



Professor M.A. Bogdevicius was born on January 2, 1958 in the Vilnius (Lithuania). Since 1995 M.A. Bogdevicius is the Head of the Department of Transport Technological Equipment of Vilnius Gediminas Technical University. He is member of the International Academy of Noosphere.

M.A. Bogdevicius has published more than 180 scientific and technical works; he is the author of 6 books and 25 inventions. M.A. Bogdevicius's interests include such topics as: computation methods, methods of optimization, dynamics of mechanical, hydraulic and pneumatic systems, problem of safety of road motion as well as software for the solution of different technical and manufacturing problems.

Professor V. Z. Aladjev was born on June 14, 1942 in the town Grodno (Byelorussia). Now, he is the First vice-president of the International Academy of Noosphere and the president of Tallinn Research Group, whose scientific results have received international recognition, first, in the field of mathematical theory of Cellular Automata (CA). He is member of a series of Russian and International Academies.



V. Z. Aladjev is the author of more than 350 scientific publications, including 65 books, published in many countries. He participates as a member of the organizing committee and/or a guest lecturer in many international scientific forums in mathematics and cybernetics.



This book represents the extended presentation of materials of lectures on the fundamentals of work in the environment of mathematical package Maple, which were given by the authors in 2001-2005 for the researchers of the Baltic Branch of the International Academy of Noosphere (Tallinn, Estonia), Vilnius Gediminas Technical University (Vilnius, Lithuania), Grodno State University (Grodno, Byelorussia). The given courses aimed at the introduction of innovations of the new software for Maple with orientation, above all, on the experts in the field of physical and mathematical sciences and also of other priority naturally-scientific directions of modern academic researches and teaching of computer science and mathematical disciplines in universities and colleges. The book demonstrates the use of package Maple for solution of a whole series of important physical and engineering problems by means of the finite element method and method of the characteristic functions in combination with functional facilities of Maple.

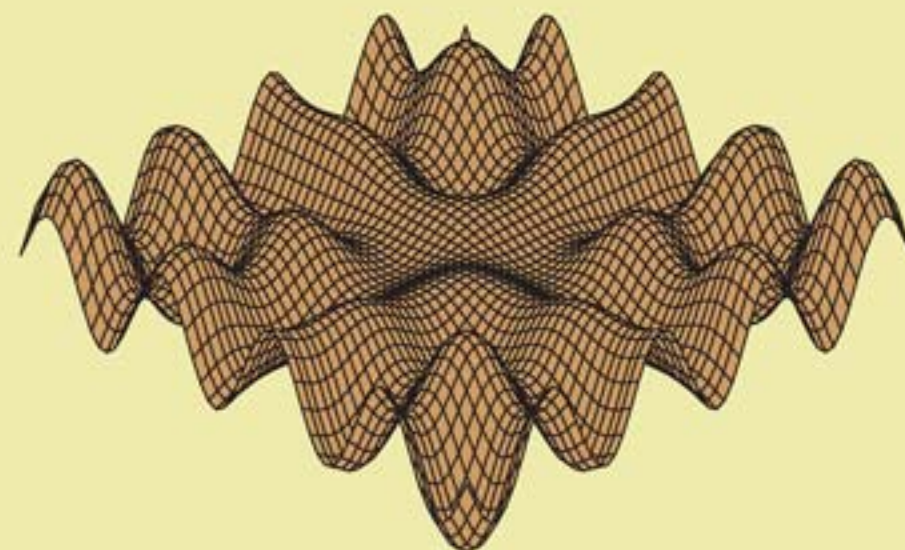
Published by Fultus Corporation
www.fultus.com



V.Z. Aladjev and M.A. Bogdevicius

Maple: Programming, Physical and Engineering Problems

Maple: Programming, Physical and Engineering Problems



V.Z. Aladjev
M.A. Bogdevicius



Maple: Programming, Physical and Engineering Problems

by

Aladjev V.Z. and Bogdevicius MA.

ISBN 1-59682-080-2

Copyright © 2006 by Aladjev V.Z. and Bogdevicius MA.
All rights reserved.



Published by Fultus Publishing

Publisher Web Site: www.fultus.com

Fultus eLibrary: elibrary.fultus.com

Online Book Superstore: store.fultus.com

Writer web site: writers.fultus.com/aladjev/



No part of this book may be used or reproduced in any manner whatsoever without written permission except
in the case of brief quotations embodied in reviews and critical articles.

The author and publisher have made every effort in the preparation of this book to ensure the accuracy of the information.
However, the information contained in this book is offered without warranty, either express or implied. Neither the author nor the publisher nor any dealer
or distributor will be held liable for any damages caused or alleged to be caused either directly or indirectly by this book.

Table of Contents

| | |
|--|-----|
| Preface | 11 |
| Part1. New software for package <i>Maple</i> of releases 6 – 10 | 17 |
| Chapter 1. General purpose software..... | 19 |
| Chapter 2. Software for work with procedural and modular <i>Maple</i> objects | 33 |
| Chapter 3. Software for work with symbols, strings, lists, sets and tables | 49 |
| Chapter 4. Software to support bit-by-bit processing of symbolic information | 55 |
| Chapter 5. Software that extends and improves the standard tools of <i>Maple</i> | 62 |
| Chapter 6. Software of working with <i>Maple</i> datafiles and documents..... | 75 |
| 6.1. General purpose software | 75 |
| 6.2. Software for operation with BINARY datafiles | 82 |
| 6.3. Software for operation with <i>Maple</i> files | 84 |
| Chapter 7. Certain useful tools for work in <i>Maple</i> | 87 |
| Part2. Application of <i>Maple</i> for solution of engineering-physical problems | 112 |
| Chapter 8. The base problems of thermal conduction | 113 |
| 8.1. Linear stationary problem of thermal conduction | 113 |
| 8.1.1. Calculated equations of the linear stationary process of heat exchange..... | 113 |
| 8.1.2. Input data for the solution of the problem..... | 117 |
| 8.1.3. Brief description of Heat_st_linear program to solve the problem | 119 |
| 8.1.4. An example of use of <i>Maple</i> -program Heat_st_linear | 119 |
| 8.2. Nonlinear stationary problem of thermal conduction | 122 |
| 8.2.1. Calculated equations of the nonlinear stationary process of heat exchange..... | 122 |
| 8.2.2. Input data for the solution of the problem..... | 124 |
| 8.2.3. Brief description of the Heat_st_nonlinear program solving the problem | 126 |
| 8.2.4. An example of use of <i>Maple</i> -program Heat_st_nonlinear | 127 |
| 8.3. Linear non-stationary problem of thermal conduction..... | 129 |
| 8.3.1. Calculated equations of a linear non-stationary process of heat exchange..... | 130 |
| 8.3.2. Input data for the solution of the problem..... | 131 |
| 8.3.3. Brief description of Heat_nonst_linear program solving the problem..... | 133 |
| 8.3.4. An example of use of <i>Maple</i> -program Heat_nonst_linear | 134 |
| 8.4. Nonlinear non-stationary problem of thermal conduction | 137 |
| 8.4.1. Calculated equations of nonlinear non-stationary process of heat exchange..... | 137 |
| 8.4.2. Input data for the solution of the problem..... | 138 |
| 8.4.3. Brief description of Heat_nonst_nonlinear program solving the problem..... | 140 |
| 8.4.4. An example of the use of <i>Maple</i> -program Heat_nonst_nonlinear | 140 |

| | |
|---|------------|
| Chapter 9. Basic problems of the elasticity theory | 144 |
| 9.1. Definition of the geometrical characteristics of plane sections..... | 144 |
| 9.1.1. Calculated expressions for definition of the geometrical characteristics of plane sections | 144 |
| 9.1.2. Input data for the solution of the problem..... | 147 |
| 9.1.3. Brief description of the Geometry program solving the problem | 148 |
| 9.1.4. An example of use of the <i>Maple</i> -program Geometry | 148 |
| 9.2. Computer-based calculation of the beam systems | 149 |
| 9.2.1. Calculated equations of the beam systems | 149 |
| 9.2.2. Input data for the solution of the problem..... | 152 |
| 9.2.3. Brief description of the Strypas program solving the problem..... | 154 |
| 9.2.4. An example of use of the <i>Maple</i> -program Strypas | 154 |
| 9.3. Plane problem of the elasticity theory | 156 |
| 9.3.1. The calculated equations of a plane problem of the elasticity theory..... | 156 |
| 9.3.2. Input data for the solution of the problem..... | 160 |
| 9.3.3. Brief description of the Plane program solving the problem | 162 |
| 9.3.4. An example of use of the <i>Maple</i> -program Plane..... | 162 |
| 9.4. Problem of contact of the elastic bodies | 164 |
| 9.4.1. The calculated equations of the problem of contact of elastic bodies | 164 |
| 9.4.2. Input data for the solution of the problem..... | 167 |
| 9.4.3. Brief description of the Contact program solving the problem | 169 |
| 9.4.4. An example of use of the <i>Maple</i> -program Contact | 169 |
| Chapter 10. Dynamic problems of the theory of elasticity..... | 173 |
| 10.1. The dynamic problem of beam systems..... | 173 |
| 10.1.1. The calculated equations of beam system's dynamics..... | 173 |
| 10.1.2. Input data for the solution of the problem..... | 176 |
| 10.1.3. Brief description of the Dynamic_strypas program solving the problem | 178 |
| 10.1.4. An example of the use of <i>Maple</i> -program Dynamic_strypas..... | 178 |
| 10.2. The plane problem of the theory of elasticity at dynamic loads | 181 |
| 10.2.1. The calculated equations of the theory of elasticity problem at dynamic loads..... | 182 |
| 10.2.2. Input data for the solution of the problem..... | 183 |
| 10.2.3. Brief description of the Dynamic_plane program solving the problem | 185 |
| 10.2.4. An example of the use of the <i>Maple</i> -program Dynamic_plane..... | 186 |
| 10.3. Dynamic problem of shells of an arbitrary contour | 190 |
| 10.3.1. The calculated equations of the shells' problem | 190 |
| 10.3.2. Input data for the solution of the problem..... | 194 |
| 10.3.3. Brief description of the Dynamic_shell program solving the problem | 197 |
| 10.3.4. An example of use of the <i>Maple</i> -program Dynamic_shell | 197 |
| 10.4. Geometrically nonlinear problem of the theory of elasticity at dynamic loadings | 202 |
| 10.4.1. The calculated equations of the nonlinear problem..... | 202 |
| 10.4.2. Input data for the solution of the problem..... | 207 |
| 10.4.3. Brief description of the Dynamic_body program solving the problem..... | 208 |

| | |
|--|-----|
| 10.4.4. An example of use of the <i>Maple</i> -program <code>Dynamic_body</code> | 209 |
| Chapter 11. Basic hydromechanics problems | 210 |
| 11.1. Nonvortical motion of a fluids, described by potential of velocities..... | 210 |
| 11.1.1. The calculated equations for potential motion of a fluid | 210 |
| 11.1.2. Input data for the solution of the problem..... | 214 |
| 11.1.3. Brief description of the <code>Pot_flow</code> program solving the problem | 215 |
| 11.1.4. An example of use of the <i>Maple</i> -program <code>Pot_flow</code> | 216 |
| 11.2. Nonvortical motion of a fluid described by the stream function..... | 221 |
| 11.2.1. The calculated equations for motion of a fluid as the stream function | 221 |
| 11.2.2. Input data for the solution of the problem..... | 223 |
| 11.2.3. Brief description of the <code>Stream_flow</code> program solving the problem | 225 |
| 11.2.4. An example of use of the <i>Maple</i> -program <code>Stream_flow</code> | 225 |
| 11.3. The Navier-Stokes's equations represented in the terms of velocity-pressure | 229 |
| 11.3.1. The calculated equations of Navier-Stokes..... | 229 |
| 11.3.2. Input data for solution of the problem | 232 |
| 11.3.3. Brief description of the <code>Navier_Stokes</code> program solving the problem | 233 |
| 11.3.4. An example of use of the <i>Maple</i> -program <code>Navier_Stokes</code> | 234 |
| 11.4. Solution of the Navier-Stokes's equations, written in the terms of variables of the vortex and stream function | 237 |
| 11.4.1. The calculated equations of Navier-Stokes in variables of vortex and stream function | 237 |
| 11.4.2. Input data for the solution of the problem..... | 239 |
| 11.4.3. Brief description of the <code>Vort_Stream</code> program solving the problem | 242 |
| 11.4.4. An example of use of the <i>Maple</i> -program <code>Vort_Stream</code> | 242 |
| 11.5. Non-stationary motion of compressible fluid, described by potential of velocity..... | 245 |
| 11.5.1. The calculated equations of non-stationary potential motion of a fluid | 245 |
| 11.5.2. Input data for the solution of the problem..... | 247 |
| 11.5.3. Brief description of the <code>Dpot_flow</code> program solving the problem..... | 249 |
| 11.5.4. An example of use of the <i>Maple</i> -program <code>Dpot_flow</code> | 249 |
| Chapter 12. Some applied problems of hydromechanics | 252 |
| 12.1. Equation of Reynolds for a layer of lubrication | 252 |
| 12.1.1. The calculated equations of Reynolds..... | 252 |
| 12.1.2. Input data for the solution of the problem..... | 255 |
| 12.1.3. Brief description of the <code>Reynold</code> program solving the problem | 256 |
| 12.1.4. An example of use of the <i>Maple</i> -program <code>Reynold</code> | 256 |
| 12.2. A model of motion of a non-Newtonian fluid | 259 |
| 12.2.1. The calculated equations of motion of a non-Newtonian fluid | 260 |
| 12.2.2. Input data for the solution of the problem..... | 263 |
| 12.2.3. Brief description of the <code>NonNiuton</code> program solving the problem..... | 265 |
| 12.2.4. An example of use of the <i>Maple</i> -program <code>NonNiuton</code> | 265 |
| 12.3. A model of convective heat exchange in a fluid..... | 268 |
| 12.3.1. The calculated equations of nonisothermal convective heat exchange | 268 |

| | |
|---|------------|
| 12.3.2. Input data for the solution of the problem..... | 271 |
| 12.3.3. Brief description of the Heat_flow program solving the problem | 274 |
| 12.3.4. An example of use of the Maple-program Heat_flow | 274 |
| 12.4. Problem of heat exchange and and mass transfer for an incompressible fluid | 278 |
| 12.4.1. The calculated equations of heat exchange and mass transfer..... | 278 |
| 12.4.2. Input data for the solution of the problem..... | 280 |
| 12.4.3. Brief description of the Heat_mass_flow program solving the problem | 284 |
| 12.4.4. An example of use of the Maple-program Heat_mass_flow..... | 284 |
| 12.5. Models of motion of a fluid in a hydraulic system | 288 |
| 12.5.1. Motion of a compressible fluid in a hydraulic system..... | 288 |
| 12.5.2. Input data for the solution of the problem..... | 294 |
| 12.5.3. Brief description of the Hydro program solving the problem | 295 |
| 12.5.4. An example of use of the Maple-program Hydro..... | 296 |
| Chapter 13. Applied problems of mechanics - 1..... | 298 |
| 13.1. Calculation of moments of inertia of a solid body..... | 298 |
| 13.1.1. Calculated expressions for definition of moments of inertia of a solid body..... | 298 |
| 13.1.2. Input data for the solution of the problem..... | 300 |
| 13.1.3. Brief description of the Mass_inertia program solving the problem | 301 |
| 13.1.4. An example of use of the Maple-program Mass_inertia | 301 |
| 13.2. Calculation of inertia moments of a system of solid bodies | 303 |
| 13.2.1. The calculated expressions for determination of inertia moments..... | 304 |
| 13.2.2. Input data for the solution of the problem..... | 305 |
| 13.2.3. Brief description of the Mass_system program solving the problem..... | 306 |
| 13.2.4. An example of use of the Maple-program Mass_system | 306 |
| 13.3. Nonlinear oscillations of mechanical systems..... | 308 |
| 13.3.1. The calculated expressions for description of nonlinear oscillations..... | 308 |
| 13.3.2. Input data for the solution of the problem..... | 310 |
| 13.3.3. Brief description of the Amplitude program solving the problem..... | 311 |
| 13.3.4. An example of use of the Maple-program Amplitude..... | 311 |
| 13.4. Derivation and solution of equations of motion of a mechanical system with the concentrated parameters..... | 313 |
| 13.4.1. The calculated expressions for deduction of equations of motion of a mechanical system | 313 |
| 13.4.2. Input data for the solution of the problem..... | 317 |
| 13.4.3. Brief description of the Pendium program solving the problem | 317 |
| 13.4.4. An example of use of the Maple-program Pendium | 317 |
| 13.5. Calculation of transition torsion oscillations in a mechanical transmission | 320 |
| 13.5.1. The calculated expressions for determination of torsional oscillations | 320 |
| 13.5.2. Input data for the solution of the problem..... | 323 |
| 13.5.3. Brief description of the Driver program solving the problem | 325 |
| 13.5.4. An example of use of the Maple-program Driver | 325 |

| | |
|--|------------|
| Chapter 14. The applied problems of mechanics - 2 | 328 |
| 14.1. Motion of a vehicle on a rough road | 328 |
| 14.1.1. The calculated expressions for definition of motion of a vehicle | 328 |
| 14.1.2. Input data for the solution of the problem | 331 |
| 14.1.3. Brief description of the Trailer program solving the problem | 332 |
| 14.1.4. An example of use of the <i>Maple</i> -program Trailer | 332 |
| 14.2. Stationary random oscillations of a vehicle | 335 |
| 14.2.1. The calculated expressions for determination of motion of a vehicle | 335 |
| 14.2.2. Input data for the solution of the problem | 337 |
| 14.2.3. Brief description of the Trailer_random program solving the problem | 338 |
| 14.2.4. An example of use of the <i>Maple</i> -program Trailer_random | 339 |
| 14.3. Motion of a carriage along a rough railway | 340 |
| 14.3.1. The calculated expressions for determination of a carriage motion | 340 |
| 14.3.2. Input data for the solution of the problem | 342 |
| 14.3.3. Brief description of the Carriage program solving the problem | 343 |
| 14.3.4. An example of use of the <i>Maple</i> -program Carriage | 344 |
| 14.4. Dynamics of pneumatic vibration-extinguisher with stationary magnets | 345 |
| 14.4.1. Equation of a vibration-extinguisher motion with stationary magnets | 346 |
| 14.4.2. Input data for the solution of the problem | 348 |
| 14.4.3. Brief description of the Damper program solving the problem | 348 |
| 14.4.4. An example of use of the <i>Maple</i> -program Damper | 349 |
| 14.5. Models of gas motion in a pneumatic system | 351 |
| 14.5.1. Motion of gas in a pneumatic system | 352 |
| 14.5.2. Input data for the solution of the problem | 357 |
| 14.5.3. Brief description of the Pneumo program solving the problem | 359 |
| 14.5.4. An example of use of the <i>Maple</i> -program Pneumo | 359 |
| Chapter 15. Application of <i>Maple 6</i> for solution of optimization problems | 361 |
| 15.1. Search methods for unconstrained optimization | 361 |
| 15.1.1. Nelder and Mead's method | 362 |
| 15.1.1.1. Input data for the solution of the problem | 364 |
| 15.1.1.2. Brief description of the Nelder_Mead program solving the problem | 364 |
| 15.1.1.3. An example of use of the <i>Maple</i> -program Nelder_Mead | 364 |
| 15.1.2. Hooke and Jeeves' method | 364 |
| 15.1.2.1. Input data for the solution of the problem | 365 |
| 15.1.2.2. Brief description of the Hooke_Jeeves program solving the optimization problem | 366 |
| 15.1.2.3. An example of use of the <i>Maple</i> -program Hooke_Jeeves | 366 |
| 15.1.3. Davidon's cubic interpolation method | 366 |
| 15.1.3.1. Input data for the solution of the problem | 368 |
| 15.1.3.2. Brief description of the Cubic_interpolation solving the optimization problem | 369 |
| 15.1.3.3. An example of use of the <i>Maple</i> -program Cubic_interpolation | 369 |
| 15.2. Gradient methods for unconstrained optimization | 369 |

| | |
|--|-----|
| 15.2.1. Gradient method | 369 |
| 15.2.1.1. Input data for the solution of the problem | 370 |
| 15.2.1.2. Brief description of the Gradient program solving the problem | 370 |
| 15.2.1.3. An example of use of the <i>Maple</i> -program Gradient | 370 |
| 15.2.2. Fletcher-Reeves and Polak-Ribiere method | 371 |
| 15.2.2.1. Input data for the solution of the problem | 372 |
| 15.2.2.2. Brief description of the Fletcher_Reeves_Polak_Ribiere program solving the problem | 372 |
| 15.2.2.3. An example of use of the <i>Maple</i> -program Fletcher-Reeves-Polak-Ribiere | 372 |
| 15.2.3. Newton's methods | 373 |
| 15.2.3.1. Input data for the solution of the problem | 374 |
| 15.2.3.2. Brief description of the Newtons_method program solving the optimization problem | 375 |
| 15.2.3.3. An example of use of the <i>Maple</i> -program Newtons_method | 375 |
| 15.3. Quasi-Newton methods | 375 |
| 15.3.1. Simple rank procedure | 375 |
| 15.3.1.1. Input data for the solution of the problem | 377 |
| 15.3.1.2. Brief description of the QUASI_SIMPLE program solving the optimization problem | 377 |
| 15.3.1.3. An example of use of the <i>Maple</i> -program QUASI_SIMPLE | 377 |
| 15.3.2. Davidon-Fletcher-Powell method | 378 |
| 15.3.2.1. Algorithm of Davidon-Fletcher-Powell method | 378 |
| 15.3.2.2. Input data for the solution of the problem | 378 |
| 15.3.2.3. Brief description of the Davidon_Fletcher_Powell program solving the problem | 379 |
| 15.3.2.4. An example of use of the <i>Maple</i> -program Davidon_Fletcher_Powell | 379 |
| 15.3.3. Broyden-Fletcher-Goldfarb-Shanno method | 379 |
| 15.3.3.1. Algorithm of Broyden-Fletcher-Goldfarb-Shanno method | 379 |
| 15.3.3.2. Input data for the solution of the problem | 380 |
| 15.3.3.3. Brief description of the BFGS program solving the optimization problem | 380 |
| 15.3.3.4. An example of use of the <i>Maple</i> -program BFGS | 380 |
| 15.3.4. Memoryless quasi-Newton method | 380 |
| 15.3.4.1. Algorithm of memoryless quasi-Newton method | 380 |
| 15.3.4.2. Input data for the solution of the problem | 381 |
| 15.3.4.3. Brief description of the Memoryless_QN program solving the optimization problem | 381 |
| 15.3.4.4. An example of use of the <i>Maple</i> -program Memoryless_QN | 381 |
| 15.4. Constrained minimization methods | 382 |
| 15.4.1. Penalty-function method | 382 |
| 15.4.1.1. Fiacco and McCormick method | 382 |
| 15.4.1.2. Input data for the solution of the problem | 383 |
| 15.4.1.3. Brief description of the Fiacco_McCorm. program solving the optimization problem | 384 |
| 15.4.1.4. An example of use of the <i>Maple</i> -program Fiacco_McCormick | 384 |
| 15.4.2. Complex method | 384 |
| 15.4.2.1. Algorithm of Complex method | 385 |
| 15.4.2.2. Input data for the solution of the problem | 385 |

| | |
|---|-----|
| 15.4.2.3. Brief description of the Complex program solving the optimization problem | 385 |
| 15.4.2.4. An example of use of the <i>Maple</i> -program Complex | 386 |
| 15.4.3. Methods based on the sequence of linear programs. The first method | 386 |
| 15.4.3.1. Cutting plane method | 387 |
| 15.4.3.2. Input data for the solution of the problem | 387 |
| 15.4.3.3. Brief description of the Cutting_Plane program solving the optimization problem | 388 |
| 15.4.3.4. An example of use of the <i>Maple</i> -program Cutting_Plane | 388 |
| 15.4.4. The method based on the sequence of linear programs. The second method | 389 |
| 15.4.4.1. Method of approximate programming | 389 |
| 15.4.4.2. Input data for the solution of the problem | 389 |
| 15.4.4.3. Brief description of the SLP program solving the optimization problem | 390 |
| 15.4.4.4. An example of use of the <i>Maple</i> -program SLP | 390 |
| 15.5. Genetic algorithms | 390 |
| 15.5.1. The procedure of genetic algorithm | 392 |
| 15.5.2. Input data for the solution of the problem | 392 |
| 15.5.3. Brief description of the Genetic_dalia program solving the optimization problem | 393 |
| 15.5.4. An example of use of the <i>Maple</i> -program Genetic_dalia | 393 |
| References | 394 |
| The information about the writers | 402 |

Preface

Computer Algebra (also known as *Symbolic Computation* or *Computational Algebra*) has found application in many fields of science such as mathematics, physics, chemistry, computer science, engineering, education, technology, computational biology, etc. The *computer algebra systems (CAS)* such as *Maple*, *Reduce*, *MuPAD*, *Axiom*, *Macsyma*, *Mathematica*, *Derive*, *Magma*, and others are becoming more and more popular in teaching, research and industry. The area of symbolic and algebraic computation aims at automation of mathematical computations of all sorts. The resulting computer systems, both experimental and commercial, are powerful tools for scientists, engineers, and educators. This research combines mathematics with advanced computing techniques. **CAS** is an interdisciplinary area between *Mathematics* and *Computer Science*. Its research focuses on the development of algorithms for performing *symbolic* manipulations with algebraic objects on computers, and design of programming languages and environments for implementing these algorithms.

