Development of software systems for the design of technically complex products

E I Martyanov and E N Malygin
Tambov State Technical University, 106, Sovetskaya str., Tambov, 392000, Russia
E-mail: martyanovei@gmail.com

Abstract. In the article, we propose the algorithm of development of program system which can be applied to designing technically complex products for the mathematical description of which the systems of differential equations with partial derivatives are used. This system allows us to obtain more accurate results than engineering methods and is relatively easy to use. As an example of implementing the proposed algorithm, we have created a subject-oriented software system designed to solve the problem of determining a two-blade stirrer's design parameters and operating characteristics.

1. Introduction
Technically complex products include airplanes, ships, cars, stirred reactors, vulcanizing presses, scrubbers, etc. These products share a common problem, namely, in their design, intricate mathematical designs based on partial differential equation systems have to be used. However, industrial enterprises use physical modeling or develop simplified engineering calculation methods to design such products. This approach is not correct because it requires a considerable amount of time to develop and implement considerable material costs and is not always very precise. Even with the simple example of mechanical mixing devices, which are installed in chemical or pharmaceutical apparatuses, it can be seen that the application of various kinds of engineering methods can determine the design parameters and regime characteristics of a stirrer. However, they do not have sufficient accuracy because they have been obtained through physical experimentation on laboratory installations for stirrers of a particular design. Any deviation from the proposed stirrer designs leads to errors and sometimes completely implausible results [1].

In the mathematical description of processes implemented in such apparatuses (hydrodynamic, thermal, etc.), it is necessary to use complex mathematical constructions based on the solution of systems of differential equations with partial derivatives. This approach is inaccessible for industrial enterprises. It requires highly skilled personnel and time-consuming implementation, so either a physical experiment in a specific production situation is usually used, or engineering methods recommended for use by parent organizations are used.

More accurate results can be obtained using various CAE systems: ANSYS, COMSOL, STAR-CD, QForm, Nastran, or Fluent [2]. However, the application of these software packages is limited by high requirements to the used computer equipment's capabilities and requires high personnel qualification. These complexes are oriented to solve a wide range of tasks and require adjustment to the subject area in question. As a result, there may be losses in computational speed and accuracy of the results obtained. Moreover, this is why developing subject-oriented software systems for calculating
technically complex products, which increase the speed and accuracy of calculations by applying specialized methods and algorithms specific to the product in question, is now very relevant.

2. Formulation of the target

The development of such software systems is based on the formulation of a specific problem [3]. For clarity, let us consider the problem of finding the operating characteristics and design parameters of a two-blade stirrer for a vertical vessel apparatus. In our case, it is formulated as follows: "At fixed diameter and height of a vertical vessel it is necessary to find the diameter of the stirrer, the width of its blades, the height of installation above the vessel bottom and rotation frequency at which the total vector of the speed of liquid flow reaches the maximum at admissible power inputs for stirring".

The varied parameters are diameter (length) $D_m$, width $H_m$ and height of the stirrer above the bottom $H_{hm}$. They are crucial for the stirrer, and changing these parameters does not reduce the technological efficiency of constructing the apparatus. It is always easier to select and create a new stirrer than to redesign the entire vessel in production restructuring or changeover of equipment. For the same reason, the dimensions that define the vessel, namely the diameter $D_t$ and height $H_t$, have been fixed. These dimensions are mostly standardized and are determined either by the technological possibilities of production or by the state standards. The speed is also a variable parameter because it has the most significant influence on the intensity of mixing and, therefore, on the equipment's efficiency in general. The parameters to be varied are in the following ranges:

$$ D_m \in [0.1D_t, 0.9D_t], $$

(1)

$$ H_m \in [0.01D_m, 0.2D_m], $$

(2)

$$ H_{hm} \in [0.4D_m, 0.8H_t], $$

(3)

$$ n \in [0.200], $$

(4)

The interval (1) is determined from the condition of compliance with the norms and rules of design of technological equipment (there must be a gap between two mutually moving parts to prevent them from colliding [4, 5]).

Intervals (2), (3), and (4) are determined by the requirements for the type of agitating device in question [4, 6].

The thickness of the stirrer blade has practically no influence on the mixing process and is determined by the formula:

$$ S_m = 0.1H_m. $$

(5)

The primary criterion for the optimum is the total vector of the speed of liquid flow $K1$. It is defined as the sum of all velocity vectors in the object in question:

$$ K1 = \sum_{i=1}^{z} \vec{u}_i. $$

(6)

where $z$ – the total number of nodes in the computational model, $\vec{u}_i$ – the velocity vector at a node in the computational model.

