OPTIMIZATION METHODS USING HIGHER PROGRAMMING LANGUAGES IN POWER ENGINEERING

Ladislav Prskavec, Jan Merta, Ladislav Musil, Oldřich Matička
Department of Electrical Power Engineering
Czech Technical University in Prague,
Technická 2, 166 27 Prague 6
Czech Republic

ABSTRACT
The paper deals with the optimizing of distribution of weight-uneven vanes along the turbine's rotor circumference. The aim is to find vanes' positions in such a way, that aggregate centre of gravity lies in the minimum distance from the axis of the rotor. A method was suggested that does not optimize the distribution absolutely over all the possible order of vanes; however it enables us to compute from the practical point of view a relatively good solution very fast. A code written by authors in higher programming language Mathematica® was developed for solving the task. An illustrative example of optimizing of distribution of 36 vanes was computed.

KEYWORDS:
Mathematica, programming, optimization,

INTRODUCTION
The task of finding the best distribution of weight-uneven vanes along the turbine's rotor circumference is a frequently solved problem in rotary machines, which must be solved in power engineering.

Due to the manufacturing tolerances the vanes mounted on the rotor of a turbine are usually of unequal mass. The fluctuation in the mass is high enough to produce considerable eccentricity of the centre of gravity of the whole turbine. Eccentricity of the centre of gravity can be minimized by appropriate order of the vanes. Therefore the task is to find vanes' positions in such a way, that aggregate centre of gravity lies in minimum distance from the axis of the rotor.

Similar optimization can be applied to all kinds of rotary machines – compressors, turbines, pumps – which are indivisible part of every power engineering application.

MATHEMATICAL FORMULATION OF THE MODEL
In the terms of Mathematica®, the problem was formulated as follows:

Let's find the minimum of the quantity R – it is distance of gravity point from axis

$$R = abs[\sum_{i=0}^{n-1} (q + \alpha_i) \exp(2\pi ij / n)], \tag{1}$$

where α_i stands for deviation from basic mass and sequence $k(i)$ loops all permutations of numbers $i = 0,1,2,3,......., n-1$, where n is the number of vanes. Symbol j is the imaginary unit and q represents the base weight of vane number s.

For great number of vanes n (greater or equal to 10), number of permutations growths rapidly. Thetime consumption of computation grows as well.

THE PRINCIPLE OF THE USED SOLUTION
Because computing the distance of the centre of gravity from the axis of a turbine over all the possible permutations of this, the authors have drawn up faster procedure that finds sub-optimal solution and shows up some interesting mechanical and geometrical consequences. The solution can be conveniently used for educational purposes.

In mentioned approach, total amount of n vanes is divided into several independent subsystems. Each of these is optimized separately – e.g. for n = 6, we can create 2 subsystems of three vanes. After sub-optimizations are these subsystems put together – linear superposition principle - and their mutual positions are found in order to receive gravity point's minimum distance from the axis.

Every α_i – its weight - is chosen randomly from interval (0; 1). Basic mass q of vane doesn't have to be considered during optimization, because

$$R = abs[\sum_{i=0}^{n-1}(q)\exp(2\pi ij/n)] = 0 \tag{2}$$

for equal weights and even distribution.

We can therefore find just minimum of

$$R = abs[\sum_{i=0}^{n-1}(\alpha_i)\exp(2\pi ij/n)] \tag{3}$$

R - it is distance-deviation of gravity point from the axis.

A problem rises when n is prime-number (all positions around rotor body are equidistant). This problem is partially solved by dividing n vanes into two basic groups. First group includes the lightest, and its weight is taken as the base one. It is q equals to mass of the lightest vane (subscript l e.g.) and therefore α_l = 0. In the second section, there are n-1 resting vanes with α_j generally equal or greater than zero. In this case, we have to reserve one position for the vane with the basic weight. This position is now occupied and can not be considered in next steps of solution. (Example: 13 vanes = 2x6 + 1, or 3x4 + 1. analogically for 17 and so on)

In the case of n not being a prime number, the total amount is subdivided into several groups (sections) of similar weight. Number of vanes in every group and total number of sections is chosen as a compromise between mass distribution and number of possibilities. The first category covers heaviest vanes, and the last group lightest. (15 = 3x5, 21 = 3x7, 18 = 2x9, 16 = 2x8 e.g.). Vanes are placed in groups, symmetrically along the circumference. Every section is placed with phase displacement to previous one. Angle among vanes in every section is $\varphi_i = (2\pi/m)$, where m is vane-count in group. Phase among particular subsystems is $\varphi_z = (2\pi/km) = (2\pi/n)$.