In a series of our books and papers [28-30, 32, 33, 41, 111-116, 120-122, 124-127, 235-247], such packages as *Maple*, *Reduce*, *MathCAD* and *Mathematica* have been considered. Our experience in the detailed testing and practical use of different mathematical and physical applications of four mathematical packages (*Reduce*, *Maple*, *MathCAD* and *Mathematica*) enables us to consider the packages *Maple* and *Mathematica* as undoubted leaders (*on the basis of a generalized index*) among all listed modern tools of computer algebra. Meanwhile, we give preference to the package *Maple* due to a number of strong reasons described in the above books enough in detail.

Computer algebra becomes a rather powerful and useful tool for scientists and experts from various fields. However, manifold applications demand that the essential significant numerical calculations be combined with algebraic ones. With each new release, the package *Maple* meets more and more requirements. The *Maple* package has been widely used not only as a tool of solving mathematical problems. The package allows us to revise approaches to teaching subjects related to mathematics in universities by defining in many cases the methods for a teaching of subjects with the use of PCs to solve mathematical problems for various purposes [111-116, 120-122, 124-127, 235-247].

Researchers use well-known *Maple* package as an essential tool when solving problems related to their investigation. The package is ideal for formulating, solving, and exploring different mathematical models. Its symbolic manipulation facilities extend greatly over a range of problems that can be solved with its help. Educators in high schools, colleges, and universities have revitalized traditional curricula by introducing problems and exercises, which use *Maple's* interactive mathematics and physics. Students can concentrate on the more fundamental concepts rather than on tedious algebraic manipulations. Finally, engineers and experts in industries use *Maple* as an efficient tool replacing many traditional resources such as reference books, calculators, spreadsheets and programming languages. These users easily solve mathematical problems, creating projects and consolidating their computations into professional reports.

Maple products embody advanced technologies such as symbolic computation, infinite precision numeric, innovative *Web* components, extensible user-interface technology, and an unrivalled suite of mathematical algorithms for intelligent management of complex mathematics. Over 3 million users benefit from advanced *Maple* technology. Virtually, all major universities and research

institutes in the world, including such as MIT, Oxford, Stanford and Waterloo, has adopted *Maple* products to enhance their education and research activities. Waterloo *Maple's* industrial customer base includes Boeing, Bosch, Canon, NASA, etc.

Meanwhile, our operational experience in the period of 1997 – 2005 with *Maple* of releases 4, 5, 6, 7, 8, 9/9.5 and 10 enabled us not only to estimate its advantages compared with other similar mathematical packages, but has also revealed a number of faults and shortcomings which were eliminated by us. Furthermore, *Maple* does not support a number of important procedures of information processing, *symbolic* and *numeric* computing, including the tools of access to datafiles. By operating *Maple* we have developed rather effective tools (*procedures* and *program modules*), largely extending the possibilities of the package. This software has been organized as a *Library* that is structurally similar to the main *Maple* library, and is provided with a rather detailed Help system analogous to *Maple* Help system.

In particular, *Maple* does not provide sufficient compatibility of releases 6-10. This fact and the incompatibility of the package found out by us at a level of base platforms – *Windows* 98SE and lower, on the one hand, and *Windows* XP and above, on the other hand, the decision of a compatibility problem for the *Library* means have demanded.

The tools represented in the *Library* increase the range and efficiency of use of the package on the *Windows* platform owing to the innovations in three basic directions: (1) *elimination of a series of basic defects and shortcomings*, (2) *extension of capabilities of a series of standard tools*, and (3) *replenishment of the package by new means which increase the capabilities of its program environment, including the facilities improving the compatibility of releases 6, 7, 8, 9/9.5 and 10*. The basic attention is devoted to additional tools created in the process of practical use and testing of the package of releases 4-8 which by some parameters considerably extend the capabilities of the package making the work with it much easier; a considerable attention is also devoted to the tools providing package compatibility of releases 6, 7, 8, 9/9.5 and 10. The experience in using the above software for various applications has confirmed its valuable operational characteristics.

It should be noted that a series of our books and papers on *Maple* problems [111-116, 120-122, 124-127, 235-247], representing tools developed by the authors and containing suggestions on further development of the package encouraged the development of such applications as package modules **FileTools**, **LibraryTools**, **ListTools** and **StringTools**. However, the means suggested by us essentially extend the capabilities of the package, which in many cases exceed those of the specified modules.

The above software has been organized into the user *Library*, whose the current version contains tools (*more than 640 procedures and program modules*) which are oriented to a wide area of computing and information processing. The *Library* is structurally similar to the main *Maple* library and is supplied with the advanced Help system about the tools located in it. In addition, it is logically connected with the main *Maple* library, providing access to the tools contained in it similarly to the package tools. The simple guide describes the installation of the *Library* at presence in the PC with the above-mentioned *Windows* platform of the installed *Maple* package of releases 6, 7, 8, 9/9.5 or 10.

Taking into account our long-term experience in operation with the *Maple* package of releases 4-10 and experience of our colleagues from universities and the academic institutes of Lithuania, Latvia, Belarus, Estonia and Russia, it is should be noted, that many of tools (*or their analogues*) of our *Library* are worth to be included into standard deliveries of subsequent *Maple* releases. At present, they are accessible to the *Maple* users as the offered *Library* supporting releases 6-10 and functioning on platforms *Windows* 95 and later. *Library* tools in many cases allow us to facilitate programming of various applied problems in the *Maple* environment of releases 6-10.

It is possible to state, that a series of our books on the *Maple* problems [120-122, 124-127, 235-247] that represent the means developed by us and contain useful tips on the further development of the package, has encouraged the development of package modules **FileTools**, **LibraryTools**, **ListTools** and **StringTools**. However, in this respect tools represented by us essentially extend capabilities of the package, exceeding those of the specified package modules in many cases.

The *Library* is designed for a wide audience of experts, teachers, post-graduates and students of natural-science professions who use *Maple* of releases 6-10 on *Windows* platform in their own professional work. The *Library* contains well-designed software (*a set of procedures and program modules*), which supplements well the already available *Maple* software with the orientation towards the widest circle of the *Maple* users, greatly enhancing its usability and effectiveness. Our experience reveals that the use of *Library* provides more opportunities of *Maple* of releases 6-10, simplifying the programming of various practical problems in its environment. This *Library* will be of special interest above all to those *who* use *Maple* not only as a highly intellectual calculator but also as environment for programming of different problems in their professional activities. The *Library* has been rewarded by "Smart Award" from Smart DownLoads Network.

Furthermore, the presence in the *Library* delivery of the text datafile "ProcUser.txt" with source codes of the *Library* software and *mws*-files with help-pages composing Help database of the *Library* allows us to adapt it to other underlying platforms different from *Windows*. Furthermore, the source codes, using both the effective and the non-standard technique, can serve as an useful enough practical programming guide on the *Maple* language. Archive with this *Library* and accompanying materials you may free-of-charge download from the web-site

Briefly, we shall present the contents of separate chapters of the book. The book consists of two parts: (1) new software for package *Maple* of releases 6-10 and (2) application of *Maple* for solution of physical & engineering problems. In the *first* chapter, the software providing the most general procedures of operation with package *Maple* of releases 6-10 is represented. Here and further every procedure is being represented in the following aspects: (1) *procedure's name with its brief characteristic*, (2) *procedure call sequence*, (3) *formal procedure parameters*, (4) *description of the procedure*, (5) *procedure source code*, and (6) *most typical examples of its use*. The source codes allow not only easily to immerse them into the *Maple* environment of releases 6-10 on many computer platforms, but also to use their as an useful enough illustrative material at a mastering of the advanced programming in the package environment. The full set of means of the given type is presented in our books [239, 240] and in the library [259] attached to the present book.

In the *second* chapter of the book, software for work with procedural and modular *Maple* objects are considered. These tools support kinds of processing such as: converting of modules into procedures; testing of presence in files of incorrect modules; check of parameters of procedures and modules; check of activity (*availability for direct use*) of a procedure or a module; check of type of a modular table; converting of files of the input *Maple* format containing modules; converting of a module of the second type into the first; converting of a file of the input *Maple* format into a file of the internal *Maple* format, and vice versa, etc. The represented tools provide many of the manifold useful operations with procedural and modular objects of *Maple*. The means represented in the given chapter allows to effectively program the above objects at solving your *Maple* appendices.

In the *third* chapter, software for work with symbols, strings, lists, sets and tables are considered. The means represented in the given chapter allows to effectively program your *Maple* appendices using these important *Maple* data structures. These tools provide a number of useful procedures such as special kinds of converting; comparison of strings or/and symbols; case sensitive pattern

searching; exhaustive substitutions into strings or symbols; inversion of symbols, strings or lists; reducing of multiplicity of entries of a symbol into a string; identification of entries of special symbols into a string; and others. The list structures play an extremely important role, defining the ordered sequences of elements. A series of procedures of the section supports useful kinds of processing such as: a special converting of lists into sets, and vice versa; operation with rarefied lists; dynamic assignment of values to elements of a list or a set; evaluation of indices of a table over its entry; representation of a special type of tables; special kinds of exhaustive substitutions into lists or sets; a series of important kinds of sorting of nested lists, and also many others. In a series of cases, these tools simplify operation with *Maple* objects of type $\{string, symbol, list, set, table\}$ in the *Maple* environment. The full set of means of the given type is presented in our books [239, 240] and in the library [259] attached to the present book.

In the *fourth* chapter, software to support bit-by-bit processing of symbolic information are considered. The package does not possess tools of the similar type. The software offered by us is represented by six useful procedures such as *Bit*, *Bit1*, *xbyte*, *xbyte1*, *xNB* and *xpack*. These procedures serve for *bit-by-bit* information processing, i.e. the user has possibility to operate with strings or symbols at a level of separate bits composing them.

In the *fifth* chapter, the tools that extend and improve the standard *Maple* tools for releases 6-10 are represented. These tools are used enough widely both at operation with the *Maple* package in interactive mode and at programming of various problems in its environment. They represent undoubted interest at programming of various problems in the *Maple* environment, both by simplifying the programming and by making it by more clear.

The *sixth* chapter, software for work with *Maple* datafiles and documents are considered. Being the programming language in the package environment, oriented, first, to symbolic calculations (*computer algebra*) the *Maple* language has the relatively limited opportunities at operation with data that are located in external computer memory. Moreover, in this respect the *Maple* language essentially yields to traditional programming languages such as C, COBOL, FORTRAN, PL/1, Pascal, ADA, Basic, etc. At the same time, the *Maple* language, oriented, first of all, onto solution of problems of mathematical character, gives a set of tools for access to datafiles which can quite satisfy a broad enough audience of users of physical and mathematical appendices of the package. In the present chapter, additional means of access to datafiles are represented, essentially extending opportunities of the package in the given direction. Many of them simplify the programming of many problems dealing with access to datafiles of different purpose. With all evidence we can assert, that new package modules **FileTools** and **LibraryTools** have been inspired by a series of our works [124-147, 235-247] with which the *Maple* developers have been acquainted. However, our set of the similar procedures is considerably more representative and is focused on wider practical use.

The *seventh* chapter represents certain useful tools for work in *Maple*. The full set of means of the given type is presented in our books [239, 240] and in the library [259] attached to the present book.

From the *eighth* chapter, the book represents applied aspects of package *Maple* by an example of solution of a wide circle of engineering-physical problems from the following fields: classical mechanics, hydrodynamics, hydromechanics, heating engineering, etc. Each problem is represented in the following context: (1) a common theoretical part (*setting of a problem*), (2) initial data for solution of the problem, (3) brief description of *Maple*-program solving the problem, and (4) examples of application of the given program for the solution of concrete application with interpretation of obtained outcomes. Given that, source codes of the debugged *Maple*-programs with the data for test examples, which are appropriated to them, are represented in the archive attached to the present book. Similar organization of the material allows the reader to apply software immediately, which considered in the book, in the own professional activity. Archive with

these codes, test examples, test data and other accompanying materials you may free-of-charge download from the web-site

The presented programs can serve as a final software product and as good material illustrating main functionalities of the package *Maple*. The material of the second part of the book is based on our previous books [124,125] and the outcomes of adapting engineering-physical problems are considered relative to releases 6-10 of *Maple*. Let us study the contents of the last seven chapters of the book, which compose its second part of the applied assignment.

The *eighth* chapter of the book considers questions of computer solution in the environment of package *Maple* of the basic problems of thermal conductivity having important value to solve many applied problems of thermal physics as well as a combination of problems in the theory of elasticity and plasticity. Solution in the environment of the package of the following problems is considered: linear (1) and nonlinear (2) stationary problems of thermal conductivity as well as linear (3) and nonlinear (4) non-stationary problems of thermal conductivity. In particular, linear non-stationary heat conductivity equation is one of the main equations describing convective heat exchange and mass exchange occurring in systems of various physical nature. Whereas the last problem has not only major independent value for the definition of non-stationary temperature fields in various matters and materials, but is an important component of more complex engineering-physical problems which have rather wide engineering-physical applications as well.

In the *ninth* chapter, four problems of linear mechanics of deformable bodies are considered. The main relations concerning the description of tensely-deformable state and the relations for resilient bodies are given. Each problem is solved by the finite element method. The main principles of solution of the next practically important problems are represented: (1) definition of geometrical parameters of the cross-sections of bodies, (2) calculation of rod constructions at static loads, (3) the plane problem of the theory of elasticity, and (4) a contact problem of two elastic bodies. The indicated problems represent an applied interest; therefore *Maple*-programs corresponding to them on a rather wide circle of the applications are oriented.

In the *tenth* chapter, the dynamic problems of the theory of elasticity representing major practical interest at designing many objects of mechanical engineering are considered. The main problems here are definition of own and forced oscillations of elastic bodies, and also research of behaviour of elastic systems at short-term loads. The problems, considered in the chapter, represent practical interest for research of the indicated questions. The main principles of solution of the next practically important problems are represented: (1) research of own and forced oscillations of linear rod systems, (2) the plane problem of the theory of elasticity at dynamical loads, (3) a dynamical problem of shells of arbitrary configuration, and (4) a geometrically nonlinear problem of the theory of elasticity at dynamical loads. The indicated problems represent an applied interest; therefore, *Maple*-programs corresponding to them on a rather wide circle of the applications are oriented.

In the *eleventh* chapter, five problems of hydrodynamics having the important applied value and allowing to solve many engineering problems, which are linked with the flow around bodies by a liquid, oscillations of constructions in a liquid, a percussion impulse action onto a liquid, etc. are considered. The following problems are represented: an vortex-free motion of a liquid in terms of potential of speeds (1) and the flow function (2); (3) the equations of Navier-Stokes describing common movement of a liquid; (4) a solution of Navier-Stokes equations in the terms of a vortical function of flow, widely used in applied problems of hydromechanics, and (5) a non-stationary movement of a compressed liquid, described by the potential of speeds.

In the *twelfth* chapter, some the following problems of hydromechanics, which play an important role in research of problems are considered: (1) hydrodynamic lubrication between movable surfaces, (2) models of movement of a non-Newtonian liquid, (3) the problem of convective heat exchange and mass transfer for an incompressible liquid, (4) a model of convective heat exchange in a liquid, and (5) the models of movement of a liquid in a hydraulic system. All these problems play an important role in many technical applications, above all, of hydro-mechanical character. The finite element method adapted for the environment of *Maple*-language of the package appears as the main tool of the solution of engineering-physical problems is represented in the book.

When solving dynamic problems of solid mechanics, there is a necessity of definition of moments of inertia of masses of bodies of complex geometrical forms as well as systems of bodies, with respect to various axes of coordinates. For example, the *thirteenth* chapter considers the problems on definition of moments of inertia of masses of a body with a complex geometrical form and of systems of such bodies. By the facilities of package *Maple*, the following interesting problems are studied: (1) calculation of moments of inertia of a solid; (2) a calculation of moments of inertia of the system of solids; (3) the nonlinear oscillations of mechanical systems; (4) derivation and solution of equations of motion of a mechanical system with concentrated parameters, and (5) calculation of transitional torsional oscillations in mechanical transmission.

The problems of moving of transport along uneven way are of great interest. When transport move along uneven way upon elements of its hanger bracket a rather large loads affect, which in turn, decrease a longevity of elements of hanger bracket and all carrier as a whole, and also essentially influence onto its comfortable quality. In this connection, the *fourteenth* chapter of the book considers the following interesting problems: (1) movements of a carrier on an uneven way; (2) the stationary random oscillations of a carrier; (3) movements of a car on an uneven railway path; (4) dynamics of pneumatic vibration extinguisher with steady magnets, and (5) the models of moving gas in the pneumatic system.

At last, the *fifteenth* chapter is devoted to application of *Maple* for solution of certain optimization problems. In this chapter, the problem of minimizing function of real variables under the assumption that no constraint is imposed on the values of these variables is considered in the light of different standpoints.

This book represents the extended presentation of materials of lectures on the fundamentals of work in the environment of mathematical package *Maple*, which were given by the authors in 2001-2004 for the researchers of the Baltic Branch of the International Academy of Noosphere (Tallinn, Estonia), Vilnius Gediminas Technical University (Vilnius, Lithuania), Grodno State University and Grodno branch of Institute of Modern Knowledge (Grodno, Byelorussia). The given courses aimed at the introduction of innovations of the new software for *Maple* with orientation, above all, on the experts in the field of physical and mathematical sciences and also of other priority naturally-scientific directions of modern academic researches and teaching of computer science and mathematical disciplines in universities and colleges.

In view of mathematical orientation of package *Maple* and widely used in this connexion of mathematical terminology and notation, at presentation of all next material we suppose the presence of a sufficient mathematical culture for the reader. Therefore, we will focus on the facilities and possibilities of the especially considered package, instead of purely mathematical subjects.

Part 1.

New software

for package *Maple* of releases 6 – 10

Chapter 4.

Software to support bit-by-bit processing of symbolic information

In this chapter, the tools, which support a *bit-by-bit* information processing in the *Maple* environment, are represented. The package does not possess tools of the similar type. The software offered by us is represented by six useful procedures such as **Bit**, **Bit1**, **xbyte**, **xbyte1**, **xNB** and **xpack**. These procedures serve for *bit-by-bit* information processing, i.e. the user has a possibility to operate with strings or symbols at a level of separate bits composing them.

Bit - *bit-by-bit information processing*

Bit1

xbyte

xbyte1

xNB

Call format of the procedures:

Bit(B, bits)

Bit1(B1, bits)

xbyte(L)

xbyte1(L1)

xNB(N)

Formal arguments of the procedures:

B - a symbol or a string of length **1**, or an integer **1**-element list

B1 - an integer from the range **0..255**

bits - a leading variable of a polynomial

L - a list from **8** binary digits

L1 - a positive binary number of length ≤ 8 , or a list from no more than **8** binary digits

N - a positive integer (*posint*)

Description of the procedures:

The tools represented below serve for a *bit-by-bit* information processing, i.e. the user has a possibility to operate with the symbols or strings on a level of separate bits composing them. The first **Bit** procedure allows effectively enough to execute *bit-by-bit* processing of separate symbols.

The procedure call **Bit(B, bits)** allows to execute the following basic *bit-by-bit* operations with a symbol, defined by the first actual **B** argument of the procedure (as argument can appear an one-element string or symbol of length **1**, and also 1-element list whose element defines a decimal code of a symbol from the range **0 .. 255**):

- (1) if the second actual *bits* argument has a view [**n1**, **n2**, ...], then the procedure call **Bit(B, bits)** returns the values of bits of symbol **B**, which are located in its **n_j** positions (**n_j** belong to the range **1..8**);

- (2) if the second actual *bits* argument has a view [$n_1=b_1, n_2=b_2, \dots$], the decimal code of a symbol, obtained by replacement of values of bits of symbol **B** (which are in its positions n_k) onto binary values b_k , is returned ($n_k = 1 \dots 8$; $b_k = \{0|1\}$).

- (3) if the second actual *bits* argument is a name of procedure defined over the lists, the result of application of the procedure to a list of values of all bits of a symbol, defined by the first actual **B** argument, is returned.

In addition, in two last cases the decimal code of a symbol - of the result of processing of the initial symbol, defined by the first actual **B** argument of the procedure, is returned.

The *Bit1(B1, bits)* procedure is a modification of the above *Bit* procedure. In the procedure the more effective algorithm of evaluations has been implemented, however a handling of special and erroneous situations are not fulfilled, and as the first actual **B1** argument only the decimal code of a symbol subjected to bit-by-bit processing is admitted. The second *bits* argument has remained the same. The given procedure is more reactive and can be successfully used in the cyclic computational constructions.

For providing of *bit-by-bit* symbolic processing two basic procedures *Bit* and *Bit1* represented above are intended. The simple procedure *xbyte(L)* fulfills an inverse function, returning a symbol with the given bit value defined by the actual **L** argument; as a value for actual **L** argument a list from 8 binary digits is used.

Provided that the arbitrary symbol "X" has a *bit-by-bit* representation **B::list(binary)**, for the procedures *xbyte* and *Bit* the following defining relations take place:

$$\text{"X"} = \text{xbyte}(\text{Bit}(\text{"X"}, [\text{k}\$\text{k}=1..8])) \quad \text{Bit}(\text{xbyte}(\text{B}), [\text{k}\$\text{k}=1..8]) = \text{B}$$

The procedure *xbyte1(L)* extends the above procedure *xbyte* onto positive binary numbers, i.e. the numbers consisting of numerals $\{0,1\}$ only; for example **11001011**. In this case we can represent a symbol in the form of such binary number of length, not greater than 8; at a smaller length the procedure supplements a *bit-by-bit* representation at the left by *zeroes*. Moreover, a binary list of length < 8 , obtained at the procedure call, is supplemented at the left by *zeroes* up to the standard list of length 8. Such approach allows in many cases more conveniently to represent the symbols at a level of bits composing them.