The intensity of mixing can be evaluated in various ways, but the proposed criterion allows this to be done as efficiently as possible. For example, let us consider two or more objects of equal diameter and height, and the number of nodal points does not differ by more than 5%. The best object will be the one with the more significant total velocity vector. Moreover, this is because it will have fewer stagnant zones (areas of low flow velocity), and therefore its efficiency will be higher than that of the others. However, it should be remembered that the mixing rate cannot be increased indefinitely, so an additional (compensation) criterion has been introduced. Namely, the power expended on mixing. It
will compensate for the excessive growth of the total velocity vector. In other words, the introduced penalty function will prevent the growth of the total velocity vector and make the designed plant not only efficient in terms of mixing but efficient in terms of energy consumption.

3. Selecting a programming language
At the moment, there are many programming languages suitable for the development of object-oriented mechanical stirrer parameters selection software: Python, Java, JavaScript, Perl, Tcl, or Smalltalk. After analyzing and comparing the characteristics of these languages, it was decided to use Python. It has the following features [7]:

- programs written in this language are significantly shorter than equivalent programs in other languages due to built-in high-level data types and dynamic typing;
- supports a programming style that uses simple functions and variables without including classes in the definition, allowing for lengthy programs;
- focuses on standard programming methodologies such as data structures and object-oriented programming.

4. The algorithm for the software system
The object-oriented software system's operation algorithm for selecting a mechanical stirrer's parameters is shown as a flowchart in figure 1.

The values of the fixed parameters $D_t$, $H_t$, $V$ (liquid volume), $\rho$ (liquid density), $v$ (kinematic viscosity) and the initial values of the varying parameters $D_m$, $H_m$, $H_{hm}$ and $n$ are transferred to FreeCAD, where a 3D model of a vertical vessel with a flat bottom is created.

FreeCAD is a general-purpose parametric CAD system with open source code. It is based on the boundary representation principle for the solids' geometric modelling and supports polygonal meshes. An exceptional feature of this program is its ability to work without a graphical interface and the possibility of creating programs with the help of Python programming language.

The 3D model is built in the background without operator intervention. The process takes less than a few seconds, as there is no need to use the GUI and draw all the objects. The 3D model and input data are transferred to another program, Gmsh, where a finite element mesh is generated according to a predefined algorithm.

Gmsh is an open-source 3D finite element generator. It is a fast, lightweight, and easy to use finite element mesh generation tool with parametric input. It includes its programming language but can also work with Python. Not only that, but it can be used to build meshes of any configuration and complexity, whether using built-in algorithms [8]. Like its predecessor, Gmsh works without a graphical interface and therefore consumes almost no computational resources of the computer.

After constructing the grid and checking the result, the data is transferred to OpenFOAM, where the vessel's velocity field is calculated.

OpenFOAM is a freely distributed computational fluid dynamics toolkit for field operations (scalar, vector, and tensor). It is designed to perform hydrodynamic problems for both non-Newtonian and Newtonian viscous fluids in both incompressible and compressible approximations, taking into account convective heat transfer and gravitational forces. RANS, LES, and DNS methods can be used to simulate turbulent flows. It is possible to solve subsonic, transonic, and supersonic problems. It also allows us to solve other tasks:

- strength calculations;
- problems of heat conduction in solids;
- multiphase problems, including a description of chemical reactions of the flow components;
- problems related to the deformation of the computational mesh;
- other problems that require the solution of partial differential equations in a complex medium geometry.
The program is based on libraries that provide tools for solving partial differential equations systems, both stationary and non-stationary. In terms of OpenFOAM, most differential and tensor operators can be represented in the program code (before translation into an executable file) in a human-readable form. The user can already select the discretization and solution method for each
operator during computation. Thus, the computational finite element mesh (discretization method), discretization of the basic equations, and their solution methods are entirely separate.

The Reynolds averaged Navier-Stokes equations (7) - (10) [9] were used to calculate the targets in OpenFOAM:

\[
\frac{d\bar{u}_L}{dt} + \bar{u}_L \frac{d\bar{u}_L}{dr} + \bar{u}_R \frac{d\bar{u}_R}{r \, d\phi} = \frac{-1}{\rho} \frac{dP}{dr} + \nu \left( \frac{d^2\bar{u}_L}{dr^2} + \frac{d^2\bar{u}_R}{r \, dr} + \frac{1}{r} \frac{d\bar{u}_R}{d\phi} \right) + \rho \left[ \frac{d}{dr} \left( -\rho \ddot{u}_L \dot{u}_R \right) + \frac{1}{r} \frac{d}{d\phi} \left( -\rho \ddot{u}_R \dot{u}_R \right) \right],
\]