All possible solutions of gravity point position for every section are calculated and the ones with minimal R is selected. The selected ones are than called the minimal permutations. Consequential gravity point is found by simple algorithms then: Minimal groups are rotated mutually and the final solution is the one with minimal R as mentioned above. This method is suboptimal, but it reaches answers precise enough. For q = 7.5kg, n=36 (for details see following parts of the paper) was gravity-point-displacement in order 10^{-5}-10^{-4} m and torque 10^{-3}-10^{-2}.

ILUSTRATIVE EXAMPLE

Process was compared for 36 vanes in three variants 6x6 ,4x9 and 9x4 vanes (k x m), for the same α_l – mass distribution function.

The basic mass of vane was q = 7,5 kg with weight deviations up to α_i = 1 kg.

Solutions are summarized in Table 2, where even subsolutions are stated (gravity points of sections, minimal permutations, total gravity-point displacement, number of rotations of each section and total computation time).

Vanes distribution	**6x6 (k x m)** **Six sections by six vanes**	**4x9** **Four systems by nine vanes**	**9x4** **Nine systems by four vanes**
Torque of the 1st systems grav. point (Nm)	0.0141307 0.0111011 – 0.00874306 i	0.00521338 0.00178698 – 0.00489756 i	0.0141307 0.0111011 – 0.00874306 i
Minimal permutation position	12	318760	12
Torque of the 2nd systems grav. point	0.0410646	0.00149026	0.0410646

(Nm)	0.0390966 + 0.0125604 ⅈ	0.00148745 − 0.000091475	0.0407418 + 0.00513935 ⅈ
Minimal permutation position	22	170160	12
Torque of the 3rd systems grav. point (Nm)	0.000957265 −0.000223231 − 0.000930873 ⅈ	0.0033988 0.001402 + 0.00309616 ⅈ	0.000957265 −0.000381484 − 0.0008779
Minimal permutation position	3	294532	3
Torque of the 4th systems grav. point (Nm)	0.0919468 0.0910723 + 0.0126512 ⅈ	0.00195448 0.00129428 − 0.00146453 ⅈ	0.0919468 −0.0113511 − 0.0912434 ⅈ
Minimal permutation position	12	64444	3
Torque of the 5th systems grav. point (Nm)	0.0172204 0.0108448 + 0.0133765 ⅈ	---------------	0.0172204 0.00886064 − 0.0147659 ⅈ
Minimal permutation position	13	---------------	11
Torque of the 6th systems grav. point (Nm)	0.0759302 0.023505 + 0.0722005 ⅈ	---------------	0.0759302 0.0606667 − 0.0456612 ⅈ
Minimal permutation position	13	---------------	12
Torque of the 7th systems grav. point (Nm)	---------------	---------------	0.0626931 0.0599724 − 0.0182686 ⅈ
Minimal permutation position	---------------	---------------	13
Torque of the 8th systems grav. point (Nm)	---------------	---------------	0.0615769 −0.0390284 − 0.0476287 ⅈ
Minimal permutation position	---------------	---------------	4
Torque of the 9th systems grav. point (Nm)	---------------	---------------	0.0103278 −0.00888736 + 0.0052610
Minimal permutation position	---------------	---------------	22
Agregate torque (Nm)	0.00507014	0.000204153	0.000842529
Phase shifts of individual subsystems with respect to fixed one, φ_c	{3, 1, 1, 4, 2}	{6, 2, 4}	{4, 1, 1, 4, 2, 2, 4, 2}
Solving time	0.11 Second	79.826 Second	8.094 Second

Table 1 Comparison of three variants of solution.

```
{{0.428829, 1}, {0.0351304, 2}, {0.995983, 3}, {0.336787, 4}, {0.251773, 5}, {0.279987, 6},
 {0.606889, 7}, {0.969561, 8}, {0.0371438, 9}, {0.482896, 10}, {0.355486, 11}, {0.874401, 12},
 {0.465889, 13}, {0.493828, 14}, {0.281335, 15}, {0.992345, 16}, {0.710039, 17}, {0.647321, 18},
 {0.252432, 19}, {0.0314859, 20}, {0.959895, 21}, {0.806242, 22}, {0.619843, 23}, {0.201133, 24},
 {0.531065, 25}, {0.771111, 26}, {0.623861, 27}, {0.864347, 28}, {0.279292, 29}, {0.491124, 30},
 {0.0169713, 31}, {0.894786, 32}, {0.242148, 33}, {0.00822824, 34}, {0.661486, 35}, {0.0203848, 36}}
```

Table 2 Selected values of α_i

Summarized distributions of vanes in several groups for all solutions, final positions of these subsections (phases) and torques before and after rotation are depicted in Figures 1 - 6. For our purpose was selected random α_i. The selected values of α_i are depicted in Table 2.