The simple procedure *xNB(N)* has been implemented by a one-string extracode and is useful enough at operation with information at the bit level. The procedure call *xNB(N)* returns the *true* value, if a positive integer **N** consists of binary numerals only, otherwise the *false* value is returned. The five procedures *Bit*, *Bit1*, *xbyte*, *xbyte1* and *xNB*, represented above, along with the introduced types $\{digit, binary, byte\}$ create an quite satisfactory basis for operation with symbolic information at the *bit-by-bit* level. The examples represented below illustrate use of all procedures represented above for a *bit-by-bit* information processing in the *Maple* environment.

```

Bit := proc(B::string, symbol, list(integer)), bits::procedure, list({integer, equation}))
local a, b, c, k;
  c := (x) -> [seq(parse(x[k]), k = 1 .. length(x))];
  if type(B, {'symbol', 'string'}) and length(B) = 1 then
    a := c(cat("", convert(op(convert(B, 'bytes')), 'binary'))); b := [0 $ ('k' = 1 .. 8 - nops(a)), op(a)]
  elif type(B, 'list('integer')) and nops(B) = 1 and belong(B[1], 0 .. 255) then
    a := c(cat("", convert(op(B), 'binary'))); b := [0 $ ('k' = 1 .. 8 - nops(a)), op(a)]
  else error "1st argument must be symbol/string of length 1 or integer list,
    but has received <%1>", B
  end if;
  if type(bits, 'list('integer')) and belong({op(bits)}, 1 .. 8) then [seq(b[k], k = {op(bits)})]

```

```

elif type(bits, 'list'('equation')) and belong({seq(lhs(k), k = bits)}, 1 .. 8) and
belong({seq(rhs(k), k = bits)}, 0 .. 1) then seq(assign('b'[lhs(bits[k])] = rhs(bits[k])),
    k = 1 .. nops(bits)); convert(parse(cat(op(map(convert, b, 'string'))), 'decimal', 'binary')
elif type(bits, 'procedure') then convert(parse(cat(op(map(convert, bits(b), 'string'))),
    'decimal', 'binary')
else error "2nd argument <%1> is invalid", bits
end if
end proc
Bit1 := proc(B::{integer, string, symbol}, bits::{procedure, list({integer, equation}})
local a, b, c, k;
    c := (x) -> [seq(parse(x[k]), k = 1 .. length(x))];
if type(B, {'symbol', 'string'}) then a := c(cat("", convert(op(convert(B, 'bytes')), 'binary')));
    b := [0 $ ('k' = 1 .. 8 - nops(a)), op(a)]
elif type(B, 'integer') then a := c(cat("", convert(B, 'binary'))); b := [0 $ ('k' = 1 .. 8 - nops(a)), op(a)]
else error "1st argument must be symbol/string of length 1 or integer list,
    but has received <%1>", B
end if;
if type(bits, 'list'('integer')) then [seq(b[k], k = {op(bits)})]
elif type(bits, 'list'('equation')) then seq(assign('b'[lhs(bits[k])] = rhs(bits[k])), k = 1 .. nops(bits));
    convert(parse(cat(op(map(convert, b, 'string'))), 'decimal', 'binary')
elif type(bits, 'procedure') then convert(parse(cat(op(map(convert, bits(b), 'string'))),
    'decimal', 'binary')
else error "2nd argument <%1> is invalid", bits
end if
end proc
xbyte := proc(L::{list(binary)})
option remember;
    if nops(L) <> 8 then error "argument %1 is invalid", L else
    convert([convert(SN(cat("", op(L))), 'decimal', 'binary')], 'bytes')
    end if
end proc
xbyte1 := proc(L::{nonnegint, list(binary)})
local k;
option remember;
    if type(L, 'list') then xbyte([0 $ (k = 1 .. 8 - nops(L)), op(L)])
    elif xNB(L) and length(L) <= 8 then
        xbyte([0 $ (k = 1 .. 8 - length(L)), op(map(SN, convert(cat("", L), 'list1')))]
    else error "argument <%1> is invalid", L
    end if
end proc
xNB := (N::nonnegint) -> belong({op(map(SN, convert(cat("", N), 'list1')))}, {0, 1})

```

Typical examples of the procedure use:

```

> Bit(`S`, [k$k=1..8]), Bit([61], [1, 3, 8]), Bit([61], [2=1, 4=0, 8=0]), Bit("G", [6]),
    Bit("0", [6=1, 7=1, 8=1]);

```

```

[0, 1, 0, 1, 0, 0, 1, 1], [0, 1, 1], 108, [1], 7

```

```

> R:= (L::list) -> [L[nops(L)-k]$k=0..nops(L)-1]: Bit("G", [k$k=1..8]), Bit("G", R),
  Bit("S", R), Bit([61], R);
                                [0, 1, 0, 0, 0, 1, 1, 1], 226, 202, 188
In particular, the given example illustrates use as the second actual bits argument of the R-
procedure providing the inversion of lists:
> K:= (L) -> [L[8],L[2],L[6],L[4]+1 mod 2, L[5]+1 mod 2,L[3],L[7],L[1]]: map(Bit1, [61,56,36,
  40,7,14], K);
                                [164, 4, 60, 20, 186, 50]
> map(xbyte, [[1,1,0,1,0,1,0,1], [0,0,1,0,0,1,0,1]]);
                                ["X", "%"]
> B:=Bit("G", [k$k=1..8]); "G"=xbyte(Bit(G,[k$k=1..8])), Bit(xbyte(B), [k$k=1..8])=B;
                                B := [0, 1, 0, 0, 0, 1, 1, 1]
                                "G" = "G" [0, 1, 0, 0, 0, 1, 1, 1] = [0, 1, 0, 0, 0, 1, 1, 1]
> map(xbyte1, [101101, 1110101, 11101011, 11011, 1011001, 1001010, 10001001, 10000001, 10101]);
                                ["-", "u", "π", "\e", "Y", "J", "%o", "Í", "□"]
> map(xbyte1, [1010101,11010101,11100011,110110,10001001,10000001,1010101,11111001]);
                                ["U", "X", "r", "6", "%o", "Í", "U", "III"]
> map(xNB, [10011,11001101,1102011,10101,1001010011101,11100011,110110,10001001]);
                                [true, true, false, true, true, true, true, true]

```

boolop - basic functions/operators of Boolean algebra

Call format of the variables exported by the module:

boolop:- X(*args*) or **with(boolop):** X(*args*)

Formal arguments of the variables calls exported by the module:

X - name of a function/operator exported by the **boolop** module

args - formal arguments corresponding to the exported function/operator

Description of the module:

The module boolop exports a series of the useful functions/operators intended for providing of the basic operations of Boolean algebra, namely:

&andB, **&orB**, **&xorB**, **¬B**, **&impB**

where **&andB** - *n*-ary logical function/operator **and**, **&orB** - *n*-ary logical function/operator **or**, **&xorB** - *n*-ary logical function/operator **xor**, **¬B** - unary logical function/operator **not**, **&impB** - *n*-ary logical function/operator of implication (*implies*).

```

boolop := module ()
export   `&andB`,      # n-ary logical function/operator and
        `&orB`,       # n-ary logical function/operator or
        `&xorB`,      # n-ary logical function/operator xor
        `&notB`,      # unary logical function/operator not
        `&impB`,      # n-ary logical function/operator of implication (implies)
description "Basic logical operators/functions with Maple of releases 6, 7, 8, 9 and 10";
options   `CopyRight (c) RANS_IAN = Tallinn-Grodno-Vilnius; 14.06.2005`, package;
`&andB` := () -> `if` (nargs = 0, ERROR("arguments are missing"), `if` (belong({args}, 0 .. 1),
  `if` (^+(args) = nargs, 1, `if` (member(1, {args}), 1, 0)), ERROR("arguments %1 are invalid",
  {args} minus {0, 1}));
`&orB` := () -> `if` (nargs = 0, ERROR("arguments are missing"), `if` (belong({args}, 0 .. 1),

```

```

`if`(`+`(args) = nargs, 1, 0), ERROR("arguments %1 are invalid", {args} minus {0, 1}));
`&xorB` := () -> `if`(nargs = 0, ERROR("arguments are missing"), `if`(belong({args}, 0 .. 1),
`if`(parse(cat(seq(`if`(args[k] = 0, " true xor", " false xor"), k=1 .. nargs))[1 .. -5]) = `true`, 0, 1),
ERROR("arguments %1 are invalid", {args} minus {0, 1}));
`&notB` := (x::{0, 1}) -> x+1 mod 2;
`&impB` := (x::{0, 1}, y::{0, 1}) -> `if`([x, y] = [0, 1], 1, 0);
end module

```

Типичные примеры применения средств, экспортируемых модулем:

```

> with(boolop); ⇒ [&andB, &impB, &notB, &orB, &xorB]
> 0 &andB 0, 1 &andB 1, 0 &andB 1, 1 &andB 0, `&andB`(1, 0, 1, 0, 1, 0); ⇒ 0, 1, 1, 1, 1
> 0 &orB 0, 1 &orB 1, 0 &orB 1, 1 &orB 0, `&orB`(1, 0, 1, 0, 1, 0); ⇒ 0, 1, 0, 0, 0
> 0 &xorB 0, 1 &xorB 1, 0 &xorB 1, 1 &xorB 0, `&xorB`(1, 0, 1, 0, 1, 0); ⇒ 1, 1, 0, 0, 0
> [ `&notB`(0), `&notB`(1) ], [0 &impB 0, 1 &impB 0, 0 &impB 1, 1 &impB 1]; ⇒ [1, 0], [0, 0, 1, 0]

```

xpack - a packing/unpacking of the positive integers

Call format of the procedure:

xpack(N)

Formal arguments of the procedure:

N - a positive integer or string

Description of the procedure:

The *xpack(N)* procedure, working according to the principle of *switch*, provides a packing of positive integers (*with providing of the subsequent restoration*) by means of putting of two decimal numerals in one byte. On a positive integer as the actual N argument, the procedure returns the string equivalent of the number N, while on the equivalent the number represented by it is being returned. However, the given operation is not reversible, i.e. not each string is an equivalent of some positive integer, as that illustrates the last example of use of the *xpack* procedure of a fragment represented below. Namely, the following relation takes place: *xpack(xpack(N)) = N*, where N - a positive integer; this relation, in general, is incorrect for arbitrary string N, excluding a case when N is a result of the procedure call *xpack(M)*, where M - a positive integer. The *xpack(N)* procedure essentially uses the *Bit* and *xbyte* procedures considered above. The examples represented below illustrate use of this procedure for a packing/unpacking of information.

```

xpack := proc(N::{nonnegint, string})
local a, b, c, d, k, s, p, h, q, Tab_con;
option remember;
    Tab_con := table([0 = [0, 1, 1, 0], 1 = [0, 1, 1, 1], 2 = [1, 0, 0, 0], 3 = [1, 0, 0, 1], 4 = [1, 0, 1, 0],
    5 = [1, 0, 1, 1], 6 = [1, 1, 0, 0], 7 = [1, 1, 0, 1], 8 = [1, 1, 1, 0], 9 = [1, 1, 1, 1]]);
    assign(q = "", h = [p $ (p = 1 .. 8)]);
if type(N, integer) then assign67(a = cat("", `if`(type(length(N), even), N, cat(`0`, N))), s = "");
    for k by 2 to length(a) do assign67('b' = SN(a[k .. k + 1])), assign67('c' = trunc(1/10*b));
    assign67('b' = b - 10*c), assign67('d' = [op(Tab_con[c]), op(Tab_con[b])]);
    assign67('s' = cat(s, xbyte(d)))
    end do;
    RETURN(s)
else
    for k to length(N) do assign67('a' = Bit(N[k], h));

```

```

    assign('b' = RTab(Tab_con, a[1 .. 4]), 'c' = RTab(Tab_con, a[5 .. 8]));
    assign67('q' = cat(q, b, c))
  end do;
  RETURN(SN(q))
end if
end proc

```

Typical examples of the procedure use:

```

> assign('G' = map(xpack, [0,1942,2004,14062004])), assign('S' = map(xpack, G)), G, S;
    ["f", "□", "tj", "zltj"], [0, 1942, 2004, 14062004]
> xpack(424706762089961), xpack(xpack(424706762089961));
    "jЉЦИИняЗ", 424706762089961
> xpack(xpack("Grodno")), xpack(05240001400453510549041414), xpack(xpack("Tallinn"));
    "gvцno", "кЉfg |j№ kļzz", "linn"
> xpack(%[2]), xpack("RANS_IAN"), xpack(8938), xpack(1302198904081996),
    xpack("yh□пjn□ь");
    524000141004535105490414140, 8938, "пћ", "yh□пjn□ь", 1302198904081996

```

logbytes – basic logical operations with bytes

Call format of the procedure:

logbytes(O::symbol {, args})

Formal arguments of the procedure:

O – basic logical operation {*may be* {`'xor'`, `'and'`, `'not'`, `'or'`, `'implies'`}}

args – (*optional*) sequence of bytes of type {*string, symbol*}

Description of the procedure:

Procedure *logbytes* provides performance of basic logical operations {'xor', 'and', 'not', 'or', 'implies'} with bytes of type {*symbol, string*}. Logical operations are done with the corresponding bits of bytes defined by *args*-sequence of bytes. The result-byte is returned in *string*-format. Are supposed *n*-ary operations {'xor', 'and', 'or'}, binary operation 'implies' and unary operation 'not'. Procedure *logbytes* processes the basic especial and erroneous situations and uses two procedures *Bit* and *xbyte1*, and the module **boolop** which are considered above.

```
logbytes := proc(O::symbol)
```

```
local a, k, j;
```

```
  if nargs = 1 then error "number of arguments must be > 1"
```

```
  elif not member(O, {'xor', 'and', 'not', 'or', 'implies'})
```

```
  then error "1st argument must be {'xor', 'and', 'not', 'or', 'implies'} but has received <%1>", O
```

```
  else a := {};
```

```
    for k from 2 to nargs do
```

```
      if (proc(x) try type(x, {'symbol', 'string'}) and type(x, 'byte') catch : return false end try
```

```
      end proc)(args[k]) then next
```

```
      else a := {op(a), k}
```

```
      end if
```

```
    end do
```

```
  end if;
```

```
  if a <> {} then error "arguments with numbers %1 must be bytes of type {string, symbol}", a
```

```
  else a := map(Bit, [args[2 .. -1]], ['k' $ ('k' = 1 .. 8)])
```

```

end if;
if O = `and` then xbyte1([seq(boolop[`&andB`](seq(a[k][j], k = 1 .. nops(a))), j = 1 .. 8)])
elif O = `or` then xbyte1([seq(boolop[`&orB`](seq(a[k][j], k = 1 .. nops(a))), j = 1 .. 8)])
elif O = `xor` then xbyte1([seq(boolop[`&xorB`](seq(a[k][j], k = 1 .. nops(a))), j = 1 .. 8)])
elif O = `not` then xbyte1([seq(boolop[`&notB`](a[1][j]), j = 1 .. 8)])
elif O = `implies` and 2 < nargs then xbyte1([seq(boolop[`&impB`](seq(a[k][j], k=1..2)), j = 1..8)])
else 'procname(args)'
end if
end proc

```

Typical examples of the procedure use:

```
> logbytes(`non`, A, V, Z, A, G, N, V, S, V);
```

Error, (in *logbytes*) 1st argument must be {`and`, `or`, `xor`, `not`, `implies`} but has received <non>

```
> logbytes(`implies`, A); ⇒ logbytes(implies, A)
```

```
> map(logbytes, [ `and`, `or`, `xor`, `not`, `implies` ], W, V, R, Q, N, K, T, R, U, S, A);
```

```
["_", "@", "F", "", ""]
```

```
> logbytes(`implies`, [1, 0, 0, 0, 1, 0, 0, 1], v, z, A, g, n);
```

Error, (in *logbytes*) arguments with numbers {2} must be bytes of type {string, symbol}

Taking into account the important role of the *bit-by-bit* information processing in the *Maple* environment, it is supposed, the tools represented in the present section, should be essentially extended with increase of their reactivity and functionality.

It is necessary to be surprised only how in mathematically oriented package such means were found oneself outside of a field of vision of developers. Up to that, the reader for a programming of algorithms dealing with bit-by-bit information processing, can use means of the present chapter of the book or use them as a basis for creation of own similar means. Experience of use of the suggested means has shown their quite satisfactory efficiency at the decision of a lot of problems bit-by-bit information processing and in problems of mathematical logic.

The full set of software of the given orientation is represented in our books [239, 240] whereas the last release of library with these means is accessible to free-of-charge loading from website [259].

Part 2.

Application of *Maple* for solution of engineering-physical problems

Finite Element Method (FEM) was first developed in 1943 by R. Courant, who utilized the Ritz method of numerical analysis and minimization of variational calculus to obtain approximate solutions to vibration systems. Shortly thereafter, a paper published in 1956 by M. J. Turner, R. W. Clough, H. C. Martin, and L. J. Topp established a broader definition of numerical analysis. By the early 70's, FEM was limited to expensive mainframe computers generally owned by the aeronautics, automotive, defense, and nuclear industries. Since the rapid decline in the cost of computers and the phenomenal increase in computing power, FEM has been developed to an incredible precision. Present day supercomputers are now able to produce accurate results for all kinds of parameters.

FEM consists of a computer model of a material or design that is stressed and analyzed for specific results. It is used in new product design, and existing product refinement. A company is able to verify a proposed design will be able to perform to the client's specifications prior to manufacturing or construction. Modifying an existing product or structure is utilized to qualify the product or structure for a new service condition. In case of structural failure, FEM may be used to help determine the design modifications to meet the new condition.

FEM uses a complex system of points called nodes which make a grid called a mesh. This mesh is programmed to contain the material and structural properties which define how the structure will react to certain loading conditions. Nodes are assigned at a certain density throughout the material depending on the anticipated stress levels of a particular area. Regions which will receive large amounts of stress usually have a higher node density than those which experience little or no stress. Points of interest may consist of: fracture point of previously tested material, fillets, corners, complex detail, and high stress areas. The mesh acts like a spider web in that from each node, there extends a mesh element to each of the adjacent nodes. This web of vectors is what carries the material properties to the object, creating many elements. A wide range of FEM applications, we uses the method for solving of numerous problems in mechanics, hydromechanics,

The *second* part of the offered book demonstrates the use of mathematical package *Maple* for solution of a whole series of important engineering-physical problems in such fields as: thermal physics, heating engineering, elasticity theory, mechanics, hydrodynamics, hydromechanics, etc. The given problems were solved by applying the following numerical methods: FEM and method of the characteristics in combination with functional facilities of mathematical package *Maple*.

Chapter 13.

Applied problems of mechanics - 1

At the solution of dynamic problems of the mechanics of a solid body there is a necessity of definition of moments of inertia of masses of bodies of the composite geometrical forms, and also of systems of bodies relative to various axes of coordinates. As examples the problems over definition of moments of inertia of masses of a body with the composite geometrical form and of systems of such bodies are considered. The package *Maple* allows successfully to obtain analytical expressions of equations of motion and then successfully to solve them, using the numerical methods, and also to reproduce animation of a motion of mechanical systems. As an example the motion of a composite pendulum is considered. In mechanical engineering the mechanical transmissions with a composite motion are widely used, which include the toothed transmissions, clutches etc. As an example of the solution of such problems, the problem of a motion of a mechanical transmission is considered, in which the backlashes in elements of transmission are taken into account.

13.1. Calculation of moments of inertia of a solid body

13.1.1. Calculated expressions for definition of moments of inertia of a solid body

At designing of mechanical systems there are problems, for solution of which it is necessary to know allocation of mass in these mechanical systems. The modern mechanical systems include a great many of details, form of which, sometimes, happens very composite. Therefore, for calculation of mechanical systems it is necessary to have a program for calculation of moments of inertia of mass of a body of any composite form. The calculation of *moments of inertia of mass* and other parameters of a body is fulfilled by the numerical method. Most suitable for this problem is the FEM. The explored body is divided by a finite number of three-dimensional quadratic isoparametric elements (*fig. 13.1*). Use of such finite elements allows precisely to approximate the geometrical form of a body.

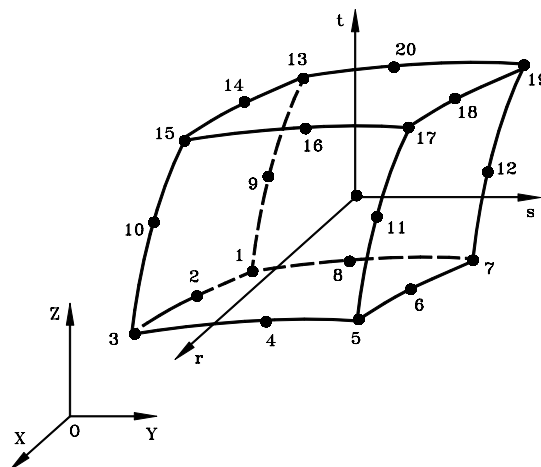


Fig. 13.1. A three-dimensional quadratic isoparametric finite element

The basic characteristics of a body are defined in a rectangular frame $X - Y - Z$ [106]. The static moments of a solid body are determined by the next relations:

$$S_x = \int_V \rho x \, dV, \quad S_y = \int_V \rho y \, dV, \quad S_z = \int_V \rho z \, dV, \quad \text{where } V - \text{volume of a body.} \quad (13.1)$$

The *moments of inertia* of a solid body relative to coordinate axes are defined as:

$$I_x = \int_V \rho (y^2 + z^2) \, dV; \quad I_y = \int_V \rho (x^2 + z^2) \, dV; \quad I_z = \int_V \rho (x^2 + y^2) \, dV \quad (13.2)$$

The *moments of inertia* of a solid body relative to coordinate planes are defined by the following relations:

$$I_{yoz} = \int_V \rho x^2 \, dV; \quad I_{zox} = \int_V \rho y^2 \, dV; \quad I_{xoy} = \int_V \rho z^2 \, dV \quad (13.3)$$

The *moment of inertia* of a solid body relative to beginning of coordinates is:

$$I_o = \int_V \rho (x^2 + y^2 + z^2) \, dV \quad (13.4)$$

While the *centrifugal moments of inertia* of a solid body are defined by the following relations:

$$I_{yz} = \int_V \rho yz \, dV; \quad I_{zx} = \int_V \rho zx \, dV; \quad I_{xy} = \int_V \rho xy \, dV \quad (13.5)$$

In a finite element *each coordinate* is approximated by the next relation:

$$x = \sum_{i=1}^p N_i(\mathbf{r}, \mathbf{s}, \mathbf{t}) x_i; \quad y = \sum_{i=1}^p N_i(\mathbf{r}, \mathbf{s}, \mathbf{t}) y_i; \quad z = \sum_{i=1}^p N_i(\mathbf{r}, \mathbf{s}, \mathbf{t}) z_i \quad (13.6)$$

where $N_i(\mathbf{r}, \mathbf{s}, \mathbf{t})$ – shape functions of a finite element; x_i, y_i, z_i – coordinates of nodes of a finite element. Three-dimensional integral over volume of a finite element is calculated according to the following formula:

$$I^{(e)} = \int_{V^{(e)}} f(x, y, z) \, dV = \int_{V^{(e)}} f(x, y, z) \det[\mathbf{J}] \, dr \, ds \, dt, \quad \text{where } f(x, y, z) - \text{intergrand;} \quad (13.7)$$

$$[\mathbf{J}] - \text{determinant of Jacobi matrix and } [\mathbf{J}] = \begin{bmatrix} \frac{\partial x}{\partial r} & \frac{\partial y}{\partial r} & \frac{\partial z}{\partial r} \\ \frac{\partial x}{\partial s} & \frac{\partial y}{\partial s} & \frac{\partial z}{\partial s} \\ \frac{\partial x}{\partial t} & \frac{\partial y}{\partial t} & \frac{\partial z}{\partial t} \end{bmatrix} \quad (13.8)$$

Three-dimensional integral over *volume* of a body is defined by the next relation:

$$I = \sum_{e=1}^{NE} I^{(e)} \quad (13.9)$$

The *coordinates of a centre of masses* of a body are defined as follows:

$$x_c = \frac{S_x}{V}; \quad y_c = \frac{S_y}{V}; \quad z_c = \frac{S_z}{V}, \text{ where } V - \text{volume of a body} \quad (13.10)$$

The *moments of inertia* of a solid body *relative to central axes X', Y', Z'*, which pass via a point C of a centre of masses and are parallel to coordinate axes X, Y, Z (fig. 13.2), are determined similarly over the above formulas on the basis of replacement of the coordinates, namely:

$$x' = x - x_c; \quad y' = y - y_c; \quad z' = z - z_c \quad (13.11)$$

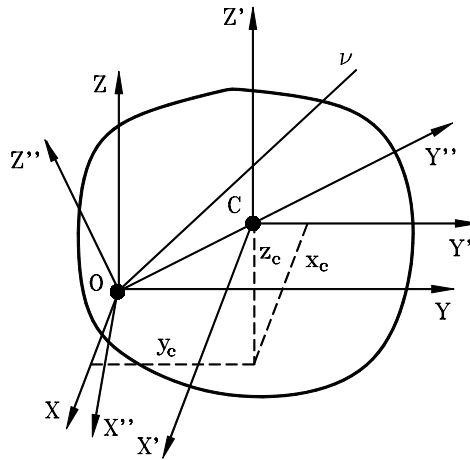


Fig. 13.2. The coordinates systems of a solid body

We draw via a point O in a frame X-Y-Z an axis ν , and turn a solid body around of this axis onto a corner α counter-clockwise. Then the frame X-Y-Z will occupy a new standing X''-Y''-Z''. The coordinates of points in the new frame X''-Y''-Z'' are defined by the following relations:

$$\{R''\} = [T(\alpha)]^T \{R\}, \quad (13.12)$$

where $\{R''\}^T = [x'' \ y'' \ z'']; \{R\}^T = [x \ y \ z]$ and $[T(\alpha)]$ – matrix of transformation of coordinates,

$$[T(\alpha)] = [E] + [R] \sin \alpha + 2[R]^2 \sin^2 \frac{\alpha}{2}; \quad (13.13)$$

$[E]$ – unit matrix, matrix $[R]$ of the next view: $[R] = \begin{bmatrix} 0 & -e_z & e_y \\ e_z & 0 & -e_x \\ -e_y & e_x & 0 \end{bmatrix}, \{e\}^T = [e_x \ e_y \ e_z]$ – unit

vector in the frame X-Y-Z defining standing of a rotation axis $O\nu$. The *moments of inertia* of a solid body in the frame X''-Y''-Z'' are calculated according to the above-mentioned formulas.