(7)

\[
\frac{d\bar{u}_R}{dt} + \bar{u}_L \frac{d\bar{u}_R}{dr} + \bar{u}_R \frac{d\bar{u}_L}{r \, d\phi} \frac{d\bar{u}_R}{r \, d\phi} + \rho \left[ \frac{d}{dr} \left( -\rho \ddot{u}_R \dot{u}_R \right) + \frac{1}{r} \frac{d}{d\phi} \left( -\rho \ddot{u}_R \dot{u}_R \right) \right],
\]

(8)

\[
\frac{d\bar{u}_\phi}{dt} + \bar{u}_L \frac{d\bar{u}_\phi}{dr} + \bar{u}_R \frac{d\bar{u}_\phi}{r \, d\phi} + \bar{u}_\phi \frac{d\bar{u}_\phi}{d\phi} + \frac{\ddot{u}_R \dot{u}_R}{r} = \frac{F_R}{\rho r} \frac{dP}{d\phi} + \nu \left( \frac{d^2\bar{u}_L}{dr^2} + \frac{d^2\bar{u}_R}{r \, dr} + \frac{1}{r} \frac{d\bar{u}_R}{d\phi} \right) + \rho \left[ \frac{d}{dr} \left( -\rho \ddot{u}_L \dot{u}_R \right) + \frac{1}{r} \frac{d}{d\phi} \left( -\rho \ddot{u}_R \dot{u}_R \right) \right],
\]

(9)

\[
\frac{d}{dt} \left( r \ddot{u}_L \right) + \frac{d}{dr} \left( r \dot{u}_R \right) + \frac{d}{d\phi} \left( r \dot{u}_R \right) = 0.
\]

(10)

where \(\bar{u}_L, \ddot{u}_R, \ddot{u}_\phi\) – averaged values of velocity vector components, m/s; \(\dot{u}_L, \ddot{u}_R, \dot{u}_\phi\) – pulsation components of the velocity vector, m/s; \(t\) – time, s; \(l\) – liquid level, m; \(r\) – vessel radius, m; \(\phi\) – polar angle, deg; \(v\) – kinematic viscosity, m²/s; \(\rho\) – liquid density, kg/m³; \(P_L, P_R, P_\phi\) – hydrodynamic pressure, Pa; \(F_L, F_R, F_\phi\) – mass forces, N.

Bernoulli integral is used to describe the liquid's hydrodynamic pressure because the homogeneous incompressible liquid is considered, and the steady-state motion of liquid takes place [10]. The Bernoulli integral to the problem under consideration is represented by equations (11) - (13).

\[
P_L = \frac{\rho u_L^2}{2} + \rho g l,
\]

(11)

where \(g\) – the acceleration of gravity, m/s².

\[
P_R = \frac{\rho u_R^2}{2} + \rho g l,
\]

(12)
P_v = \frac{\rho u_0^2}{2r} + pg l. \tag{13}

In addition to hydrodynamic pressure, mass forces (14) - (16), which act on liquid from stirrer while it rotates around its axis, are also taken into account in the calculation [9]:

F_L – the force acting on the particles in the liquid, which causes them to settle (float):

\begin{equation}
F_L = 0, \tag{14}
\end{equation}

This happened because we are considering a uniform homogeneous liquid where no settling or stratification process happens.

F_R – the centrifugal force:

\begin{equation}
F_R = \overline{R} \frac{d\varphi^2}{dt}, \tag{15}
\end{equation}

where \( \overline{R} \) – the radius-vector of the point in the coordinate system in use.

F_\varphi – the force exerted by the agitator on the fluid side:

\begin{equation}
F_\varphi = \frac{P_m}{R^2} \frac{d\varphi}{dt}. \tag{16}
\end{equation}

where \( P_m \) – the power of the stirrer motor, W.

Also, the RNG \( k-\varepsilon \) turbulence model (17) - (18) is used for calculation. This model was used because it has the following features [9]:

- it has an additional condition in the equation for turbulent dissipation rate \( \varepsilon \), which improves the accuracy of solving highly stressed flows;
- it takes into account the effect of circulating turbulence, which improves the solution accuracy for high velocity rotating and circulating flows;
- the analytical relationship for calculating Prandtl number for the flow during the solution, while in the standard \( k-\varepsilon \) turbulence model this parameter is a constant, is introduced;
- the analytical formula for the dynamic viscosity is introduced, which allows for the more qualitative calculation of low Reynolds number turbulent flows, but works with qualitative grid resolution in the boundary layer region.

