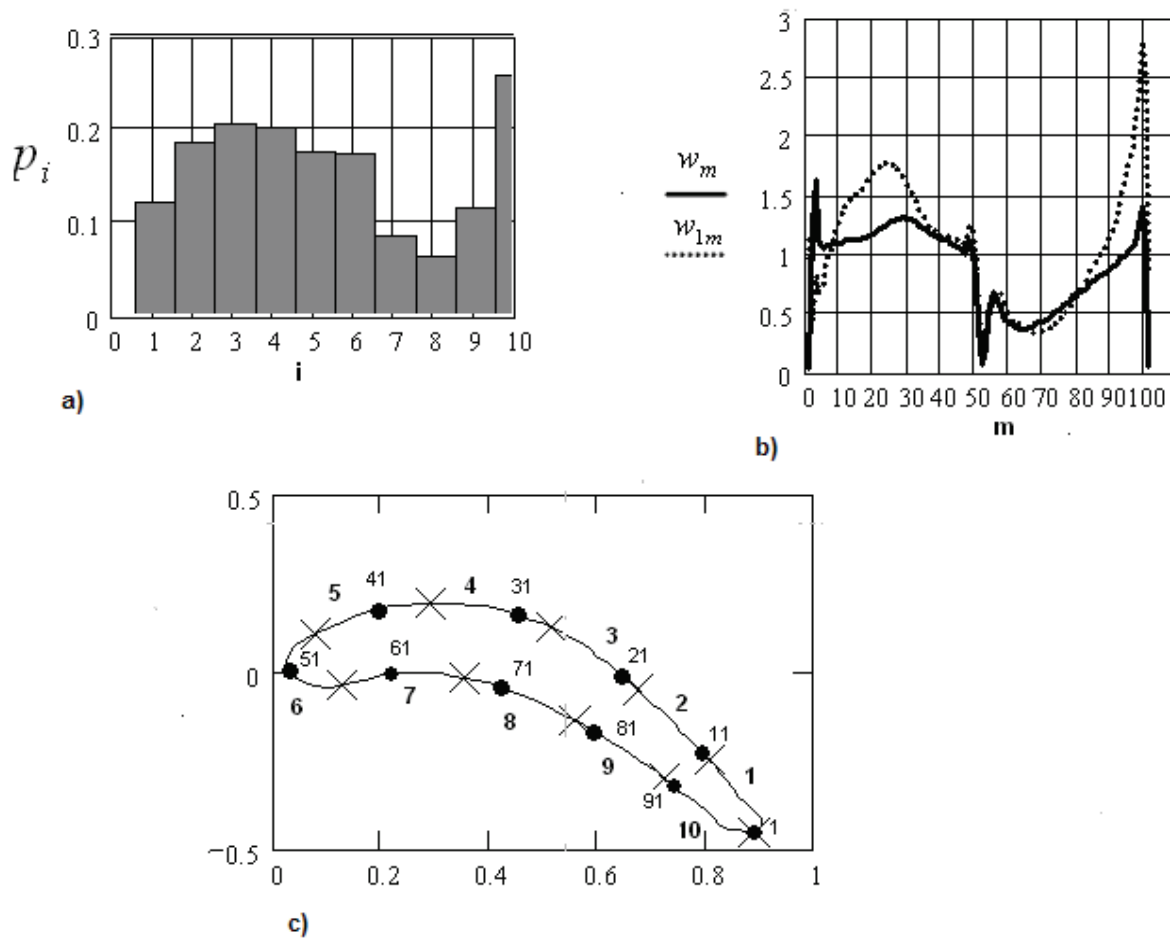


Fig. 4.
a) Histogram of optimal density after 2 iterations on the blade;
b) Speed graph calculated of the speed by means of semi-statistical method (2 iterations 400 points each) and speed graph calculated by means of method of rectangles.
c) With the symbol “x” borders of intervals from histogram on the Fig.3 a) are marked; numbers 1,2,...10 are the numbers of these intervals. Bold points are checkpoints (marked every 10 points starting with first); numbers 1,11,21,...92 – numbers of these points

On the Fig. 5 the results of computations on blade are shown, received after five iterations using adaptive algorithm for choice of optimal density and the results, received after five iterations with even distribution of generated points. It is easy to see that with the same number of generated points the results of adaptive algorithm are more precise. From the Fig. 5 it is clear that using adaptive algorithm makes standard mean-square distance lower in shorter period, that with even distribution. It allows reducing the number of thrown points which is necessary to achieve predefined precision.