13.1.2. Input data for the solution of the problem

For solution of a problem concerning of definition of moments of inertia of a solid body the *Mass_inertia* program is used. All data necessary for operation of this program should be recorded into a file in advance; therefore, variable F, necessary qualifier of the file, is ascribed. The data in the

file are placed in the strict order, namely. In the *first* line the number of finite elements (*parameter nelem*), number of nodes (*npoin*) and number of groups of finite elements (*ngr*) are coded. Into the corresponding group of the finite elements are included those elements, which have *identical values* of density of a material.

In the *second* line, a print code of intermediate results (*kprint*), angle of rotation of a frame (*alfa*) and also elements of the **Avector(3)** array are coded. If *kprint=0*, then the intermediate results are not printed; otherwise, they are printed out. Into the array **Avector(3)** the elements of an unit vector **{e}** are written, which defines a standing of rotation axis **Ov** in the frame **X-Y-Z**.

In each subsequent *nelem* lines, the number of a finite element and the numbers of nodes of this element {array **Mtop(nelem, nnode)**, where *nnode* – number of nodes of the finite element, i.e. *nnode* =20} are coded, and also the number of group, to which belongs this finite element {array **Mgr(nelem)**}. Behind of arrays **Mtop** and **Mgr** into the data file the elements of **Coord(npoin, ndime)** array, where *ndime* – dimensionality of the problem (*ndime*=3), are recorded line by line. Into the *first* column of the **Coord** array the **x**-coordinates, while into the *second* and the *third* columns – the **y**- and **z**-coordinates of nodes of finite elements accordingly are recorded. Into each line of the file the number of a node, and also its (**x, y, z**)-coordinates are recorded.

Behind of the **Coord** array the elements of the **Param(ngr)** array are recorded line by line. Into each line of the datafile, a line number of **Param** array and a value of *density of a material* of a solid body, are recorded. The lines of **Akd** array should correspond to the lines of **Lkd** array. The line number of **Param** array should strictly correspond to the number of group of finite elements. In the datafile between the recorded arrays the lines of the comments with names of appropriate arrays are located. At reading of the information from the datafile these text lines are skipped.

Schematic structure of the datafile:

Text line *
nelem, npoin, kprint, alfa, Avector(3)
 Text line *
 Arrays *Mtop(nelem, nnode), Mgr(nelem)*
 Text line *
 Array *Coord(npoin, ndime)*
 Text line *
 Array *Param(ngr)*

13.1.3. Brief description of the Mass_inertia program solving the problem

The *Mass_inertia* program was programmed on the *Maple*-language; it consists of the basic program and 13 procedures. All procedures can be divided into two groups: procedures for data entry and for calculation. Memory size necessary for the solution of a concrete problem, and a time of its solution depend on the used number of finite elements, and number of nodes. The program calculates *coordinates of centre of masses* and *moments of inertia* of a solid body of the arbitrary form, which consists of the *various materials*. The calculation results are output on the monitor and are recorded into a file, for that to variable *file_rez1* of the program must be ascribed a qualifier of a target file. Into the file *file_rez1* values of coordinates of centre of masses and moments of inertia of a solid body are recorded.

13.1.4. An example of use of the Maple-program Mass_inertia

As an example of application of the *Mass_inertia* program, the definition of moments of inertia of a solid body represented on the *fig. 13.3*, is considered. The explored solid body consists of two

groups of finite elements: to the *first* group concern the *first* and the *second* finite element, while to the *second* group – the *third* finite element.

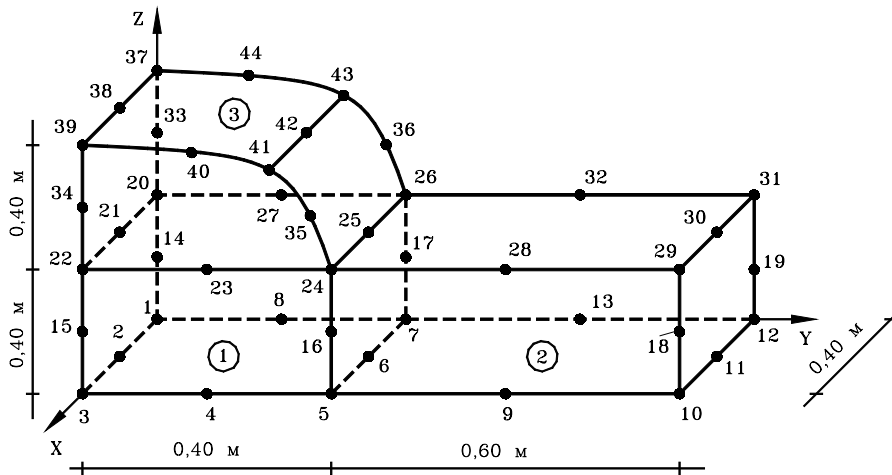


Fig. 13.3. The calculated scheme of the explored solid body

Input data for the test example: number of the finite elements $nelem=3$, number of nodes $npoin=44$, number of groups of finite elements $ngr=2$, number of boundary conditions $nbond=1$, $kprint=0$, $nq=4$, $nst=2$, $nforc=1$, $ksolve=0$, $niter=500$, $toler=10^{(-9)}$, $ngama=1$, $nkd=1$, $ntime=500$, $dtime=10^{(-4)}$ s, $neigen=5$, $niterv=100$, $nout=2$, $nstep=2$, $tolerv=10^{(-6)}$. Density of a material of the first group

$\rho_1 = 2700 \frac{\text{kg}}{\text{m}^3}$ and the second group $\rho_2 = 7850 \frac{\text{kg}}{\text{m}^3}$ accordingly; angle of rotation of the frame

$\alpha = 45^\circ$; unit vector $\{e\}^T = [0 \ 0,707 \ 0,707]$. The remaining data necessary for calculation, are represented on the fig. 13.3.

Results of calculation over the test example:

Moments of Inertia, Mass, Volume, Coordinates of Centre Mass:

- Inertia tensor $I_{xx}=3.142988e+02$
- Inertia tensor $I_{xy}=-5.648783e+01$
- Inertia tensor $I_{xz}=-6.198557e+01$
- Inertia tensor $I_{yy}=1.986480e+02$
- Inertia tensor $I_{yz}=-7.965422e+01$
- Inertia tensor $I_{zz}=2.036212e+02$
- Mass of body $mass=8.247217e+02$
- Volume of body $V=2.100282e-01$
- Coordinate of centre of mass $xc=2e-01$
- Coordinate of centre of mass $yc=3.424660e-01$
- Coordinate of centre of mass $zc=3.757968e-01$
- Inertia tensor $I_{xcxc}=1.011031e+02$
- Inertia tensor $I_{xcyc}=1.135817e-14$
- Inertia tensor $I_{xczc}=2.365034e-15$
- Inertia tensor $I_{ycyc}=4.918921e+01$
- Inertia tensor $I_{yczc}=2.648553e+01$
- Inertia tensor $I_{zczc}=7.390651e+01$
- Rotation about an Axes:

Inertia tensor $I_{xx_new}=2.936519e+02$
 Inertia tensor $I_{xy_new}=-2.880000e+01$
 Inertia tensor $I_{xz_new}=-5.249549e+01$
 Inertia tensor $I_{yy_new}=1.504609e+02$
 Inertia tensor $I_{yz_new}=-8.997157e+01$
 Inertia tensor $I_{zz_new}=2.724303e+02$
 Coordinate of centre of mass $x_{c_new}=1.581019e-01$
 Coordinate of centre of mass $y_{c_new}=4.473306e-01$
 Coordinate of centre of mass $z_{c_new}=2.709322e-01$
 Inertia tensor $I_{xcxc_new}=6.808895e+01$
 Inertia tensor $I_{xcyc_new}=2.952597e+01$
 Inertia tensor $I_{xczc_new}=-1.716951e+01$
 Inertia tensor $I_{ycyc_new}=6.931003e+01$
 Inertia tensor $I_{yczc_new}=9.979061e+00$
 Inertia tensor $I_{zczc_new}=8.678953e+01$

Initial standing of a solid body and its standing at turning around of the given axis are represented on the *fig. 13.4*.

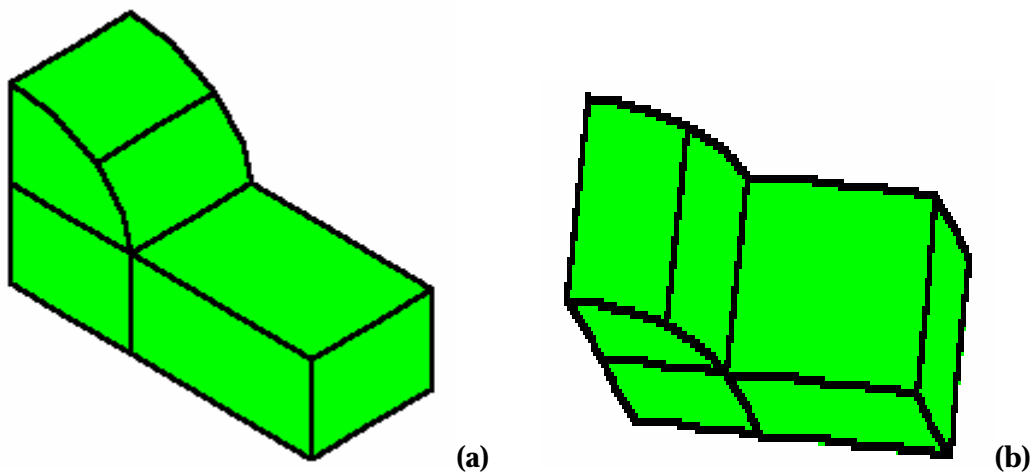


Fig. 13.4. Standings of the explored solid body: a – initial; b – at rotation around the given axis

The *Mass_inertia* program is intended for calculation of coordinates of centre of masses and moments of inertia of a solid body. The source module of the program in *Maple*-language, initial data for the test example, and also outcomes of its solution are represented in the PROBLEMS directory of archive attached to the present book in datafiles *Mb10_1_new.mws*, *Mb10_1.dat* and *Mb10_1_1.rez* accordingly.

13.2. Calculation of inertia moments of a system of solid bodies

The dynamic properties of mechanical systems depend on many parameters. The *moments of inertia* of solid bodies, and also *moments of inertia of all mechanical system* as a whole concern to these parameters. The composite mechanical systems consist of separate bodies of the manifold form and from various materials, densities of which are various. As examples of such mechanical systems can be: automobile and railway trains, robots, lifting-transport machines, building-road machines, metal-working machines etc. For the solution of dynamic problems of such mechanical systems with concentrated parameters it is necessary to determine moments of inertia of a system of solid bodies.

13.2.1. The calculated expressions for determination of inertia moments of a system of solid bodies

For determination of *inertia moments* of a system of solid bodies the common frame $\mathbf{X}-\mathbf{Y}-\mathbf{Z}$ is introduced. Each solid body has own frame $\mathbf{X}_i-\mathbf{Y}_i-\mathbf{Z}_i$, which is located at a centre of masses of this body (fig. 13.5).

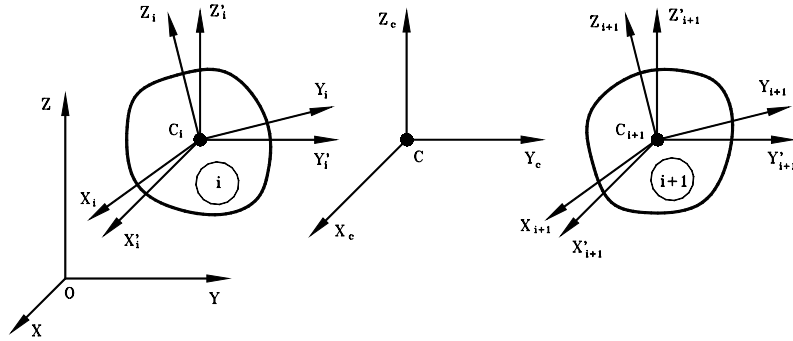


Fig. 13.5. Frames

The *orientation* of central coordinate axes \mathbf{X}_i , \mathbf{Y}_i and \mathbf{Z}_i of the solid bodies relative to common frame $\mathbf{X}-\mathbf{Y}-\mathbf{Z}$ is defined by the following dependence:

$$\begin{Bmatrix} \mathbf{x}_i \\ \mathbf{y}_i \\ \mathbf{z}_i \end{Bmatrix} = [\mathbf{T}] \begin{Bmatrix} \mathbf{x} \\ \mathbf{y} \\ \mathbf{z} \end{Bmatrix}, \text{ where } [\mathbf{T}] - \text{matrix of transformation of coordinates} \quad (13.14)$$

The *coordinates of centre of masses* of a system of solid bodies are defined by the following relations:

$$\mathbf{x}_c = \frac{\sum_{i=1}^n m_i \mathbf{x}_i}{\sum_{i=1}^n m_i}; \quad \mathbf{y}_c = \frac{\sum_{i=1}^n m_i \mathbf{y}_i}{\sum_{i=1}^n m_i}; \quad \mathbf{z}_c = \frac{\sum_{i=1}^n m_i \mathbf{z}_i}{\sum_{i=1}^n m_i} \quad (13.15)$$

Moments of inertia of a solid body relative to axes \mathbf{X}' , \mathbf{Y}' and \mathbf{Z}' , which pass via a centre of masses and are parallel to axes \mathbf{X} , \mathbf{Y} and \mathbf{Z} , are defined by the following relations:

$$\begin{aligned} I_{y'_i} &= (t_{11}^2 + t_{31}^2) I_{y_i o_i z_i} + (t_{12}^2 + t_{32}^2) I_{z_i o_i x_i} + (t_{13}^2 + t_{33}^2) I_{x_i o_i y_i} + \\ &+ 2(t_{11}t_{12} + t_{31}t_{32}) I_{x_i y_i} + 2(t_{11}t_{13} + t_{31}t_{33}) I_{z_i x_i} + 2(t_{12}t_{13} + t_{32}t_{33}) I_{y_i z_i}; \\ I_{x'_i} &= (t_{21}^2 + t_{31}^2) I_{y_i o_i z_i} + (t_{22}^2 + t_{32}^2) I_{x_i o_i z_i} + (t_{23}^2 + t_{33}^2) I_{x_i o_i y_i} + \\ &+ 2(t_{21}t_{22} + t_{31}t_{32}) I_{x_i y_i} + 2(t_{21}t_{23} + t_{31}t_{33}) I_{x_i z_i} + 2(t_{22}t_{23} + t_{32}t_{33}) I_{y_i z_i}; \\ I_{z'_i} &= (t_{11}^2 + t_{21}^2) I_{y_i o_i z_i} + (t_{12}^2 + t_{22}^2) I_{x_i o_i z_i} + (t_{13}^2 + t_{23}^2) I_{x_i o_i y_i} + \\ &+ 2(t_{11}t_{12} + t_{21}t_{22}) I_{x_i y_i} + 2(t_{11}t_{13} + t_{21}t_{23}) I_{x_i z_i} + 2(t_{12}t_{13} + t_{22}t_{23}) I_{y_i z_i}, \end{aligned} \quad (13.16)$$

where t_{ij} – elements of a matrix of transformation of coordinates. Centrifugal moments of inertia of a solid body relative to axes X' , Y' and Z' , which pass via a centre of masses and are parallel to axes X , Y and Z , are defined by the following relations:

$$\begin{aligned}
 I_{x'y'_i} &= t_{11}t_{21}(I_z - I_x) + t_{12}t_{22}(I_z - I_y) + (t_{11}t_{22} + t_{12}t_{21}) I_{xy} + \\
 &\quad + (t_{11}t_{23} + t_{13}t_{21}) I_{xz} + (t_{12}t_{23} + t_{13}t_{22}) I_{yz}; \\
 I_{z'x'_i} &= t_{11}t_{31}(I_z - I_x) + t_{12}t_{32}(I_z - I_y) + (t_{11}t_{32} + t_{12}t_{31}) I_{xy} + \\
 &\quad + (t_{11}t_{33} + t_{13}t_{31}) I_{xz} + (t_{12}t_{33} + t_{13}t_{32}) I_{yz}; \\
 I_{y'z'_i} &= t_{21}t_{31}(I_z - I_x) + t_{22}t_{32}(I_z - I_y) + (t_{21}t_{32} + t_{22}t_{31}) I_{xy} + \\
 &\quad + (t_{22}t_{33} + t_{23}t_{32}) I_{yz} + (t_{21}t_{33} + t_{33}t_{31}) I_{zx}.
 \end{aligned} \tag{13.17}$$

The *axial* and *centrifugal inertia moments* of a system of solid bodies relative to axes X_c , Y_c and Z_c , passing via centre of masses of the system, are determined as follows:

$$\begin{aligned}
 I_{x_c} &= \sum_{i=1}^n I_{x'_i} + (x_i - x_c)^2 m_i; \quad I_{y_c} = \sum_{i=1}^n I_{y'_i} + (y_i - y_c)^2 m_i; \\
 I_{z_c} &= \sum_{i=1}^n I_{z'_i} + (z_i - z_c)^2 m_i; \quad I_{x_c y_c} = \sum_{i=1}^n I_{x'_i y'_i} + (x_i - x_c)(y_i - y_c) m_i; \\
 I_{y_c z_c} &= \sum_{i=1}^n I_{y'_i z'_i} + (y_i - y_c)(z_i - z_c) m_i; \quad I_{z_c x_c} = \sum_{i=1}^n I_{z'_i x'_i} + (z_i - z_c)(x_i - x_c) m_i
 \end{aligned} \tag{13.18}$$

13.2.2. Input data for the solution of the problem

For solution of a problem concerning of definition of moments of inertia of a system of solid body the *Mass_system* program is used. All data necessary for operation of this program should be recorded into a file in advance; therefore, variable **F**, necessary qualifier of the file, is ascribed. The data in the file are placed in the strict order, namely.

In the *first* line the number of solid bodies (*parameter nelem*), number of lines which link some solid bodies (*nline*) and a print code of intermediate results (*kprint*) are coded. If *kprint*=0, then the intermediate results are not printed; otherwise, they are printed out.

In each subsequent *nelem* lines, the number of a solid body, its mass, moments of inertia relative to planes $I_{x_o y_i}$, $I_{y_o z_i}$, $I_{z_o x_i}$, and centrifugal moments of inertia $I_{x_i y_i}$, $I_{y_i z_i}$, $I_{z_i x_i}$ (*array Amase(nelem, 7)*). Behind of array *Amase(nelem, 7)* and into the data file the elements of *Coord(nelem, ndime)* array, where *ndime* – dimensionality of the problem (*ndime*=3), are recorded line by line. Into the *first* column of the *Coord* array the *x*-coordinates, while into the *second* – *y*-coordinates of nodes, and into the *third* – *z*-coordinates of centres of masses of solid bodies are recorded. Into each line of the file the number of a solid body, and also its (*x,y,z*)-coordinates of centre of masses are recorded.

Behind of the *Coord* array the elements of the *Tensor(3*nelem, nnode)* array are recorded line by line. Into the array *Tensor* the matrixes of transformation of coordinates $[T]$ of each solid body are

recorded one after another. Into each line of the datafile, the number of a solid body and three values of direction cosines of the matrix $[T]$ are recorded.

Behind of the **Tensor** array into the datafile the elements of the arrays **Mline**(*nline*, 2) and **Cline**(*nline*, 6) are recorded line by line. Into columns of the array **Mline** the numbers of two solid bodies, between which the straight line is traced, are recorded. Into columns of the array **Cline** the coordinates x_i , y_i and z_i of centre of masses of the *first* and the *second* solid bodies, between which the straight line is traced, are recorded. Into each line of the datafile are recorded the parameters such as: a line number, two numbers of solid bodies and the coordinates x_i , y_i , z_i of the *first* and the *second* solid bodies. In the datafile between the recorded arrays the lines of the comments with names of appropriate arrays are located. At reading of the information from the datafile these text lines are skipped.

Schematic structure of the datafile:

Text line *
nelem, nline, kprint
Text line *
Array *Amase(nelem, 7)*
Text line *
Array *Coord(nelem, ndime)*
Text line *
Array *Tensor(3*nelem, ndime)*
Text line *
Arrays *Mline(nline, 2), Cline(nline, 6)*

13.2.3. Brief description of the Mass_system program solving the problem

The *Mass_system* program was programmed on the *Maple*-language; it consists of the basic program and 10 procedures. All procedures can be divided into two groups: procedures for data entry and for calculation. Memory size necessary for the solution of a concrete problem, and a time of its solution depend on the used number of finite elements, and number of nodes. The given program calculates *coordinates of centre of masses* and *moments of inertia* of a system of solid bodies. The calculation results are output on the monitor and are recorded into a file, for that to variable *file_rez1* of the program must be ascribed a *qualifier* of a target file. Into the file *file_rez1* values of *coordinates of centre of masses* and *moments of inertia* of a system of solid bodies are recorded.

13.2.4. An example of use of the Maple-program Mass_system

As an example of application of the *Mass_system* program, the definition of inertia moments of the system of solid bodies, represented on the *fig. 13.6*, is considered.

Input data for the test example: number of the solid bodies *nelem*=10, the number of straight lines which link some solid bodies *nline*=8, the masses of solid bodies: $m_1 = m_3 = m_7 = m_9 = 20 \text{ kg}$; $m_2 = 100 \text{ kg}$; $m_4 = 200 \text{ kg}$; $m_5 = 50 \text{ kg}$; $m_6 = 500 \text{ kg}$; $m_8 = 150 \text{ kg}$; $m_{10} = 70 \text{ kg}$. The moments of inertia relative to centre of masses of a solid body: $I_1 = I_3 = I_7 = I_9 = 0,60 \text{ kg} \cdot \text{m}^2$; $I_2 = 15,0 \text{ kg} \cdot \text{m}^2$; $I_4 = 30,0 \text{ kg} \cdot \text{m}^2$; $I_5 = 3,0 \text{ kg} \cdot \text{m}^2$; $I_6 = 125,0 \text{ kg} \cdot \text{m}^2$; $I_8 = 20,0 \text{ kg} \cdot \text{m}^2$; $I_{10} = 5,0 \text{ kg} \cdot \text{m}^2$. The remaining data necessary for calculation, are represented on the *fig. 13.6*.