Moreover, it has the following form [9]:

\begin{equation}
\frac{d}{dt}(\rho k) + \frac{d}{dr}(\rho k \overline{u}_L) + \frac{1}{r} \frac{d}{dr}(\rho k \overline{u}_r) + \frac{1}{r} \frac{d}{d\varphi}(\rho k \overline{u}_\varphi) = \\
= \frac{d}{dl} \left( \alpha_k \mu_{\text{eff}} \frac{dk}{dl} \right) + \frac{d}{dr} \left( \alpha_k \mu_{\text{eff}} \frac{dk}{dr} \right) + \frac{1}{r} \frac{d}{d\varphi} \left( \alpha_k \mu_{\text{eff}} \frac{1}{r} \frac{dk}{d\varphi} \right) + G_K + G_b - \rho \varepsilon, \tag{17}
\end{equation}

\begin{equation}
\frac{d}{dt}(\rho \varepsilon) + \frac{d}{dr}(\rho \varepsilon \overline{u}_L) + \frac{1}{r} \frac{d}{d\varphi}(\rho \varepsilon \overline{u}_r) + \frac{1}{r} \frac{d}{d\varphi}(\rho \varepsilon \overline{u}_\varphi) = \\
= \frac{d}{dl} \left( \alpha_\varepsilon \mu_{\text{eff}} \frac{d\varepsilon}{dl} \right) + \frac{d}{dr} \left( \alpha_\varepsilon \mu_{\text{eff}} \frac{d\varepsilon}{dr} \right) + \frac{1}{r} \frac{d}{d\varphi} \left( \alpha_\varepsilon \mu_{\text{eff}} \frac{1}{r} \frac{d\varepsilon}{d\varphi} \right) + C_{1_\varepsilon} \frac{\varepsilon}{k} \left( G_K + C_{3_\varepsilon} G_b \right) - \\
- C_{2_\varepsilon} \rho \frac{\varepsilon^2}{k} - R_\varepsilon. \tag{18}
\end{equation}
where \( k \) – kinetic energy, N; \( \varepsilon \) – energy dissipation; \( \mu_{\text{eff}} \) – effective viscosity; \( a_k, a_\varepsilon \) – inverse effective Prandtl numbers for \( k \) and \( \varepsilon \) respectively; \( G_k \) – turbulent kinetic energy, N; \( G_b \) – kinetic energy of pushing force, N; \( C_{1\varepsilon}, C_{2\varepsilon}, C_{3\varepsilon} \) – empirical constants.

However, the presented model does not consider the surface roughness of the apparatus wall and bottom and the surface of the agitator itself. Since in most cases the inner surface is smooth enough (roughness height not more than 25 \( \mu \)m), especially for the food and pharmaceutical industry, while the surface of the stirrer under the action of friction forces is polished and acquires an almost mirror-like surface (roughness height not more than 12.5 \( \mu \)m) [5]. Also, friction force on the apparatus’s inner surface is not considered as there are no turbulence baffles in the wall area. Therefore, a steady laminar flow is observed, so the velocities \( \bar{u}_L, \bar{u}_R, \bar{u}_\phi \) conditionally remain unchanged, and the kinetic energy \( k \) and dissipation velocity \( \varepsilon \) are equal to zero. The OpenFOAM program runs in the background.

5. Analysis of the results of the study

For testing the adequacy of the mathematical model and the proposed approach, the data obtained during the experiment was compared with the data obtained during the calculation in the software packages ANSYS and COMSOL and the developed software system.

A laptop with the following specifications was used for the calculations:
- Processor: Intel Pentium 2020M (2.4 GHz),
- RAM: 8GB DDR3 (800MHz),
- Graphics card: NVIDIA GeForce GT 720M 2GB (DDR3),
- Hard drive: WD Blue 500GB (5400rpm).

The results obtained were summarized in table 1.