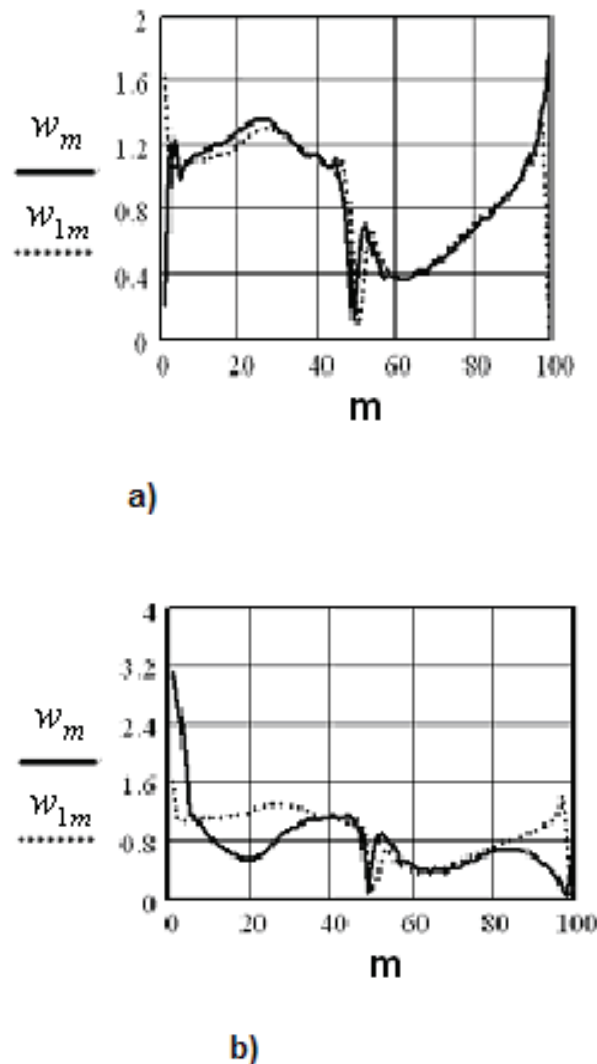


Fig. 5. Results of computational modeling on blade:
a) Speed graph, calculated by means of semi-static method (5*400) and speed graph, with the use of adaptive algorithm, calculated by means of method of rectangles
b) Speed graph, calculated by means of semi-static method (5*400) and speed graph, without the use of adaptive algorithm, calculated by means of method of rectangles

7. Conclusions

- To sum up the results of computational modeling, following conclusions can be drawn:
- By means of semi-static method quite precise results can be achieved solving the problem of potential lattice cascade flow.
 - In accordance with theoretical computations adaptive algorithm works for optimization of nodes on the domain of integration. It fastens convergence, reducing selective dispersion.
 - However in strongly-stretched areas convergence rate is not very fast. The problem of fastening the rate of convergence, which is necessary to make semi-static method successive in case of strongly-stretched areas and make it competitive to deterministic

methods in calculation speed, is still important. One of the ways to solve this problem is, evidently, improvement of the adaptive algorithm of optimization.

8. References

- Arsenjev D.G., Ivanov V.M., Kul'chitsky O.Y. (1999). *Adaptive methods of computing mathematics and mechanics. Stochastic variant*. World Scientific Publishing Co., Singapore
- Arsenjev D.G., Ivanov V.M., Korenevsky M.L. (2004). Semi-statistical and projection-statistical methods of numerical solving integral equations. *World Scientific and Engineering Academy and Society (WSEAS) Transactions on Circuits and Systems*, Issue 9, Volume 3: 1745-1749
- Arsenjev D.G., Ivanov V.M., Berkovskiy N.A. (2004). Application of semi-statical method to inside Dirichle problem in three-dimensional space. *Scientific proceedings of SPbSPU*, St.Petersburg, № 4 (38):52-59
- Isakov S.N., Tugushev N.U., Pirogova I.N. (1984). Report on scientific research. Development of computational software for calculating field of velocities and profile losses in lattices in compression in turbine buckets. *Proceedings of Ural polytechnical institute*, Sverdlovsk
- Jhukovskiy M.I. (1967). *Aerodynamic computing of flow in axial turbo-machines*. Mashgiz, Leningrad
- Vochmyanin S.M., Roost E.G., Bogov I.A. (1997). *Computing cooling systems for gas turbine blades. Bundled software GOLD*. International academy of high school, Petersburg

IntechOpen



Numerical Simulations - Applications, Examples and Theory

Edited by Prof. Lutz Angermann

ISBN 978-953-307-440-5

Hard cover, 520 pages

Publisher InTech

Published online 30, January, 2011

Published in print edition January, 2011

This book will interest researchers, scientists, engineers and graduate students in many disciplines, who make use of mathematical modeling and computer simulation. Although it represents only a small sample of the research activity on numerical simulations, the book will certainly serve as a valuable tool for researchers interested in getting involved in this multidisciplinary field. It will be useful to encourage further experimental and theoretical researches in the above mentioned areas of numerical simulation.

How to reference

In order to correctly reference this scholarly work, feel free to copy and paste the following:

D.G. Arsenjev, V.M. Ivanov and N.A. Berkovskiy (2011). Hybrid Type Method of Numerical Solution Integral Equations and its Applications, Numerical Simulations - Applications, Examples and Theory, Prof. Lutz Angermann (Ed.), ISBN: 978-953-307-440-5, InTech, Available from:
<http://www.intechopen.com/books/numerical-simulations-applications-examples-and-theory/hybrid-type-method-of-numerical-solution-integral-equations-and-its-applications>

INTECH
open science | open minds

InTech Europe

University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China

Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821

© 2011 The Author(s). Licensee IntechOpen. This chapter is distributed under the terms of the [Creative Commons Attribution-NonCommercial-ShareAlike-3.0 License](https://creativecommons.org/licenses/by-nc-sa/3.0/), which permits use, distribution and reproduction for non-commercial purposes, provided the original is properly cited and derivative works building on this content are distributed under the same license.

IntechOpen

IntechOpen