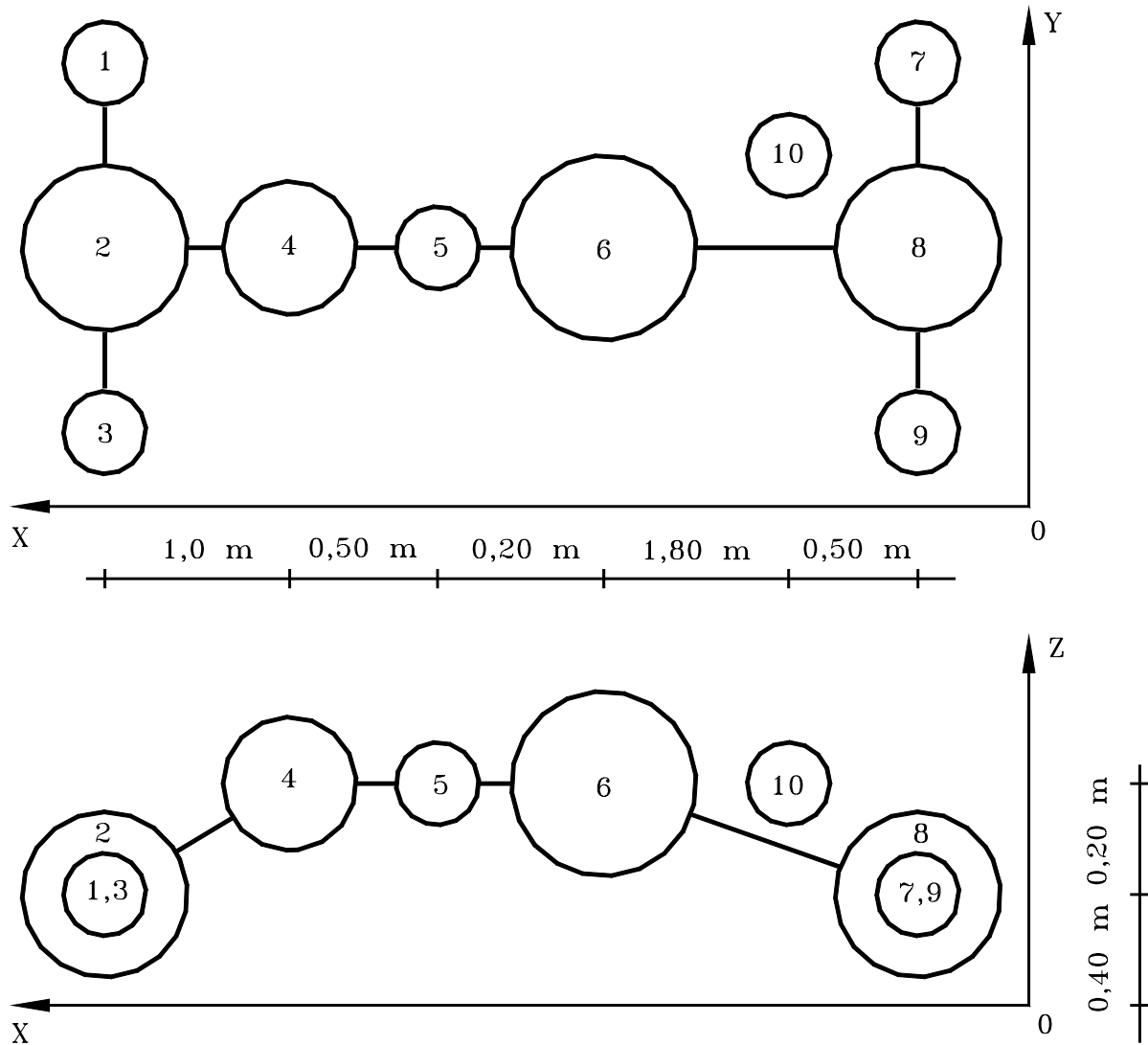


Fig. 13.6. System of solid bodies (form of each solid body is a sphere)

Results of calculation over the test example:

Mass, Coordinates of Center Mass and Inertia Moments:

Mass of system of rigid bodies $mass=1.15e+03$
 Coordinate of center of mass $x_c=2.147826e+00$
 Coordinate of center of mass $y_c=1.030434e+00$
 Coordinate of center of mass $z_c=5.426086e-01$
 Inertia moments about yoz plane $I_{yoz}=1.910269e+03$
 Inertia moments about zox plane $I_{zox}=2.968347e+02$
 Inertia moments about xoy plane $I_{xoy}=2.098121e+02$
 Inertia moments $I_{xy}=-5.767391e+01$
 Inertia moments $I_{yz}=2.008695e+00$
 Inertia moments $I_{zx}=2.975652e+01$

The scheme of the system of solid bodies is represented on the following fig. 13.7.

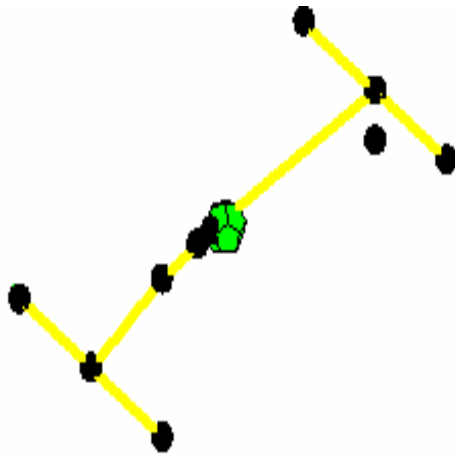


Fig. 13.7. The scheme of the system of solid bodies

The *Mass_system* program is intended for calculation of coordinates of centre of masses and moments of inertia of a system of solid bodies. The source module of the program in *Maple*-language, initial data for the test example, and also outcomes of its solution are represented in the PROBLEMS directory of archive attached to the book in files *Mb10_2_new.mws*, *Mb10_2.dat* and *Mb10_2_1.rez* accordingly.

13.3. Nonlinear oscillations of mechanical systems

In some mechanical systems the connection between separate solid bodies can be nonlinear. At investigation of oscillations of such mechanical systems there are difficulties at definition of amplitude-frequency characteristics. Therefore, such problems have practical significance at investigation of nonlinear oscillations of mechanical systems.

13.3.1. The calculated expressions for description of nonlinear oscillations of mechanical systems

A *nonlinear mechanical system* consisting of solid bodies, which are connected by *nonlinear elastic and dissipative elements*, is considered (fig. 13.8).

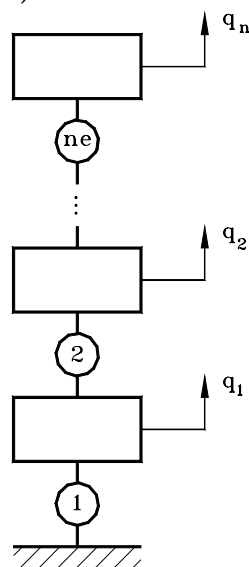


Fig. 13.8. The scheme of the mechanical system (by circle are marked the elements, consisting of a spring and a dampfer)

The *equations of motion* of a mechanical system are defined on the basis of the Lagrange equation of the second type, namely:

$$\frac{d}{dt} \left(\frac{\partial T}{\partial \dot{q}_i} \right) - \frac{\partial T}{\partial q_i} + \frac{\partial \Phi}{\partial \dot{q}_i} + \frac{\partial \Pi}{\partial q_i} = Q_i \quad (13.19)$$

where: T – kinetic energy of a mechanical system: $T = \sum_{i=1}^n \frac{1}{2} m_i \dot{q}_i^2$ (13.20)

$$\Pi$$
 – potential energy of a mechanical system: $\Pi = \sum_{i=1}^{ne} \frac{1}{2} k_i (\Delta_i)^2$ (13.21)

$$\Delta_i = q_i - q_{i-1} \quad (13.22)$$

$$\Phi$$
 – dissipation function: $\Phi = \sum_{i=1}^{ne} \frac{1}{2} h_i (\dot{\Delta}_i)^2$ (13.23)

$$\dot{\Delta}_i = \dot{q}_i - \dot{q}_{i-1} \quad (13.24)$$

q_i, \dot{q}_i – the generalized coordinates and velocity; Q_i – the generalized force; t – time; m_i – mass of a body; k_i, h_i – coefficients of stiffness and damping; n – the number of bodies; ne – the number of elements. The generalized force is represented as follows:

$$Q_i = A_{i,0} + \sum_{k=1}^{nf} A_{i,k} \cos kvt + B_{i,k} \sin kvt \quad (13.25)$$

where $A_{i,k}, B_{i,k}$ – known coefficients; v – excitation frequency; nf – harmonics of excitation. The coefficients of *stiffness* and *damping* are functions of the generalized coordinates, namely:

$$k_i = \sum_{j=0}^{nk} k_{ij} \Delta^j; \quad h_i = \sum_{j=0}^{nh} h_{ij} \Delta^j; \quad (i=1 \dots ne). \quad (13.26)$$

The equations system of motion of a mechanical system have the next form:

$$[M]\{\ddot{q}\} + \{F(q, \dot{q})\} = \{Q\}. \quad (13.27)$$

The generalized coordinate q_i with a period T is expanded into Fourier series:

$$q_i(t) = \frac{1}{2} a_{i,0} + \sum_{k=1}^{nq} a_{i,k} \cos k\omega t + b_{i,k} \sin k\omega t; \quad \omega = \frac{2\pi}{T}; \quad (i=1..n) \quad (13.28)$$

By substituting expressions (13.28) of the generalized coordinates into the equations system of a motion (13.27), we shall receive a vector of the equations with unknown coefficients $a_{i,k}$ and $b_{i,k}$, namely:

$$\{\Phi(\mathbf{a}, \mathbf{b})\} = \mathbf{0} \quad (13.29)$$

Each i -th equation, as a real periodic function with a period T , is expanded into Fourier series of the following form:

$$\Phi_i(t) = \frac{1}{2} z_{i,0} + \sum_{k=1}^{nq} (z_{c_{i,k}} \cos k\omega t + z_{s_{i,k}} \sin k\omega t), \text{ where } z_{i,0} = \frac{2}{T} \int_{-T/2}^{T/2} \Phi_i dt, \quad (13.30)$$

$$z_{c_{i,k}} = \frac{2}{T} \int_{-T/2}^{T/2} \Phi_i \cos k\omega t dt, \quad z_{s_{i,k}} = \frac{2}{T} \int_{-T/2}^{T/2} \Phi_i \sin k\omega t dt \quad (13.31-13.33)$$

Then, the equations system of a motion can be presented as follows:

$$\{\mathbf{z}(\mathbf{a}, \mathbf{b})\} = [z_1, \dots, z_n]^T = \mathbf{0}, \quad \{z_i\}^T = [z_{i,0} \ z_{c_{i,1}} \ z_{s_{i,1}} \ \dots \ z_{c_{i,nq}} \ z_{s_{i,nq}}] \quad (13.34)$$

The obtained equations system (13.34) is a system of nonlinear algebraic equations relative to coefficients $a_{i,k}$ and $b_{i,k}$. The given equations system is solved by the Newton method, namely:

$$[J]_i \{\Delta \mathbf{X}\}_i = -\{z\}_i, \text{ where: } [J]_i - \text{Jacobi matrix of the form } [J]_i = \frac{\partial \{z\}_i}{\partial \{X\}}; \quad (13.35)$$

$$\{X\} - \text{vector of unknowns, } \{X\}^T = \{a_{1,0} \ a_{1,1} \ b_{1,1} \ a_{1,2} \ b_{1,2} \ \dots \ a_{n,nq} \ b_{n,nq}\}. \quad (13.36)$$

13.3.2. Input data for the solution of the problem

For solution of a problem concerning of definition of coefficients of the corresponding harmonics of variables of a nonlinear mechanical system the *Amplitude* program is used. All data necessary for operation of this program should be recorded into a file in advance; therefore, variable F , necessary qualifier of the file, is ascribed. The data in the file are placed in the strict order, namely. In the *first* line the next parameters are recorded: number of bodies (*parameter nbody*); number of elastic and dissipative elements (*nelem*); number of harmonics (*nq*); number of addends defining coefficients of a stiffness (*nk*); number of addends defining coefficients of a damping (*nh*); number of working external forces (*nforce*) and the number of harmonics of external forces (*nf*).

In the *second* line the next parameters are recorded: a print code of intermediate results (*kprint*); number of steps on frequency (*nw*); a minimum frequency (*wmin*); a step of frequency (*dw*); number of iterations (*niter*) and precision of the solution (*toler*) at the solution of a system of the nonlinear equations by the Newton method. If *kprint*=0, then the intermediate results are not printed; otherwise, they are printed out.

In each subsequent *nelem* lines, the elements of the array $\mathbf{Amase}(nbody)$ are coded. Into each line of the datafile the number of a solid body and its mass are recorded. Behind of the array $\mathbf{Amase}(nbody)$ into the datafile the elements of the $\mathbf{Mtop}(nelem, 2)$ array are recorded line by line. Into this array are recorded the numbers of bodies, between which elements of elasticity and

damping are located. Into each line of the datafile the number of an element and also numbers of bodies are recorded. Behind of the array **Mtop** into the datafile the elements of the **Ak(nelem, nk)** and **Ah(nelem, nh)** arrays are recorded line by line. Into arrays **Ak** and **Ah** the addends for determination of coefficients of stiffness and dampings are recorded. Into each line of the datafile the number of an element and also **nk** coefficients of stiffness and **nh** coefficients of dampings are recorded.

Behind of the arrays **Ak** and **Ah** into the datafile the elements of the arrays **Mjega(nforce)**, **Pw(nforce)** and **Pjega(nforce)** are recorded line by line. Into each line of the array **Mjega** the number of a body onto which the external force affects is recorded; into the array **Pw** the frequency of excitations is recorded, at last into the array **Pjega** the components of external force are recorded. Into each line of the datafile the number of a body and **nf** of components of external forces are recorded. In the datafile between the recorded arrays the lines of the comments with names of appropriate arrays are located. At reading of the information from the datafile these text lines are skipped.

Schematic structure of the datafile:

Text line *
nbody, nelem, nq, nk, nh, nforce, nf
Text line *
kprint, nw, wmin, dw, niter, toler
Text line *
Array *Amase(nbody)*
Text line *
Array *Mtop(nelem, 2)*
Text line *
Arrays *Ak(nelem, nk), Ah(nelem, nh)*
Text line *
Arrays *Mjega(nforce), Pw(nforce), Pjega(nforce)*

13.3.3. Brief description of the Amplitude program solving the problem

The *Amplitude* program was programmed on the *Maple*-language; it consists of the basic program and 17 procedures. All procedures can be divided into three groups: procedures for data entry, for calculations and for output of results. Memory size necessary for the solution of a concrete problem, and a time of its solution depend on the used number of bodies, the number of steps on frequency and precision of solution. The program calculates coefficients of the corresponding harmonics of the variables of a nonlinear mechanical system, and also eigenvalues and own frequencies of a linear mechanical system. The calculation results are output on the monitor and are recorded into files, for that to variables *file_rez1* and *file_rez2* of the program must be ascribed the qualifiers of target files. Into the file *file_rez1* values of coefficients of the corresponding harmonics of the variables of a nonlinear mechanical system, while into the file *file_rez2* eigenvalues and own frequencies of a linear mechanical system are recorded accordingly.

13.3.4. An example of use of the Maple-program Amplitude

As an example of application of the *Amplitude* program, the oscillations of the mechanical system, represented on the *fig. 13.9*, for definition of coefficients of Fourier series of variables of the given system are considered.

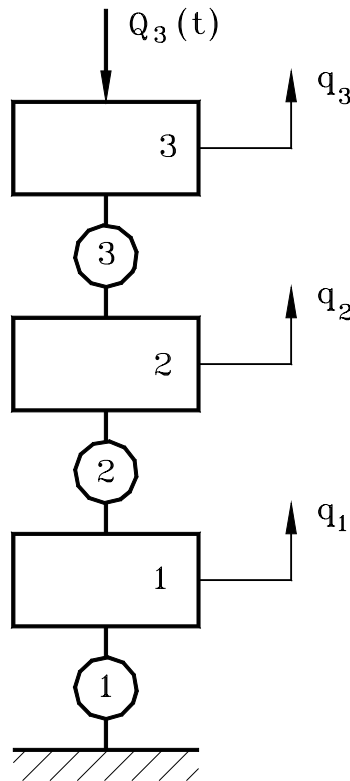


Fig. 13.9. The scheme of explored system

Input data for the test example: number of the solid bodies $nbody=3$; the number of elastic bodies $nelem=3$; number of harmonics $nq=3$; number of addends of coefficients of stiffness $nk=2$ and dampings $nh=2$; number of active external forces $nforce=1$; number of components of external force $nf=5$; $kprint=0$, $nw=21$, $wmin=5$ Hz, $dw=1$ Hz, $niter=5$, $toler=10^{(-6)}$. The masses of bodies: $m_1 = 10$ kg ; $m_2 = 5$ kg ; $m_3 = 2$ kg . Coefficients of stiffness of elements: $k_{1,0} = 5 \cdot 10^5$ N/m ; $k_{1,1} = 0$; $k_{2,0} = 2 \cdot 10^5$ N/m ; $k_{2,1} = 0$; $k_{3,0} = 1 \cdot 10^5$ N/m ; $k_{3,1} = 10^7$ N/m. Coefficients of dampings of elements: $h_{1,0} = 5 \cdot 10^{-2}$ Ns/m ; $h_{3,0} = 10^{-2}$ Ns/m ; $h_{2,0} = 3 \cdot 10^{-2}$ Ns/m ; $h_{1,1} = h_{2,1} = h_{3,1} = 0$. Excitation frequency $\nu = 15$ Hz and external force $Q_{31} = 5$ N ; $Q_{32} = 2$ N ; $Q_{33} = 6$ N ; $Q_{34} = 10$ N .

Results of calculation over the test example:

Eigenvalues (real and imaginary part) and own frequencies:

| | | |
|----------------------|----------------------|-------------------------------|
| Re[1]= -6.386769e-03 | Im[1]= 3.210184e+02 | frequency[1]= 1.791698e+01 Hz |
| Re[2]= -6.386769e-03 | Im[2]= -3.210184e+02 | frequency[2]= 1.791698e+01 Hz |
| Re[3]= -3.132243e-03 | Im[3]= 2.470895e+02 | frequency[3]= 1.571908e+01 Hz |
| Re[4]= -3.132243e-03 | Im[4]= -2.470895e+02 | frequency[4]= 1.571908e+01 Hz |
| Re[5]= -9.809867e-04 | Im[5]= 1.260711e+02 | frequency[5]= 1.122814e+01 Hz |
| Re[6]= -9.809867e-04 | Im[6]= -1.260711e+02 | frequency[6]= 1.122814e+01 Hz |

On the *fig. 13.10* the *dependences of coefficients* of the corresponding harmonics of the third variable of the nonlinear mechanical system from *frequency* are represented.

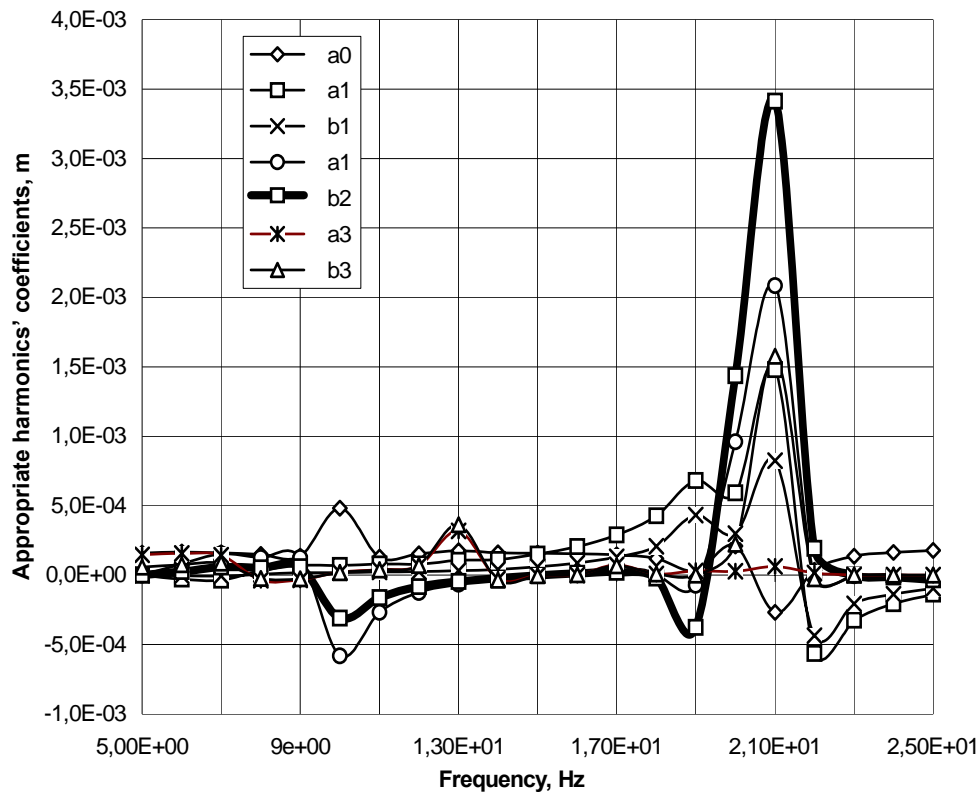


Fig. 13.10. Dependences of coefficients of the corresponding harmonics of the third variable of the nonlinear mechanical system from frequency

The *Amplitude* program is intended for the solution of problem on determination of *coefficients of the corresponding harmonics of variables of a nonlinear mechanical system*. The source module of the program in *Maple*-language, initial data for the test example, and also outcomes of its solution are represented in the PROBLEMS directory of archive attached to the book in files *Mb10_3_new.mws*, *Mb10_3.dat* and *Mb10_3_1.rez* accordingly.

13.4. Derivation and solution of equations of motion of a mechanical system with the concentrated parameters

The *Maple* package gives an opportunity to output and to solve equations of motion of mechanical systems with the concentrated parameters. At a composite motion of a mechanical system the equations of motion of this system have an unwieldy view and at their deduction the errors of a various kind are possible. The equations of motion can successfully be deduced, using the Lagrange equation of the *second* type. As an example of a mechanical system with the concentrated parameters, the system of an interesting construction consisting of a central pendulum and four satellites is explored. For providing of an unstable motion of such mechanical system the stationary magnets are used.

13.4.1. The calculated expressions for deduction of equations of motion of a mechanical system

The motion of a mechanical system in a vertical plane is considered. A mechanical system consists of six masses \mathbf{m}_i ($i=1 \dots 6$). The centre of masses of each body on the *fig. 13.11* is indicated by points \mathbf{c}_i ($i=1 \dots 6$). Thickness of a constant magnet is designated Δ_i ($i=1 \dots 3$), and the distances up to

magnets from points **O**, **O1** and **O2** are designated as **h1**, **h2** and **h3** accordingly. In each body, excepting the second, the stationary magnets are located. A poles of stationary magnets in points **O4**, **O5** and **O6** coincide with poles of stationary magnets located in the rotation bodies. Thus, onto the body which is located near to a constant magnet, a magnetic force affects. The equations of motion of a mechanical system are deduced, using the Lagrange equation of the second type, namely:

$$\frac{d}{dt} \left(\frac{\partial T}{\partial \dot{\mathbf{q}}} \right) - \frac{\partial T}{\partial \mathbf{q}} + \frac{\partial \Phi}{\partial \dot{\mathbf{q}}} + \frac{\partial \Pi}{\partial \mathbf{q}} = \{\mathbf{Q}\} \quad (13.37)$$

where **T**, **Π** – kinetic and potential energy of a mechanical system accordingly; **Φ** – dissipation function; **{q}**, **{q̇}** – vectors of generalized coordinates and velocities accordingly; **{Q}** – vector of generalized forces; **t** – time. The angles **φ1**, **φ2** and **φ3** are accepted as the generalized coordinates.