| Parameters | Stirrer speed, rpm | Power, mW | Power deviation from experimental data, % | Number of elements in a model | Calculation time, sec | Time deviation from the calculation in the developed program, % |
|------------|------------------|-----------|----------------------------------------|-----------------------------|-----------------------|-------------------------------------------------------------|
| Experimental data | 20 | 0.35 | - | - | 54 000 | ~ +35 000 |
| | 40 | 2.47 | - | - | 149 | - 2 |
| | 60 | 8.10 | - | 220103 | 183 | + 30 |
| | 80 | 19.30 | - | - | 224 | + 35 |
| | 100 | 38.50 | - | - | 272 | + 47 |
| Calculation in ANSYS | 20 | 0.32 | - 9 | - | 149 | - 2 |
| | 40 | 2.25 | - 10 | - | 137 | + 8 |
| | 60 | 7.74 | - 5 | 220103 | 183 | + 30 |
| | 80 | 18.60 | - 4 | - | 224 | + 35 |
| | 100 | 36.90 | - 4 | - | 272 | + 47 |
| Calculation in COMSOL | 20 | 0.28 | - 25 | - | 156 | + 3 |
| | 40 | 2.08 | - 18 | 213692 | 216 | + 53 |
| | 60 | 7.12 | - 14 | - | 168 | + 32 |
| | 80 | 17.20 | - 12 | - | 252 | + 52 |
| | 100 | 35.10 | - 10 | - | 336 | + 82 |
| Calculation in the developed program | 20 | 0.38 | + 9 | - | 152 | - |
| | 40 | 2.54 | + 3 | - | 127 | - |
| | 60 | 7.89 | + 2 | 210401 | 141 | - |
| | 80 | 18.91 | - 2 | - | 166 | - |
| | 100 | 36.96 | - 4 | - | 185 | - |
As you can see, the use of calculation software reduces the time required to obtain the results several times. The time indicated in the row "Experimental data" is the actual time spent for each power measurement, including checking and readjusting the equipment. The experiment's essence was to determine the electric motor's power consumption using the indirect method, i.e., by measuring the current on the winding of the electric motor itself.

The time specified in the lines "Calculation in ANSYS" and "Calculation in COMSOL" is given without considering building the model and computational grid. In other words, it is time spent only on a single calculation. Simultaneously, the time indicated in the line "Calculation in the developed program" takes into account the whole calculation cycle without taking into account the optimization process. It takes into account time spent on model building followed by mesh generation and calculation of parameters.

6. Conclusion

As it is possible to notice on the data submitted in table 1, the received subject-oriented programming system (figure 2) not only is more convenient in circulation but also gives more exact results (the deviation from the received experimental data does not exceed 10 %) for a shorter interval of time (speed of calculation in some cases almost in 2 times less than at similar programs).

As the offered algorithm of working out the program system can be applied to designing other products for which mathematical description of the differential equations system with partial derivatives is used, for example, to the tasks of determining:

- thermal fields distributed over the surface of heating plates of vulcanizing hydraulic presses;
- efficiency of gas cleaning in cyclones or scrubbers;
- longitudinal section of beams of metal structures of load-carrying structures and metal building trusses;
- the design of load-bearing elements located outdoors in various climatic zones is subject to wind loads or precipitation.

![Figure 2. Screenshot of the developed object-oriented software system.](image-url)
Such an approach will be useful for industrial enterprises' design and engineering departments. They develop typical products of different configurations and comprehensive nomenclature or size range, namely for enterprises of chemical engineering, heavy engineering, machine-tool, and instrument-making plants.

A new approach to solving mechanical mixing device design problems using a new mathematical model for describing the mixing process has been proven to be effective.

7. Referents

[1] Martyanov E I and Malygin E N 2018 Virtual Modelling, Prototyping and Industrial Design: Proceedings of the V Int. Scientific and Practical Conf. vol 1 (Tambov: Tambov State University Press) pp 549-53

[2] Borovkov A I 2012 Computer engineering (St. Petersburg: Polytechnic University Press) p 93

[3] Malygin E N 2010 Mathematical methods in technical calculations (Tambov: Tambov State University Press) p 80

[4] Braginsky L N, Begachev V I and Barabash V M 1984 Agitation in fluid media. Physical fundamentals and engineering calculation methods (Leningrad: Publisher Chemistry) p 336

[5] Laszczyński A A and Tolczynski A R 1970 The fundamentals of design and calculation of chemical reactors (Leningrad: Publisher Mechanical Engineering) p 752

[6] Karpushkin S V, Krasnyansky M N and Borisenko A B 2009 Calculation and selection of mechanical stirring for vertical vessels (Tambov: Tambov State University Press) p 168

[7] Heinemann D, Poyais G and Seyakov S 2017 Handbook with examples in C, C++, Java and Python (St. Petersburg: Alpha Books Ltd) p 432

[8] Geuzaine C and Remacle J-F 2009 International Journal for Numerical Methods in Engineering vol 39 (New York: John Wiley & Sons) pp 1309-31

[9] Martyanov E I and Malygin E N 2020 Proceedings of the St. Petersburg State Institute of Technology vol 54 (St. Petersburg: St. Petersburg State University Press) pp 107-11

[10] Wallander S V 1978 Lectures on hydroaeromechanics (Leningrad: Leningrad University Press) p 296