Nine systems by four vanes

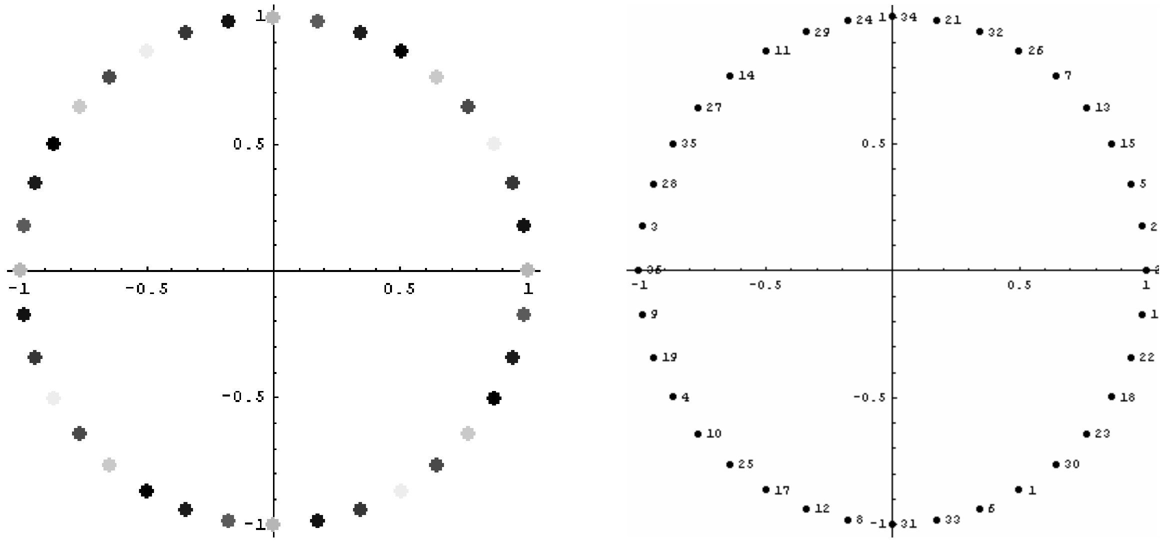

Figure 1 Positions of 9x4 vane subsystems (each subsystem is signed by different colour) on the left and order of vanes on the right

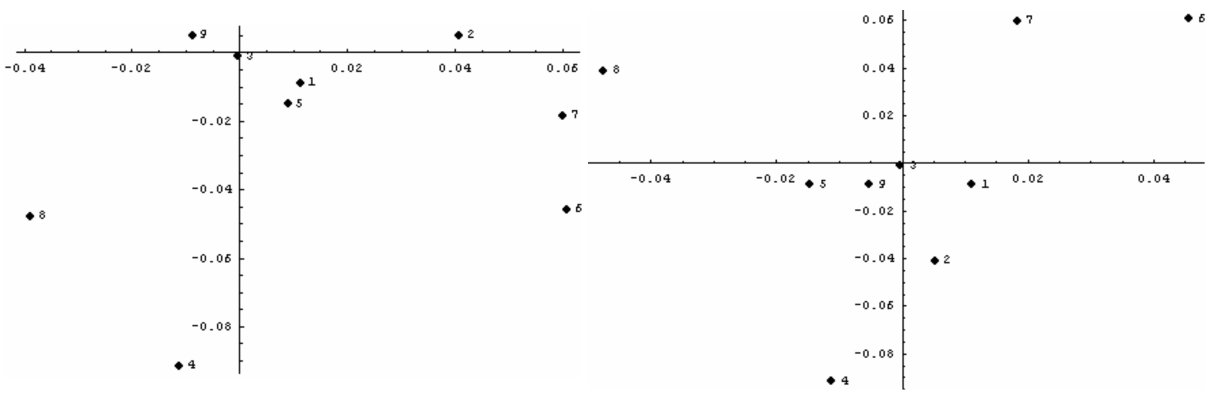

Figure 2 Gravity points of 9x4 vane subsystems positions before (on the left) and after (on the right) the optimization of total torque