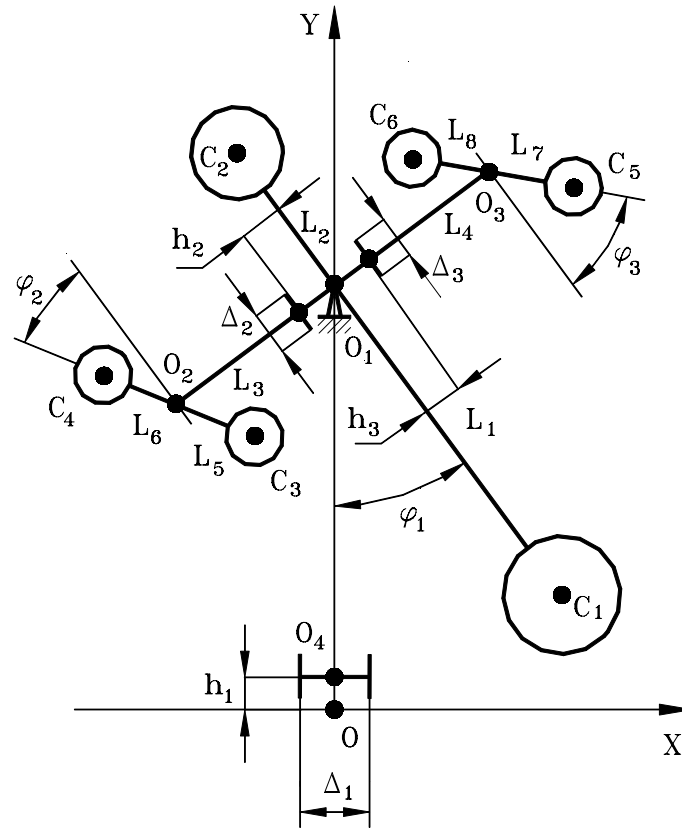


Fig. 13.11. The calculated scheme of explored mechanical system

The kinetic energy of a mechanical system is determined as follows:

$$T = \frac{1}{2} \sum_{i=1}^6 m_i v_{c_i}^2, \quad (13.38)$$

where **m_i** – masses of bodies and **v_{c₁}** = **L₁φ̇₁(cos φ₁ + sin φ₁)**; **v_{c₂}** = **-L₂φ̇₁(cos φ₁ + sin φ₁)**;

$$\begin{aligned} \mathbf{v}_{c_3} = & \dot{\phi}_1 [L_3 \sin \phi_1 + L_5 \cos(\phi_1 + \phi_2)] + \dot{\phi}_2 L_5 \cos(\phi_1 + \phi_2) + \\ & + \dot{\phi}_1 [-L_3 \cos \phi_1 + L_5 \sin(\phi_1 + \phi_2)] + \dot{\phi}_2 L_5 \sin(\phi_1 + \phi_2); \\ \mathbf{v}_{c_4} = & \dot{\phi}_1 [L_3 \sin \phi_1 - L_6 \cos(\phi_1 + \phi_2)] - \dot{\phi}_2 L_6 \cos(\phi_1 + \phi_2) - \\ & - \dot{\phi}_1 [L_3 \cos \phi_1 + L_6 \sin(\phi_1 + \phi_2)] - \dot{\phi}_2 L_6 \sin(\phi_1 + \phi_2); \end{aligned} \quad (13.39)$$

$$\begin{aligned} \mathbf{v}_{c_5} = & \dot{\varphi}_1[-L_4 \sin \varphi_1 + L_7 \cos(\varphi_1 + \varphi_3)] + \dot{\varphi}_3 L_7 \cos(\varphi_1 + \varphi_3) + \\ & + \dot{\varphi}_1[L_4 \cos \varphi_1 + L_7 \sin(\varphi_1 + \varphi_3)] + \dot{\varphi}_3 L_7 \sin(\varphi_1 + \varphi_3) ; \end{aligned}$$

$$\begin{aligned} \mathbf{v}_{c_6} = & \dot{\varphi}_1[-L_3 \sin \varphi_1 - L_8 \cos(\varphi_1 + \varphi_3)] - \dot{\varphi}_3 L_8 \cos(\varphi_1 + \varphi_3) + \\ & + \dot{\varphi}_1[L_4 \cos \varphi_1 - L_8 \sin(\varphi_1 + \varphi_3)] - \dot{\varphi}_3 L_8 \sin(\varphi_1 + \varphi_3) . \end{aligned}$$

The *potential energy* of a mechanical system is defined by the next relation:

$$\Pi = 0 \quad (13.40)$$

The *dissipation function* of a mechanical system is defined as follows:

$$\Phi = \frac{1}{2} \sum_{i=1}^3 H_i \dot{\varphi}_i^2, \text{ where } H_i - \text{a damping coefficient} \quad (13.41)$$

The *work of force of own weight* is determined as follows:

$$W = \sum_{i=1}^6 \{\mathbf{F}_{c_i}\}^T \delta \{\mathbf{r}_{c_i}\} = \sum_{i=1}^6 \{\mathbf{F}_{c_i}\}^T \left[\frac{\partial \mathbf{r}_{c_i}}{\partial \varphi_j} \right] \{\delta \varphi_j\}, \quad j = 1, 2, 3 \quad (13.42)$$

where $\{\mathbf{F}_{c_i}\}$ – vector of own weight of i -th mass, $\{\mathbf{F}_{c_i}\}^T = [0 \quad -m_i g \quad 0]$; $\{\mathbf{r}_{c_i}\}$ – vector of centre of masses

$$\text{of } i\text{-th body and } \{\mathbf{r}_{c_1}\} = \begin{Bmatrix} L_1 \sin \varphi_1 \\ L_9 - L_1 \cos \varphi_1 \\ 0 \end{Bmatrix}; \quad \{\mathbf{r}_{c_2}\} = \begin{Bmatrix} -L_2 \sin \varphi_1 \\ L_9 + L_2 \cos \varphi_1 \\ 0 \end{Bmatrix};$$

$$\{\mathbf{r}_{c_3}\} = \begin{Bmatrix} -L_3 \cos \varphi_1 + L_5 \sin(\varphi_1 + \varphi_2) \\ L_9 - L_3 \sin \varphi_1 - L_5 \cos(\varphi_1 + \varphi_2) \\ 0 \end{Bmatrix}; \quad \{\mathbf{r}_{c_4}\} = \begin{Bmatrix} -L_3 \cos \varphi_1 - L_6 \sin(\varphi_1 + \varphi_2) \\ L_9 - L_3 \sin \varphi_1 + L_6 \cos(\varphi_1 + \varphi_2) \\ 0 \end{Bmatrix}; \quad (13.43)$$

$$\{\mathbf{r}_{c_5}\} = \begin{Bmatrix} L_4 \cos \varphi_1 + L_7 \sin(\varphi_1 + \varphi_3) \\ L_9 + L_4 \sin \varphi_1 - L_7 \cos(\varphi_1 + \varphi_3) \\ 0 \end{Bmatrix}; \quad \{\mathbf{r}_{c_6}\} = \begin{Bmatrix} L_4 \cos \varphi_1 + L_8 \sin(\varphi_1 + \varphi_3) \\ L_9 + L_4 \sin \varphi_1 - L_8 \cos(\varphi_1 + \varphi_3) \\ 0 \end{Bmatrix}$$

The work of magnetic forces acting onto the first body, is defined as follows:

$$W_{M, c_1} = F_{M, c_1} \delta l_{c_1, o_4} \quad (13.44)$$

where: F_{M, c_1} – force of a constant magnet acting onto the first body;

$$F_{M, c_1} = c e^{-a l_{c_1, o_4}}; \quad (13.45)$$

c, a – constants; l_{c_1, o_4} – distance from a point C_1 up to a point O_4 (see fig. 13.11),

$$l_{c_1, o_4} = \left(\{\mathbf{R}_{c_1, o_4}\}^T \{\mathbf{R}_{c_1, o_4}\} \right)^{\frac{1}{2}}; \quad \{\mathbf{R}_{c_1, o_4}\} = \{\mathbf{r}_{c_1}\} - \{\mathbf{r}_{o_4}\}; \quad \{\mathbf{r}_{o_4}\}^T = [0 \quad h_1 \quad 0]. \quad (13.46)$$

The variation of distance l_{o_4, c_1} is defined by the following relation:

$$\delta I_{c_1, o_4} = \frac{\partial I_{c_1, o_4}}{\partial \{R_{c_1, o_4}\}} \delta \{R_{c_1, o_4}\} = \frac{1}{I_{c_1, o_4}} \{R_{c_1, o_4}\}^T \delta \{R_{c_1, o_4}\} = \frac{1}{I_{c_1, o_4}} \{R_{c_1, o_4}\}^T \left\{ \frac{\partial R_{c_1, o_4}}{\partial \varphi_1} \right\} \delta \varphi_1 \quad (13.47)$$

Then the generalized magnetic force acting onto the first body, is determined as follows:

$$W_{M, c_1} = F_{M, c_1} \frac{1}{I_{c_1, o_4}} \{R_{c_1, o_4}\}^T \left\{ \frac{\partial R_{c_1, o_4}}{\partial \varphi_1} \right\} \delta \varphi_1 = Q_{M, c_1} \delta \varphi_1, \quad (13.48)$$

$$\text{where } Q_{M, c_1} = \frac{F_{M, c_1}}{I_{c_1, o_4}} \{R_{c_1, o_4}\}^T \left\{ \frac{\partial R_{c_1, o_4}}{\partial \varphi_1} \right\} \quad (13.49)$$

The *generalized magnetic forces* acting onto remaining bodies, are similarly determined, namely:

$$Q_{M, c_3} = \frac{F_{M, c_3}}{I_{c_3, o_5}} \{R_{c_3, o_5}\}^T \left\{ \frac{\partial R_{c_3, o_5}}{\partial \varphi_2} \right\}; \quad Q_{M, c_4} = \frac{F_{M, c_4}}{I_{c_4, o_5}} \{R_{c_4, o_5}\}^T \left\{ \frac{\partial R_{c_4, o_5}}{\partial \varphi_2} \right\}; \quad (13.50)$$

$$Q_{M, c_5} = \frac{F_{M, c_5}}{I_{c_5, o_6}} \{R_{c_5, o_6}\}^T \left\{ \frac{\partial R_{c_5, o_6}}{\partial \varphi_3} \right\}; \quad Q_{M, c_6} = \frac{F_{M, c_6}}{I_{c_6, o_6}} \{R_{c_6, o_6}\}^T \left\{ \frac{\partial R_{c_6, o_6}}{\partial \varphi_3} \right\}$$

The *generalized magnetic forces* act onto bodies under next *conditions*:

$$-\frac{\Delta_1}{2} \leq L_1 \sin \varphi_1 \leq \frac{\Delta_1}{2}; \quad -\frac{\Delta_2}{2} \leq -L_5 \cos \varphi_2 \leq \frac{\Delta_2}{2}; \quad -\frac{\Delta_2}{2} \leq L_6 \cos \varphi_2 \leq \frac{\Delta_2}{2}; \\ -\frac{\Delta_3}{2} \leq -L_7 \cos \varphi_3 \leq \frac{\Delta_3}{2}; \quad -\frac{\Delta_3}{2} \leq -L_8 \cos \varphi_3 \leq \frac{\Delta_3}{2} \quad (13.51)$$

The *force of resistance* to moving of *i*-th body in air is determined as follows:

$$F_{\text{resist}, i} = \frac{1}{2} c \rho S v^2 \quad (13.52)$$

where *c* – form factor of a body (for sphere $c \approx 0.5$); ρ – density of air; *S* – frontal area of a body; *v* – absolute velocity of motion of a body. Then, the moment of resistance to moving of *i*-th body is determined by the next relation:

$$M_{\text{resist}, i} = [R_i] \{F_{\text{resist}, i}\} = \begin{bmatrix} -R_{y, i} & R_{x, i} \end{bmatrix} \begin{Bmatrix} F_{\text{resist}, x, i} \\ F_{\text{resist}, y, i} \end{Bmatrix}, \quad (13.53, 13.54)$$

where $\{F_{\text{resist}, i}\}$ – vector of resistance forces; $\{F_{\text{resist}, i}\} = -\frac{F_{\text{resist}, i}}{\sqrt{\dot{x}_{c_i}^2 + \dot{y}_{c_i}^2}} \begin{Bmatrix} \dot{x}_{c_i} \\ \dot{y}_{c_i} \end{Bmatrix}$; \dot{x}_{c_i} , \dot{y}_{c_i} – projections of

velocity of motion of a body in the frame *X* - *Y*; $R_{x, i}$, $R_{y, i}$ – components of radius-vector from a rotation axis up to centre of masses of a body;

$$R_{x, 1} = L_1 \sin \varphi_1; \quad R_{y, 1} = -L_1 \cos \varphi_1; \quad R_{x, 2} = -L_1 \sin \varphi_1; \quad R_{y, 2} = L_1 \cos \varphi_1; \quad R_{x, 3} = L_5 \sin(\varphi_1 + \varphi_2);$$

$$R_{y, 3} = -L_5 \cos(\varphi_1 + \varphi_2); \quad R_{x, 4} = -L_6 \sin(\varphi_1 + \varphi_2); \quad (13.55)$$

$$R_{y, 4} = L_6 \cos(\varphi_1 + \varphi_2); \quad R_{x, 5} = L_7 \sin(\varphi_1 + \varphi_3); \quad R_{y, 5} = -L_7 \cos(\varphi_1 + \varphi_3);$$

$$R_{x, 6} = -L_8 \sin(\varphi_1 + \varphi_3); \quad R_{y, 6} = L_8 \cos(\varphi_1 + \varphi_3)$$

13.4.2. Input data for the solution of the problem

For the solution of a problem over investigation of *motion of the mechanical system* represented on the *fig. 13.12*, the *Pendulum* program is used. All data necessary for operation of this program should be recorded into a file in advance; therefore, variable **F**, necessary qualifier of the file, is ascribed. The data in the file are placed in the strict order, namely. In the *first* line the following parameters are coded: print code of intermediate results (*kprint*), number of integration steps (*ntime*), number indicating through how much of integration steps the results of evaluations are recorded (*nstep*), and integration step (*dtime*). If *kprint*=0, then the intermediate results are not printed; otherwise, they are printed out. In the *second* line, a line number and nine geometrical parameters L_i ($i=1,\dots$) of array **AL(9)** are coded.

In subsequent *six* lines, the number of a body and elements of arrays **BM(6)**, **R(6)** and **AH(6)** are coded. Into these arrays a mass, radius and a damping coefficient of each ball are recorded. In subsequent three lines are coded: the number of a constant magnet and also elements of arrays **DD(3)** and **DH(3)**, into which a *thickness* of a magnet (*parameter* Δ) and *distance* up to a magnet from a point **O_i** (*parameter* **hi**) are recorded accordingly. In the datafile between the recorded arrays the lines of the comments with names of appropriate arrays are located. At reading of the information from the datafile these text lines are skipped.

Schematic structure of the datafile:

Text line *
kprint, ntime, nstep, dtime
 Text line *
 Array **AL(7)**
 Text line *
 Arrays **BM(6), R(6), AH(6)**
 Text line *
 Arrays **DD(3), DH(3)**
 Text line *
 Array **Xvarb(6)**

13.4.3. Brief description of the Pendium program solving the problem

The *Pendium* program was programmed on the *Maple*-language; it consists of the basic program and 23 procedures. All procedures can be divided into *three* groups: procedures for data entry, for calculation and for output of results. Memory size necessary for the solution of a concrete problem, and a time of its solution depend on the used number of integration steps. The program *Pendium* calculates the *generalized* coordinates and velocities of motion of the mechanical system represented on the *fig. 13.11*, and also provides animation of a resultant motion of the system. The calculation results are output on the monitor and are recorded into files, for that to variables *file_rez1* and *file_rez2* of the program must be ascribed *qualifiers* of target files. Into the file *file_rez1* values of the *generalized* coordinates and velocities of motion are recorded. While into the file *file_rez2* the expressions for equations of motion of a mechanical system with the concentrated parameters are recorded.

13.4.4. An example of use of the Maple-program Pendium

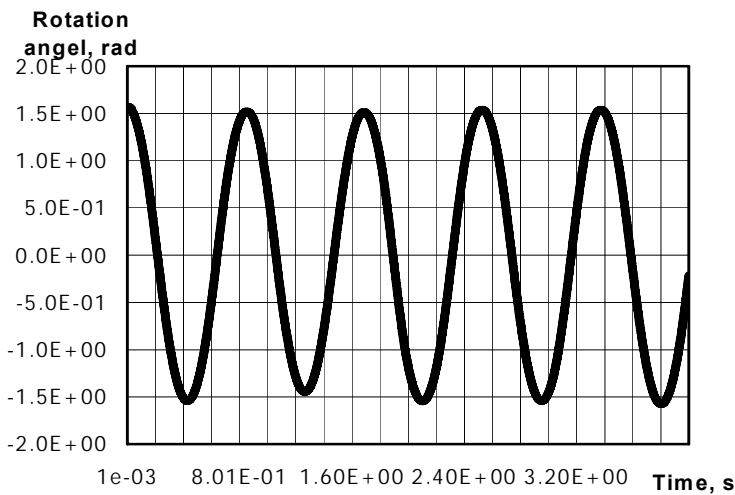
As an example of application of the *Pendium* program, the generalized coordinates and velocities of motion of the mechanical system indicated on the *fig. 13.11*, are determined.

Input data for the test example: a print code *kprint*=0, number of integration steps *ntime*=5000, *nstep* = 50, integration step *dtime*= 10^{-3} s. Geometrical parameters:

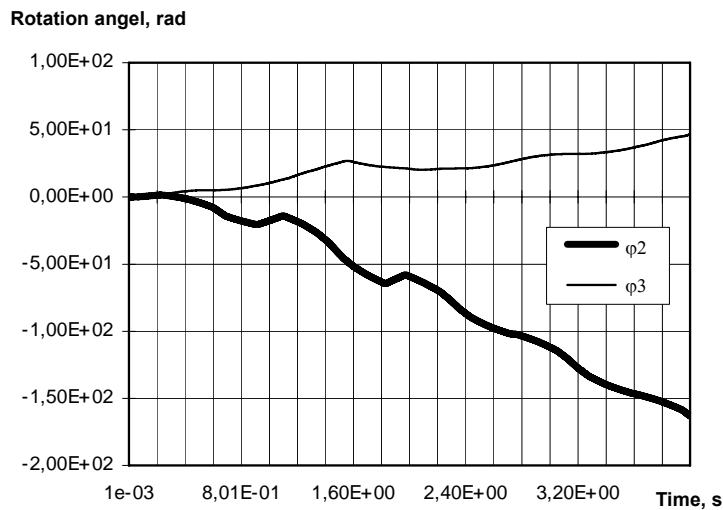
$$\begin{aligned}
 &AL(1)=0,070 \text{ m} ; AL(2)=0,085 \text{ m} ; AL(3)=AL(4)=0,040 \text{ m} ; \\
 &AL(5)=AL(6)=AL(7)=AL(8)=0,020 \text{ m} ; AL(9)=0,0975 \text{ m} ; BM(1)=0,050 \text{ kg} ; \\
 &BM(2)=0,010 \text{ kg} ; BM(3)=BM(4)=BM(5)=BM(6)=0,005 \text{ kg} ; R(1)=0,0125 \text{ m} ; \\
 &R(1)=0,050 \text{ m} ; R(3)=R(4)=R(5)=R(6)=0,010 \text{ m} ; \\
 &AH(1)=AH(2)=AH(3)=AH(4)=AH(5)=AH(6)=0 ; DD(1)=DD(2)=DD(3)=0,005 \text{ m} ; \\
 &DH(1)=0,005 \text{ m} ; DH(2)=DH(3)=0,010 \text{ m} ; Xvarb(1)=0,005 \text{ }^\circ ; \\
 &Xvarb(2)=Xvarb(3)=Xvarb(4)=Xvarb(5)=Xvarb(6)=0 .
 \end{aligned}$$

Results of calculation over the test example:

The dependences of rotation angles of masses and rotation velocities in a time are represented on the *figs.* 13.12 and 13.13, accordingly. On the *fig.* 13.14 four fragments (a, b, c, d) of motion of the explored mechanical system are represented.



(a)



(b)

Fig. 13.12. Dependences of rotation angles of masses of the explored mechanical system: (a) – angle φ_1 , and (b) – angles φ_2 and φ_3

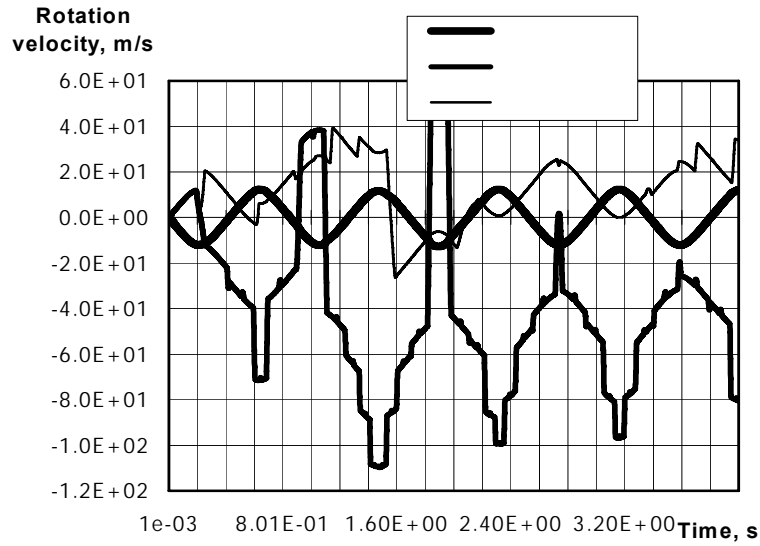


Fig. 13.13. Dependences of rotation velocities of masses of the explored mechanical system

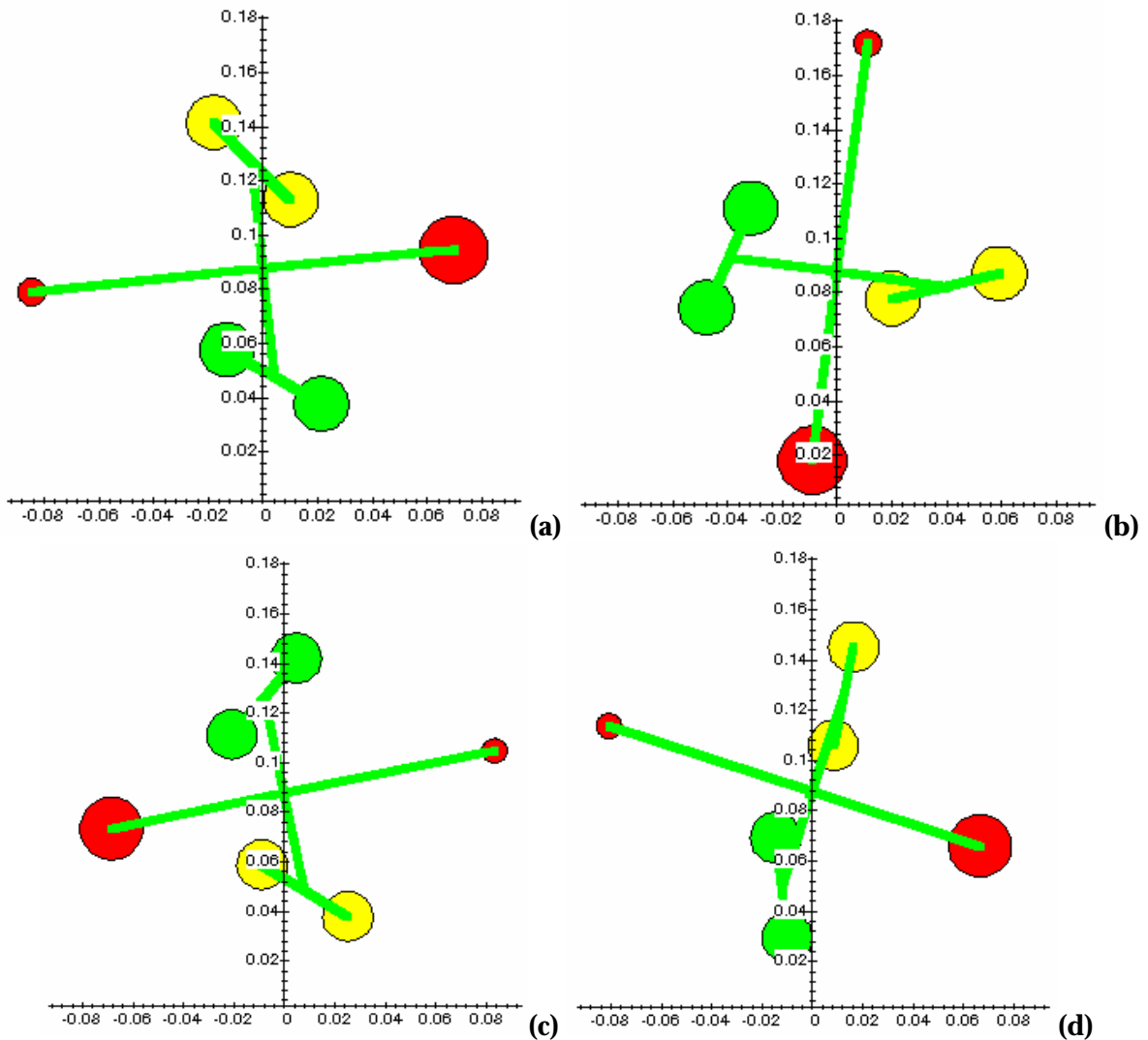


Fig. 13.14. Fragments of motion of the mechanical system

The obtained equations of motion of the explored system have the next form:

$$Eq_1 = .000357250 u_{tt1} + .400000 10^{-5} u_{tt2} + .400000 10^{-5} u_{tt3} + h_1 u_{t1} + .0259965 \sin(u_1) - 1. Qp_1 - 1. Qm_1 - 1. Qp_2 - 1. Qm_2$$

$$Eq_2 = .400000 10^{-5} u_{tt1} + .400000 10^{-5} u_{tt2} + h_2 u_{t2} - 1. Qp_3 - 1. Qm_3 - 1. Qp_4 - 1. Qm_4$$

$$Eq_3 = .400000 10^{-5} u_{tt1} + .400000 10^{-5} u_{tt3} - 1. Qp_5 - 1. Qm_5 - 1. Qp_6 - 1. Qm_6 ,$$

where \mathbf{u} – displacement; \mathbf{u}_t – velocity; \mathbf{u}_{tt} – acceleration; \mathbf{h} – coefficient of damping; \mathbf{Qp} – generalized force of resistance to motion of the i -th body in air; \mathbf{Qm} – generalized magnetic force acting onto a body. The *Pendium* program is intended for calculation of the *generalized* coordinates and velocities of motion of the mechanical system represented on the *fig.* 13.11, for deduction of equations of motion of the mechanical system with the concentrated parameters, and also for *animation* of motion of the explored mechanical system. The source module of the program in *Maple*-language, initial data for the test example, and also outcomes of its solution are represented in the PROBLEMS directory of archive attached to the present book in files Mb10_4_new.mws, Mb10_4.dat and Mb10_4_1.rez accordingly.

13.5. Calculation of transition torsion oscillations in a mechanical transmission

In technique the mechanical transmissions consisting from gears and shafts, which transmit a torsional moment from the drive to an operating mechanism, are widely used. The reason of torsional oscillations in shafts of transmission systems consists in the *nonuniformity* of the torsional moments from propellents and forces of resistance, which cause changes of angular velocity of rotation of the shaft, i.e., now acceleration, now deceleration of its rotation. The shaft has elasticity and the *concentrated masses* are placed on it, therefore in each cross-section of the shaft takes place own degree of nonuniformity.

The basic noisemaker and vibration in transmission systems (*for example, in transmissions*) is the knock of gears, which is characterized by vibrational blows in engagement of gears, which arise owing to a presence of a backlash between cogs. As a result of torsional oscillations, cyclic angular accelerations of details of a system of gears arise. When the torsional moment of inertia of a driven gear exceeds the moment of a loading, cogs of gears go away and blows arise. The continuous shocks called as vibrational blows, can reduce to excessive noise and major dynamical loadings onto cogs. Therefore, the calculation of torsional oscillations of mechanical transmissions consisting of the drive, shafts and gears with backlashes, represents major practical interest.

13.5.1. The calculated expressions for determination of torsional oscillations of mechanical transmissions

A mechanical transmission consisting of the asynchronous electric motor, coupler, shafts and gears is considered. The influence of the drive onto the dynamics of all driving gear can be considerable, therefore in dynamic model of a driving gear it is necessary to take into account the influence of the drive. At investigation of asynchronous drives the two-phase mathematical models adequately reflecting processes happening in the drive are used. The common form of *mathematical* description of two-phase models is the system of the differential and algebraic equations [110], namely:

$$\left\{ \frac{dx}{dt} \right\} = [A]\{x\} + \{F(\psi, t)\}; \quad \frac{d\omega}{dt} = \frac{(M_{eng} - M_{resist}) p}{I_n}; \quad (13.56)$$

$$\mathbf{M}_{\text{eng}} = 1,5pL_{\mu}a(\psi_{\alpha s}\psi_{\beta r} - \psi_{\alpha r}\psi_{\beta s}); \quad [\mathbf{A}] = \begin{bmatrix} -ar_sL_r & 0 & ar_sL_{\mu} & 0 \\ 0 & -ar_sL_r & 0 & ar_sL_{\mu} \\ ar_rL_{\mu} & 0 & -ar_rL_s & 0 \\ 0 & ar_rL_{\mu} & 0 & -ar_rL_s \end{bmatrix}; \quad \{\mathbf{x}\} = \begin{Bmatrix} \psi_{\alpha s} \\ \psi_{\beta s} \\ \psi_{\alpha r} \\ \psi_{\beta r} \end{Bmatrix};$$

$$\{\mathbf{F}(\psi, t)\} = \begin{Bmatrix} \sqrt{2} U_n \cos \omega_s t \\ -\sqrt{2} U_n \sin \omega_s t \\ \omega \psi_{\beta r} \\ -\omega \psi_{\alpha r} \end{Bmatrix}, \quad \text{where } a = \frac{1}{L_s L_r - L_{\mu}^2}; \quad L_{\mu} = 1,5 L_0; \quad L_s = L_{s\sigma} + 1,5 L_0;$$

$L_r = L_{r\sigma} + 1,5 L_0$; $L_{s\sigma}$, $L_{r\sigma}$ – diffusion inductance of stator and rotor; L_0 – inductance of a magnetizing contour; r_s , r_r – active resistance of stator and rotor; U_n – rated voltage; ω_s – angular frequency of a supply voltage; M_{eng} – moment of drive; p – number of pairs of poles; ω – angular frequency of rotation of drive rotor; I_n – moment of inertia.

A mathematical model of torsional oscillations of a mechanical transmission with the concentrated parameters is considered. The model represents a system consisting of shafts, coupler and gears. In the model of the general view there are three nonlinear elements of stiffness: many-stage stiffness of a coupler, backlash between splines of coupler and a backlash between *cogs* of a pair gears. The common nonlinear element $f_i(\delta_i)$ is represented on the following fig. 13.15.

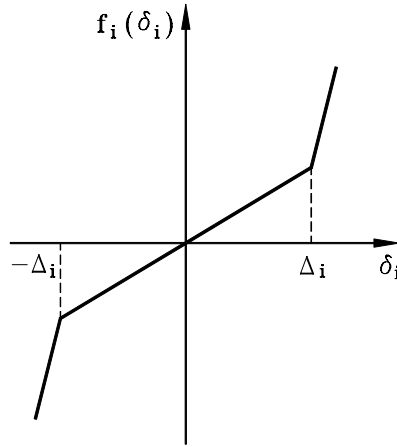


Fig. 13.15. Dependence of nonlinear force $f_i(\delta_i)$ from displacement δ_i

The equations of motion of *basic elements* of a mechanical drive unit, are considered, namely: *coupler* (fig. 13.16.a):

$$I_i \ddot{\varphi}_i = k_{i-1,i} f_{i-1}(\delta_{i-1}) R_{i,1} - k_{i,i+1} f_i(\delta_i) R_{i,2} - h_{i-1,i}(\dot{\varphi}_i - \dot{\varphi}_{i-1}) - h_{i,i+1}(\dot{\varphi}_i - \dot{\varphi}_{i+1});$$

$$I_{i+1} \ddot{\varphi}_{i+1} = k_{i,i+1} f_i(\delta_i) R_{i+1,2} - k_{i+1,i+2} f_{i+1}(\delta_{i+1}) R_{i+1,1} - h_{i,i+1}(\dot{\varphi}_{i+1} - \dot{\varphi}_i) -$$

$$- h_{i+1,i+2}(\dot{\varphi}_{i+1} - \dot{\varphi}_{i+2});$$

$$\delta_{i-1} = \varphi_{i-1} R_{i-1,1} - \varphi_i R_{i,1}; \quad \delta_i = \varphi_i R_{i,2} - \varphi_{i+1} R_{i+1,1}; \quad \delta_{i+1} = \varphi_{i+1} R_{i+1,1} - \varphi_{i+2} R_{i+2,1}, \quad (13.57)$$

where: I_i , I_{i+1} – moment of inertia of a half-coupler; $k_{i-1,i}$, $k_{i+1,i+2}$ – stiffnesses of shafts; $k_{i,i+1}$ – stiffness of a coupler; $h_{i-1,i}$, $h_{i,i+1}$, $h_{i+1,i+2}$ – coefficients of damping; φ_i , $\dot{\varphi}_i$, $\ddot{\varphi}_i$ – angular displacements, velocities and accelerations accordingly; δ_i – relative displacements; $R_{i,1}$, $R_{i+1,1}$ – radiuses of dividing

splices of a coupler; $R_{i,2}$, $R_{i+1,2}$ – the radiuses of circles, on which are located springs of a coupler.

The nonlinear function $f_i(\delta_i)$ of the coupler is defined by the next relation:

$$f_i(\delta_i) = \begin{cases} \delta_i - \Delta_i(1 - \alpha_i), & \delta_i > \Delta_i; \\ \alpha_i \delta_i, & -\Delta_i \leq \delta_i \leq \Delta_i; \\ \delta_i + \Delta_i(1 - \alpha_i), & \delta_i < -\Delta_i, \end{cases} \quad (13.58)$$

where: Δ_i – salient point for steps of nonlinearity; α_i – measure of nonlinearity.

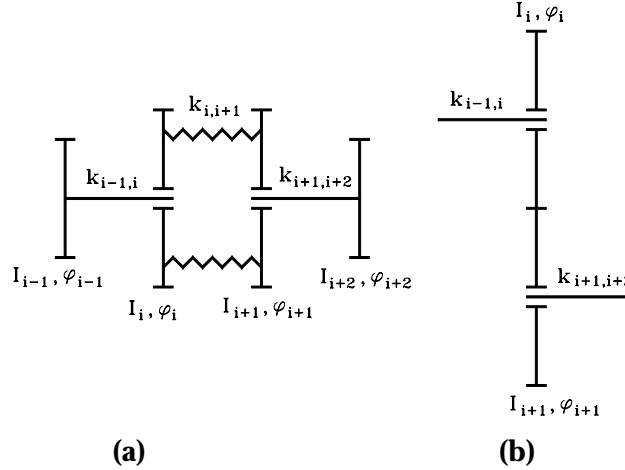


Fig. 13.16. Basic elements of mechanical transmission: (a) a coupler and (b) a pair of gears pair of gears (fig. 13.16.b):

$$I_i \ddot{\varphi}_i = k_{i-1,i} f_{i-1}(\delta_{i-1}) R_{i,1} - k_{i,i+1} f_i(\delta_i) R_{i,2} - h_{i-1,i} (\dot{\varphi}_i - \dot{\varphi}_{i-1}); \quad (13.59)$$

$$I_{i+1} \ddot{\varphi}_{i+1} = k_{i,i+1} f_i(\delta_i) R_{i+1,2} - k_{i+1,i+2} f_{i+1}(\delta_{i+1}) R_{i+1,1} - h_{i+1,i+2} (\dot{\varphi}_{i+1} - \dot{\varphi}_{i+2}),$$

where: I_i , I_{i+1} – moment of inertia of gears; $k_{i,i+1}$ – stiffness of a hitch; $h_{i-1,i}$, $h_{i,i+1}$, $h_{i+1,i+2}$ – coefficients of damping of shafts; $R_{i,1}$, $R_{i+1,1}$ – radiuses of dividing circles of splines; $R_{i,2}$, $R_{i+1,2}$ – radiuses of dividing circles of a leading and driven gears. The nonlinear function $f_i(\delta_i)$ for a backlash between cogs is defined as follows:

$$f_i(\delta_i) = \begin{cases} \delta_i - \Delta_i(1 - \alpha_i), & \delta_i > \Delta_i; \\ 0, & -\Delta_i \leq \delta_i \leq \Delta_i; \\ \delta_i + \Delta_i(1 - \alpha_i), & \delta_i < -\Delta_i. \end{cases} \quad (13.60)$$

Generally, the external torsional moment is defined by the following relation:

$$M_{resist} = M_{resist0} + M_{resist1} \sin(\omega_{resist} t), \quad (13.61)$$

where: $M_{resist0}$, $M_{resist1}$ – constant and variable part of a torsional moment. The parameters $M_{resist0}$, $M_{resist1}$ and ω_{resist} can be by constant values or independent random quantities, distributed under the normal law. The value of a random quantity y submitting to the normal distribution, is defined as follows:

$$y = \mu + \sigma \left(\sum_{i=1}^{12} R_i - 6 \right), \quad (13.62)$$

where: μ – average value; σ^2 – variance; R_i – random number from the interval $[0, 1]$. Then the common equations system of motion of the mechanical transmission with an electromotor accepts the following form:

$$\{\dot{\mathbf{x}}\} = [\mathbf{A}]\{\mathbf{x}\} + \{\mathbf{F}(t, \mathbf{x})\}; \quad [\mathbf{M}]\{\ddot{\phi}\} = \{\mathbf{G}(t, \phi, \dot{\phi}, \mathbf{x})\} \quad (13.63)$$

We solve a problem of eigenvalues of system of the linear equations without taking the equations of the electromotor into consideration.

13.5.2. Input data for the solution of the problem

For solution of a problem concerning of *definition of torsion oscillations of mechanical transmissions* the *Driver* program is used. All data necessary for operation of this program should be recorded into a file in advance; therefore, variable **F**, necessary qualifier of the file, is ascribed. The data in the file are placed in the strict order, namely.

In the *first* line the next parameters are recorded: number of bodies of a mechanical transmission (*parameter nbody*); the number of elements linking bodies of mechanical transmission (*nelem*); the number of the moments of resistance (*nmpas*); the number of degrees of freedom whose values of unknowns are recorded into the resultant file (*nwr*) and code of a loading (*kodpas*). If *kodpas*=1, then takes place a casual loading; otherwise – determinate loading.

In the *second* line the next parameters are recorded: a print code of intermediate results (*kprint*); number indicating through how much of integration steps the results of evaluations are recorded (*nstep*); number of iterations steps (*ntime*) and iteration step (*dtime*). If *kprint*=0, then the intermediate results are not printed; otherwise, they are printed out.

In each subsequent *nelem* lines, the elements of the array **Mtop**(*nelem*, 5) are coded. Into each line of the datafile the line and five elements of this array are recorded, namely: the number of the first and the second body, which are connected among themselves by an element of transmission, and also three numbers (0 or 1), which encode a type of linking of the bodies. The first number defines a splice linking, the second – a cog linking, the third – a *coupler*. The number 1 or 0 indicates presence of linking or its absence accordingly.

Behind of the array **Mtop** into the datafile the elements of the **Lpar**(*nbody*, 2) are recorded line by line. Into each line of the datafile the line number and elements of the array **Lpar** are recorded. Into each line of the array **Lpar** the code defining number of a body to which an electromotor is connected, and number of the external moment of resistance are recorded. If the electromotor is connected to the corresponding body, the code equals 1, otherwise – 0. If onto the corresponding body the external moment of resistance does not act, then into the *second* column of the array **Lpar** the value 0 is recorded; otherwise, the serial number of the external moment of resistance is recorded.

Behind of the array **Lpar** into the datafile the elements of the **Par**(*nbody*, 10) array are recorded line by line. Into each line of the datafile the line number and elements of the **Par** array are recorded. Into each line of this array are recorded the following parameters: moment of inertia of mass I_i ; radius of dividing circle of splices $R_{i,1}$; a salient point of a linking $\Delta_{i,1}$; a measure of nonlinearity $\alpha_{i,1}$ of splice linking; radius of dividing circle of gears $R_{i,2}$; salient point of linkings $\Delta_{i,2}$; a measure of nonlinearity $\alpha_{i,2}$ of cog linking; radius of a circle, on which the springs of a coupler $R_{i,3}$ are located; a salient point of linking $\Delta_{i,3}$ and also measure of nonlinearity $\alpha_{i,3}$ of coupler linking.

Behind of the array **Par** into the datafile the elements of the **Par1**(*nelem*, 2) array are recorded line by line. Into each line of the datafile the line number and elements of the **Par1** array are recorded. Into each line of this array the coefficients of stiffness and damping of an element of linking are recorded.

Behind of the array **Par1** into the datafile the elements of the **Par_m**(*nmpas*, 6) array are recorded line by line. Into each line of the datafile the line number and elements of the **Par_m** array are recorded. Into each line of this array are recorded the following parameters: the line number; average value and mean square deviation of a constant component of the moment of resistance; average value and mean square deviation of amplitude of the moment of resistance; average value and mean square deviation of frequency of the moment of resistance. The line number of the array **Par_m** must correspond to the number of an linkage element.

Behind of the array **Par_m** into the datafile the elements of the **Par_v**(7) array are recorded line by line. Into each line of the datafile the line number and elements of the **Par_v** array are recorded. Into the array **Par_v** the parameters of an asynchronous electromotor are recorded, namely: rated voltage U_n ; number of pairs of poles p ; an inductance dispersion of a stator winding $L_{s\sigma}$; the reduced inductance dispersion of a rotor winding $L_{r\sigma}$; inductance of a magnetizing contour L_0 ; active resistance of stator r_s and rotor r_r .

Behind of the array **Par_v** into the datafile the elements of the **Mwr**(*nwr*) array are recorded line by line. In each line of the datafile the line number and also the number of a degree of freedom (*number of a body*), the values of angular displacements, velocities and accelerations of which are recorded into the resultant file, are coded.

Behind of the array **Mwr** into the datafile the elements of the array **Xvarb**(*neq*) {*neq* – number of unknowns, *neq*=4+2**nbody*} are recorded line by line. In each line of the datafile the line number of the array and also an initial value of an unknown are coded. In the datafile between the recorded arrays the lines of the comments with names of appropriate arrays are located. At reading of the information from the datafile these text lines are skipped.

Schematic structure of the datafile:

Text line *
nbody, nelem, nmpas, nwr, kodpas, kprint, nstep, ntime, dtime
Text line *
Array **Mtop**(*nelem*, 5)
Text line *
Array **Lpar**(*nbody*, 2)
Text line *
Array **Par**(*nbody*, 10)
Text line *
Array **Par1**(*nelem*, 2)
Text line *
Array **Par_m**(*nmpas*, 6)
Text line *
Array **Par_v**(7)
Text line *
Array **Mwr**(*nwr*)
Text line *
Array **Xvarb**(*neq*)

13.5.3. Brief description of the Driver program solving the problem

The *Driver* program was programmed on the *Maple*-language; it consists of the basic program and 17 procedures. All procedures can be divided into two groups: procedures for data entry and for calculations. Memory size necessary for the solution of a concrete problem, and a time of its solution depend on the used number of bodies of the concrete mechanical transmission. The program calculates values of angular displacements, velocities and accelerations of bodies of a mechanical transmission in a time dependence, and also eigenvalues and own frequencies. The calculation results are output on the monitor and are recorded into files, for that to variables *file_rez1*, *file_rez2*, *file_rez3*, *file_rez4* and *file_rez5* of the program must be ascribed the qualifiers of target files. Into these files the following data are recorded: into *file_rez1* – values of *angular displacements* of bodies of a mechanical transmission, into *file_rez2* – values of *angular velocities*, into *file_rez3* – values of *angular accelerations*, into *file_rez4* – values of of a *rotational moment* of the electromotor, and into *file_rez5* – *eigenvalues* and *own frequencies*.

13.5.4. An example of use of the Maple-program Driver

As an example of application of the *Driver* program, the mechanical transmission represented on the *fig. 13.17* and consisting from asynchronous electromotor **4A100/4SY3**, of a coupler and two cog transmissions is considered.

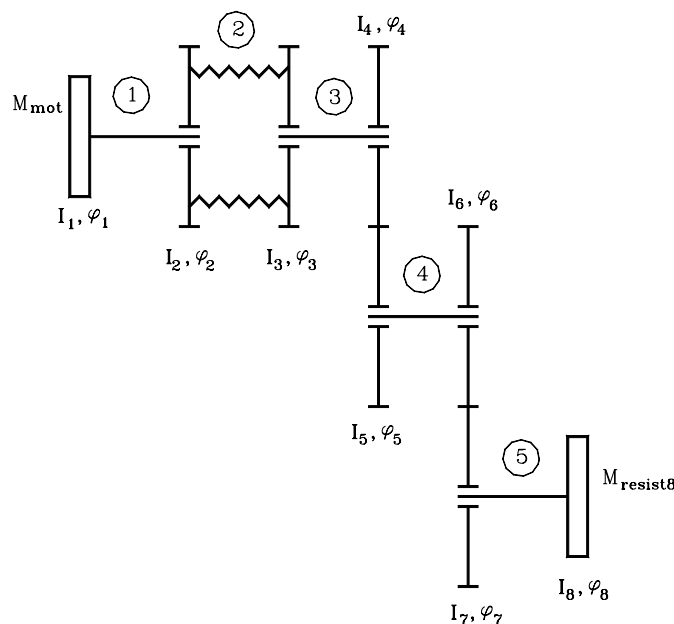


Fig. 13.17. The scheme of explored mechanical transmission

Input data for the test example: number of the bodies *nbody*=8; the number of elements linking the bodies *nelem*=5; number of nodes *npoint*=44; number of the moments of resistance *nmpas*=1; number of degrees of freedom, whose values of unknowns are recorded into the resultant file *nwr*=3; a code of a loading *kodpas*=0; *kprint*=1; *nstep*=2; *ntime*=5000 and *dtime*= $2 \cdot 10^{-4}$ s. The moment of resistance acting to the eighth body of the system:

$M_{\text{resist}0} = 50 \text{ Nm}$, $M_{\text{resist}1} = 10 \text{ Nm}$, $\omega_{\text{resist}0} = 62,8 \text{ s}^{-1}$. Coefficients of elements stiffness:
 $k_1 = 188,4 \text{ MN/m}$; $k_2 = 1,02 \text{ MN/m}$; $k_3 = 110,0 \text{ MN/m}$; $k_4 = 83,9 \text{ MN/m}$;
 $k_5 = 323,3 \text{ MN/m}$; $k_6 = 146,4 \text{ MN/m}$; $k_7 = 8,28 \text{ MN/m}$. Coefficients of elements damping:
 $h_i = 10^{-5} \text{ Ns/m}$, $i=1..7$. Initial values of unknowns: $x_i = \dot{x}_i = 0$, $i=1..7$. Backlashes of the size:

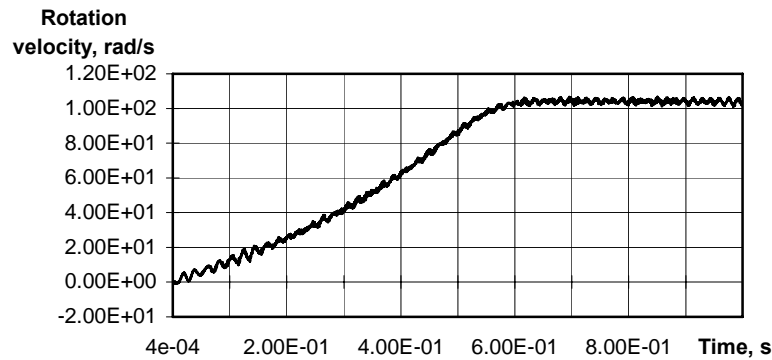
$\Delta_{6-7} = 10^{-4} \text{ m}$; $\Delta_8 = 2 \cdot 10^{-4} \text{ m}$. Measures of nonlinearity: $\alpha_{6-7} = 0$; $\alpha_8 = 0$.