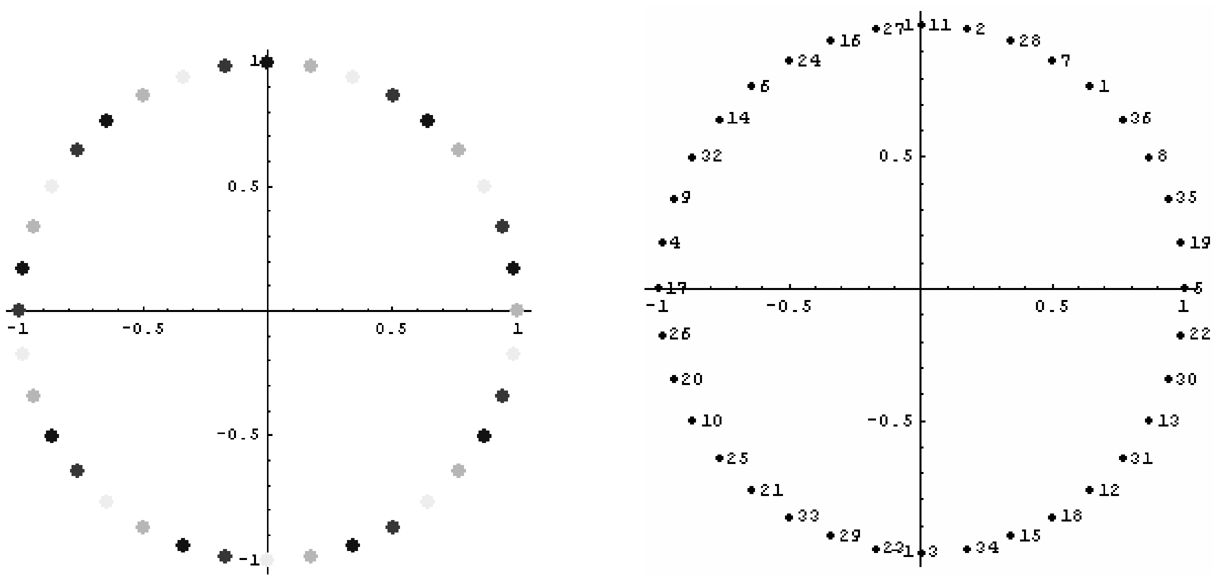

Figure 3 Positions of 4x9 vane subsystems (each subsystem is signed by different colour) on the left and order of vanes on the right

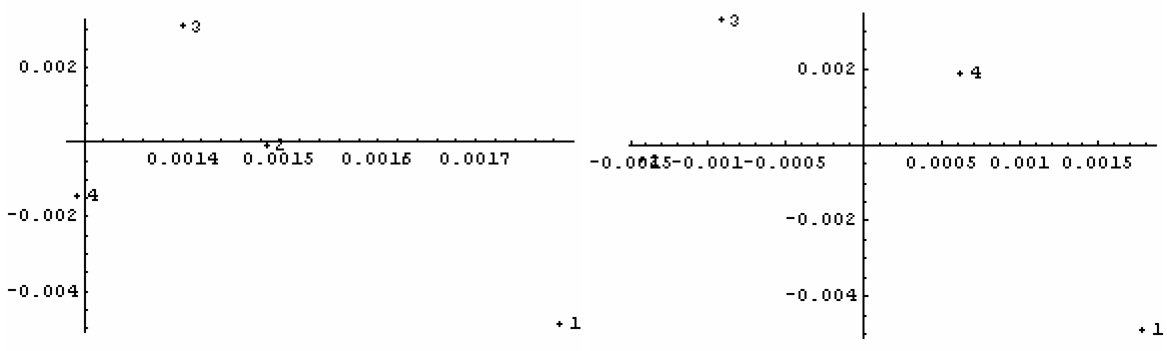

Figure 4 Gravity points of 4x9 vane subsystems positions before (on the left) and after (on the right) the optimization of total torque

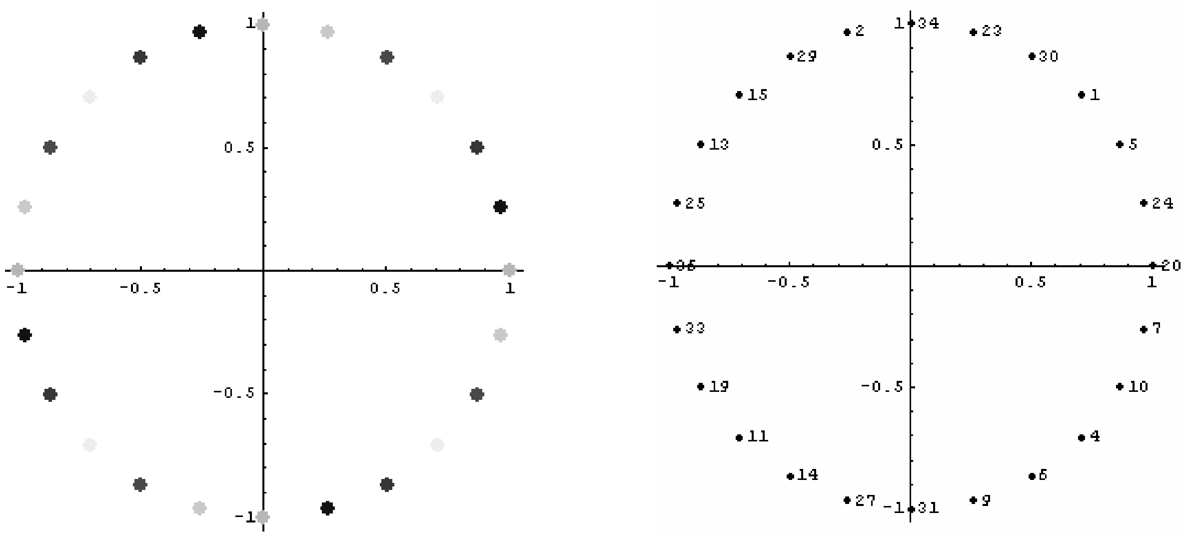

Figure 5 Positions of 6x6 vane subsystems (each subsystem is signed by different colour) on the left and order of vanes on the right

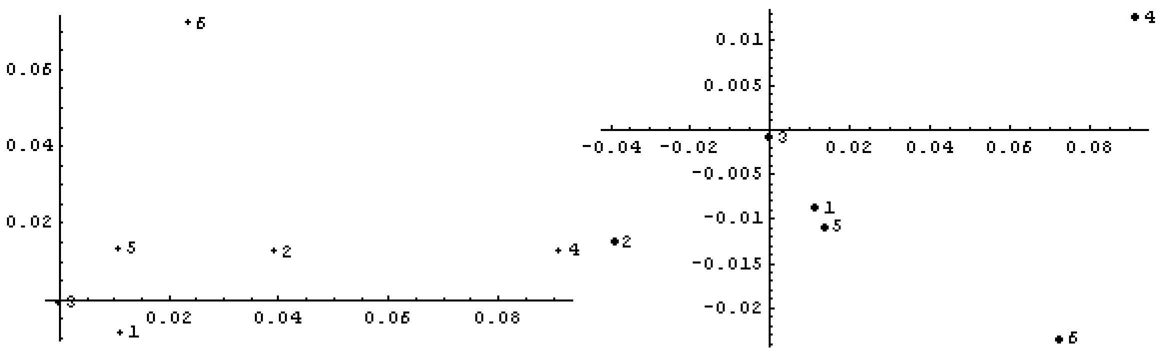

Figure 6 Gravity points of 6x6 vane subsystems positions before (on the left) and after (on the right) the optimization of total torque

The solution shows up, that if system of vanes can be subdivided into several groups of similar mass, then principle of linear superposition can be used to lower the computation time. Every section is optimized separately. Partial solutions are combined then. Even if not all permutations considered, as in this case, solution is precise enough. The more subsystems, we divide vanes into, the lower numeric exacting is, without considerable loss of precision.

CONCLUSION

Higher programming language provides us a powerful tool for advanced engineering problems solutions. Of course, the knowledge of "traditional" mathematics is essential for the access to the problem, but a link between the possibilities of it and the possibilities of higher programming languages and environments are substantial, too. The paper shows the method using the theory of a complex variable together with Mathematica®.

REFERENCES

[1] Wolfram, S.: The *Mathematica® book.* USA: Wolfram Media and Cambridge University Press, 1996.
[2] Šulista, M.: *Fundamental of Complex Analysis.*(in Czech) Praha: SNTL, 1981.