Results of calculation over the test example:

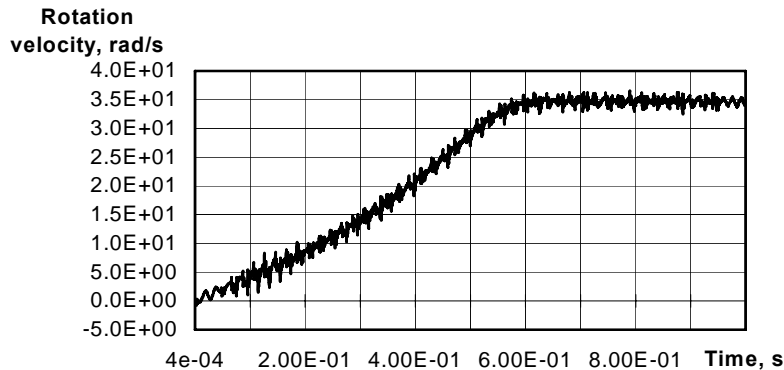
Eigenvalues (real and imaginary part) and own frequencies:

| | | |
|-----------------------|----------------------|-------------------------------|
| Re[1]= -2.817913e-06 | Im[1]=8.965371e+03 | frequency[1]=1.506968e+01 Hz |
| Re[2]= -2.817913e-06 | Im[2]=-8.965371e+03 | frequency[2]=1.506968e+01 Hz |
| Re[3]= -1.660601e-06 | Im[3]=6.102317e+03 | frequency[3]=1.243275e+01 Hz |
| Re[4]= -1.660601e-06 | Im[4]=-6.102317e+03 | frequency[4]=1.243275e+01 Hz |
| Re[5]= -1.149593e-06 | Im[5]=2.763744e+03 | frequency[5]=8.366986e+00 Hz |
| Re[6]= -1.149593e-06 | Im[6]=-2.763744e+03 | frequency[6]=8.366986e+00 Hz |
| Re[7]= -2.51011e-06 | Im[7]=2.613511e+03 | frequency[7]=8.136400e+00 Hz |
| Re[8]= -2.51011e-06 | Im[8]=-2.613511e+03 | frequency[8]=8.136400e+00 Hz |
| Re[9]= -1.207071e-06 | Im[9]=2.080419e+03 | frequency[9]=7.259313e+00 Hz |
| Re[10]= -1.207071e-06 | Im[10]=-2.080419e+03 | frequency[10]=7.259313e+00 Hz |
| Re[11]= 2.799271e-18 | Im[11]=0e-01 | frequency[11]=0e-01 Hz |
| Re[12]= -3.939402e-07 | Im[12]=3.699337e+02 | frequency[12]=3.061132e+00 Hz |
| Re[13]= -3.939402e-07 | Im[13]=-3.699337e+02 | frequency[13]=3.061132e+00 Hz |
| Re[14]= -3.605612e-07 | Im[14]=4.320143e+02 | frequency[14]=3.308028e+00 Hz |
| Re[15]= -3.605612e-07 | Im[15]=-4.320143e+02 | frequency[15]=3.308028e+00 Hz |
| Re[16]= 0e-01 | Im[16]=0e-01 | frequency[16]=0e-01 Hz |

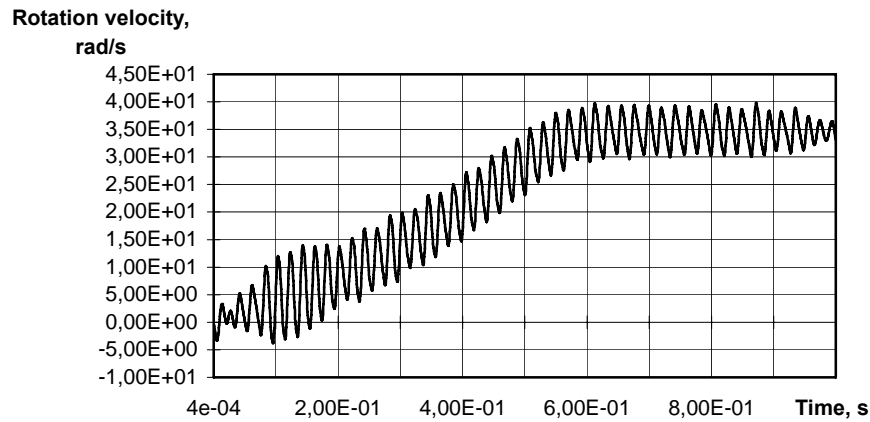
The *fig.* 13.18 represents the dependences of angular velocities of bodies **6**, **7** and **8**.



(a)



(b)



(c)

Fig. 13.18. Dependences of angular velocities of the body 6 (a), 7 (b) and 8 (c) accordingly

The *Driver* program is intended for the solution of a problem over definition of torsion oscillations of mechanical transmissions; the angular displacements, velocities and accelerations of bodies in a time dependence, and also *eigenvalues* and *own frequencies* are calculated. The source module of the program in *Maple*-language, initial data for the test example, and also outcomes of its solution are represented in the PROBLEMS directory of archive attached to the book in files Mb10_5_new.mws, Mb10_5.dat and Mb10_5_1.rez accordingly.

References

1. Aladjev V.Z. *To the Theory of the Homogeneous Structures*.- Tallinn: Estonian Academy Press, 1972
2. Aladjev V.Z., Osipov O.B. *Introduction to Operating System of IBM/360 Computers*, Vol. 1-2.- Tallinn: CPTI V/O Sojuztorgsystem Press, 1975, 276 p. (in Russian with extended English summary)
3. Aladjev V.Z., Osipov O.B. *Introduction to Architecture of Computers IBM/360*.- Tallinn: Valgus Press, 1976, 332 p. (in Russian with extended English summary)
4. Aladjev V.Z. *Mathematical Theory of Homogeneous Structures and Their Applications*.- Tallinn: Valgus Press, 1980, 267 p.
5. Aladjev V.Z. et al. *Mathematical Developmental Biology*.- Moscow: Science Press, 1982, 254 p.
6. Aladjev V.Z. *Architecture and Software of PDP-11 Computers*.- Tallinn: IVC Gosbanka USSR, 1983
7. Aladjev V.Z. *Lecture on Personal Computer WANG 2200MVP*.- Tallinn: SKB MPCM ESSR, 1986
8. Aladjev V.Z., Shirodza I.B. *Solution of Engineering Problems in Operating Media of Language Basic for PC WANG 2200MVP*.- Harkov: HAI Press, 1988, 128 p. (in Russian with English summary)
9. Aladjev V.Z. et. al. *Personal Computer WANG 2200MVP*.- Kiev: USE Press, 1989, 150 p.
10. Aladjev V.Z. et. al. *Design of Software for PC WANG 220MVP*.- Kiev: Technika Press, 1989, 255 p.
11. Aladjev V.Z., Shilenko V.F. *Personal Computer IBM PC/XT*.- Kiev: USE Press, 1990, 490 p.
12. Aladjev V.Z. *Homogeneous Structures: Theoretical and Applied Aspects*.- Kiev: Technika Press, 1990
13. Aladjev V.Z., Shirodza I.B. *Personal Computers IBM PC/XT: Instrumental Tools and Software Design*.- Moscow: Higher School Press, 1991, 457 p. (in Russian with extended English summary)
14. Aladjev V.Z. *Scientific-Technical CADs*.- Kiev: USE Press, 1991, 432 p.
15. Aladjev V.Z., Gershgorin N.A. *Computational Problems on Personal Computer*.- Kiev: Technika Press, 1991, 248 p., ISBN 5-335-01064-9 (in Russian with extended English summary)
16. Aladjev V.Z. et. al. *Utilities for Personal Computer*.- Kiev: Technika Press, 1992, 152 p.
17. Aladjev V.Z., Shirodza I.B. *Computer Mixture*.- Tallinn: CAPT Press, 1992, 355 p.
18. Aladjev V.Z., Tupalo V.G. *Computer Reader*.- Kiev: USE Press, 1993, 448 p.
19. Aladjev V.Z., Tupalo V.G. *Turbo-Pascal for All*.- Kiev: Technika Press, 1993, 185 p.
20. Aladjev V.Z., Tupalo V.G. *Scientific Editing and Statistics on Personal Computer*.- Moscow: Mintopenergo Press, 1993, 106 p. (in Russian with extended English summary)
21. Aladjev V.Z., Tupalo V.G. *Computer Telecommunication*.- Moscow: Mintopenergo Press, 1993
22. Aladjev V.Z., Tupalo V.G. *Algebraic Computations on Computer*.- Moscow: Mintopenergo, 1993
23. *Software of IBM/360 Computers and CACS* // Ed. V.Z. Aladjev.- Tallinn: Valgus Press, 1978
24. *A Data Base Management System on the Basis of Operating System MINIOS and DBMS OKA* // Ed. V.Z. Aladjev.- Tallinn: Valgus Press, 1980, 180 p. (in Russian with extended English summary)
25. *Parallel Processing and Parallel Algorithms* // Ed. V.Z. Aladjev.- Tallinn: Valgus Press, 1981, 320 p.
26. *Parallel Processing Systems* // Ed. V.Z. Aladjev.- Tallinn: Valgus Press, 1983, 376 p.
27. *Structurally-Analytical Models and Algorithms for Recognition and Identification of Control Objects* // Ed. V.Z. Aladjev.- Kiev: Technika Press, 1993, 386 p. (in Russian with extended English summary)
28. Aladjev V.Z., Hunt U.J., Shishakov M.L. *Mathematics on Personal Computer* // Ed. A.D. Ursul.- Gomel: BELGUT Press, 1996, 498 p., ISBN 3-420-614023-3 (in Russian with English summary)
29. Aladjev V.Z., Hunt U.J., Shishakov M.L. *Fundamentals of Computer Informatics* // Ed. A.D. Ursul.- Gomel: TRG & Salcombe Esti Ltd. & Russian Academy of Noosphere, 1997, 498 p.
30. Aladjev V.Z., Hunt U.J., Shishakov M.L. *Basics of Informatics. Text-book*. 2nd edition.- Moscow: FILIN Press, 1999, 525 p. (in Russian with extended English summary)

31. *Efimova M.O., Hunt U.J. Geometry of Drawing: Graphical Package AutoTouch* // Ed. V.Z. Aladjev.- Gomel: VASCO & Salcombe Eesti Ltd. & TRG, 1997, 72 p. (in Russian with English summary)
32. *Aladjev V.Z., Shishakov M.L. Introduction to Media of Package Mathematica 2.2.*- Moscow: FILIN Press, 1997, 363 p., ISBN 5-89568-004-6 (in Russian with extended English summary)
33. *Aladjev V.Z., Vaganov V.A., Hunt U.J., Shishakov M.L. Introduction to Media of Mathematical Package Maple V.*- Gomel: Russian Academy of Noosphere, 1998, 470 p.
34. *Hunt U.J., Shishakov M.L. Theory of Probability and Mathematical Statistics* // Ed. V.Z. Aladjev.- Gomel: Salcombe Eesti Ltd. & TRG & Russian Academy of Noosphere, 1997, 120 p.
35. *Aladjev V.Z., Hunt U.J., Shishakov M.L. Mathematical Theory of the Classical Homogeneous Structures.*- Gomel: TRG & VASCO & Salcombe Eesti Ltd., 1998, 300 p., ISBN 9-063-56078-9
36. *Aladjev V.Z., Hunt U.J., Shishakov M.L. Questions of Mathematical Theory of the Classical Homogeneous Structures.*- Gomel: BELGUT Press, 1997, 151 p., ISBN 5-063-56078-5
37. *Aladjev V.Z., Veetyusme R.A., Hunt U.J. General Theory of Statistics.*- Tallinn: TRG & SALCOMBE Eesti Ltd., 1995, 201 p. (in Russian with extended English summary)
38. *Aladjev V.Z., Hunt U.J., Shishakov M. Course of the General Theory of Statistics* // Ed. A. D. Ursul.- Gomel: BELGUT Press, 1995, 201 p. (in Russian with extended English summary)
39. *Aladjev V.Z., Tupalo V.G. Scientific and Practical Activity of the TRG: Final Results During the 25 Years (1969 - 1993).*- Moscow: Mintopenergo Press, 1994, 116 p. (in Russian with English summary)
40. *Aladjev V.Z., Hunt U.J., Shishakov M.L. Research Activity of the Tallinn Research Group: Scientific Report for Period 1995-1998.*- Tallinn-Gomel-Moscow: Russian Academy of Noosphere & Salcombe Eesti, 1988, 80 p. (in Russian with extended English summary)
41. *Aladjev V.Z., Vaganov V.A., Hunt U.J., Shishakov M.L. Programming in Mathematical Package Maple V.*- Tallinn-Gomel-Moscow: TRG, 1999, 470 p. (in Russian with extended English summary)
42. *Heck A. Introduction to Maple.*- Berlin-Heidelberg: Springer-Verlag, 1997, 699 p.
43. *Redfern D. The Maple Handbook.*- Berlin-Heidelberg: Springer-Verlag, 1996
44. *Burkhardt W. First Steps in Maple.*- Berlin-Heidelberg: Springer-Verlag, 1993
45. *Corless R.M. Essential Maple.*- Berlin-Heidelberg: Springer-Verlag, 1995
46. *Kamerich E. A Guide to Maple V.*- Berlin-Heidelberg: Springer-Verlag, 1995, 235 p.
47. *Rosen K. Exploring Discrete Mathematics with Maple.*- New York: McGraw-Hill, 1997
48. *Klimek G. Discovering Curves and Surfaces with Maple.*- Uppsala: Springer, 1997
49. *Zenkevich O. Finite Element Method in Engineering.*- Moscow: Mir Press, 1975
50. *Lojtsjansky L.G. Mechanics of Liquid and Gas.*- Москва: Nauka Press, 1987, 840 p.
51. *Bogdevicius M., Spruogis B. Theoretical researches of drive systems of transport machines with elastic link* // Transportas, Vilnius: Technika Press, 1996, no. 2 (13), pp. 70-81.
52. *Bezuhov N.I. Basics of the theory of elasticity, plasticity and creep.*- Moscow, 1968
53. *Zube R., Glebus S., Bogdevicius M. Theoretical Analysis of the Cothode Clearing Parameters of the Electrode Wire for Aluminium Alloys* // Mechanical Engineering, Vilnius: Technika Press, 1994, no. 1, pp. 122-126.
54. *Bogdevicius M. Berechnung instationarer Stromungen in elastisch-plastischen und viskoplastischen Rohrleitungen* // Universitat Stuttgart, 1991, 48 p. (in Germany with English summary)
55. *Bogdevicius M. Non-stationary current of liquid in elastic and ductile-plastic pipelines* // Applied mechanics, Kaunas, 1993, pp. 117-124 (in Lithuanian with English summary)
56. *Bogdevicius M. Simulation of hydraulic system of pumps by method of the characteristics* // Transportas, Vilnius: Technika Press, 1997, no. 2 (15).
57. *Char B. et al. Using Maple.*- Philadelphia: TELOS, Springer-Verlag, 1998
58. *Cornil J., Testud P. Maple V Release 4.*- Versailles: Springer-Verlag, 1997
59. *Dumas P., Courdon X. Maple.*- Le Chesnat: Springer-Verlag, 1997, 460 p.

60. *Gander W., Hrebicek J. Solving Problems in Scientific Computing Using Maple and MatLab.*- Zurich: Springer-Verlag, 1997, 408 p.
61. *Heal K. et al. Maple V: Learning Guide.*- Waterloo: Waterloo Maple Inc., 1996
62. *Monagan M. et al. Maple V: Programming Guide.*- Waterloo: Waterloo Maple Inc., 1996
63. *Algebraic Methodology and Software Technology*// Ed. **M. Johnson.**- Sydney, 1997
64. *Recent Trends in Algebraic Development Techniques*// Ed. **F. Parisi-Puscice.**- Roma: Springer, 1998
65. *Graph Drawing*// Ed. **G. Di Battista.**- Roma: Springer-Verlag, 1997, 448 p.
66. *Betounes D. Partial Differential Equations for Computational Science Analysis/ With Maple and Vector Analysis.*- Hattiesburg: Springer-Verlag, 1998, 530 p.
67. *Deebe E., Gunawardena A. Interactive Linear Algebra with Maple V.*- Houston, 1998
68. *Emtsev B.G. Technical Hydromechanics.*- Moscow: Nauka Press, 1978, 462 p.
69. *Paskonov V.M., Polezhaev V.I., Chudov L.A. The Numerical Simulation of Processes of Heat Exchange and Mass Transfer.*- Moscow: Nauka Press, 1984, 288 p.
70. *Ershov N.F., Shahverdi G. Finite Element Method in Problems of Hydrodynamics and Hydroelasticity.*- Leningrad: Nauka Press, 1984, 236 p. (in Russian with extended English summary)
71. *Vol'mir A.C., Kurapov B.A., Turbaitvsky A.T. Statics and Dynamics of Complex Structures: Applied Multilevel Research Techniques.*- Moscow: Nauka Press, 1989 (in Russian with English summary)
72. *Polezhaev V.I. et al. Mathematical Simulation of Convective Heat Exchange and Mass Transfer on the basis of the Navier-Stokes's equations.*- Moscow: Nauka Press, 1987
73. *Litvinov V.G. Flow of Nonlinear Viscous Liquid.*- Moscow: Nauka Press, 1982
74. *Isachenko V.P., Osipova V.A., Sukotel A.C. Heat Transfer.*- Moscow: Energy Press, 1975
75. *Seegerlind L. Application of Finite Element Method.*- Moscow: Nauka Press, 1979
76. *Bothe K.J., Wilson E.L. Numerical Methods in Finite Element Analysis.*- London, 1976
77. *Bogdevicius M. Finite Element Method in Analysis of Rotating Pipe Conveying Fluid*// *Lithuanian Journal of Computational Mechanics*, vol. 33, 1994, pp. 56-65.
78. *Scientific Computing*// Books-Journals-Electronic Media.- N.Y: Springer, 1997/1998
79. *Gander W. Solving Problems in Scientific Computing Using Maple V and MatLab.*- Berlin: Springer-Verlag, 1997, 408 p.
80. *Govoruhin V., Tsybulin V. Introduction to Maple: Mathematical Package for All.*- Moscow: Mir Press, 1997 (in Russian with extended English summary)
81. *Prohorov G. et al. Symbolic Calculations with Maple.*- Moscow: Petit Press, 1997
82. *Flegontov A., Zajtsev V. Discretely-Group Methods of Integration of the Ordinary Differential Equations.*- Leningrad: Automation Institute, 1991 (in Russian with extended English summary)
83. *Abramowitz M., Stegun I. Handbook of Mathematical Functions.*- New York: Dover, 1965
84. *Spanier J., Oldham K. An Atlas of Functions.*- Washington: Hemisphere, 1987.
85. *Bathe K.J., Wilson E.L. Numerical Methods in Finite Element Analysis*, New York, 1976.
86. *Malinin N.N. The Applied Theory of Plasticity and Creep.*- Moscow, 1975, 399 p.
87. *Morozov E.M., Nikishkov G.P. Finite Element Method in Mechanics of Corrupting.*- Moscow, 1980
88. *Marlet R., Marcal P. Finite element analysis of nonlinear structures / J. Struct. Div. Proc. ASCE*, 94, 1968, pp. 2081-2105
89. *Waszczyszczyn Z., Cichon Cz., Radwanska M. Metoda elementow skonchonych w statecznosc konstrukcji.*- Warszawa, 1990 (in Polish with English summary)
90. *Nicolaidis R. Maple V: A Comprehensive Introduction.*- Cambridge Univ. Press, 1996
91. *Judson P. Calculus Explorations Usung Maple.*- Saunders College Publishing, 1996
92. *Zwillinger D. Standard Maple Interactive.*- CRC Press, 1998
93. *Zachary J. Introduction to Scientific Programming: Computational Problem Solving Using Maple and C.*- TELOS, 1996

94. *Tocci C., Adams S. Applied Maple for Engineers and Scientists.*- Artech House, 1996
95. *Robertson J. Engineering Mathematics with Maple.*- N.Y.: McGraw-Hill, 1996
96. *Parker R. Maple for Basic Calculus.*- Delmar Publishing, 1997
97. *Lopez R. Maple V: Mathematics and Its Applications.*- Berlin: Birkhäuser, 1994
98. *Karian Z. Probability and Statistics Explorations with Maple.*- N.Y.: Prentice Hall, 1995
99. *Etchells T. Mathematical Activities with Computer Algebra.*- Chartwell-Bratt, 1997
100. *Borwein J. et al. Interactive Math Dictionary.*- Berlin: Springer-Verlag, 1998
101. *Abell M., Braselton J. Maple V by Examples.*- N.Y.: Academic Press, 1994
102. *Heal K.M. et al. Maple V Learning Guide for Release 5.*- Berlin: Springer, 1997
103. *Monagan M. et al. Maple V Programming Guide for Release 5.*- Springer, 1997
104. *Anderson G. Applied Mathematics with Maple.*- London: Chartwell-Bratt, 1997
105. *Beitzer R. Engineering Analysis with Maple/Mathematica.*- Academic Press, 1995
106. *Favorin M.V. Moments of Inertia of Bodies.*- Moscow: Nauka Press, 1977. 509 p.
107. *Vershinsky S.E. et al. Dynamics of Cars.*- Moscow, 1978, 352 p.
108. *Gerts E.V. Dynamics of Pneumatic Systems.*- Moscow, 1985, 256 p.
109. *Bogdevicius M. Dynamic Characteristics of Vibration-absorbing Elements with Permanent Magnets // Second Intern. Conf. of Mech. Engineering.*- Vilnius, 1997, 2, pp. 5-28.
110. *Basharin A.V., Postnikov Ju.V. Examples of Calculation of Automated Drive on Computers.*- Leningrad, 1990, 512 p. (in Russian with extended English summary)
111. *Aladjev V.Z., Bogdevicius M.A. Application of the package Maple V for solution of physical and technical problems // Internat. Conf. TRANSBALTICA-99, April 1999, VGTU, Vilnius*
112. *Aladjev V.Z., Hunt U.J. A Workstation for mathematician // Int. Conf. TRANSBALTICA-99, April 1999, VGTU, Vilnius, pp. 392-395. (in Russian with extended English summary)*
113. *Aladjev V.Z., Hunt U.J. A Workstation for mathematician // Internat. Conf. "Perfecting of mechanisms of management", April 1999, ISZ, Grodno (in Russian with extended English summary)*
114. *Aladjev V.Z., Shishakov M.L. Programming in package Maple V // 2nd Intern. Conf. "Computer Algebra in Fundamental and Applied Researches and Education".- Minsk: BSU, 1999, pp. 10-12.*
115. *Aladjev V.Z., Shishakov M.L. Automated working place of the mathematician // 2nd Intern. Conf. "Computer Algebra in Fundamental and Applied Researches and Education".- Minsk: BSU, 1999*
116. *Aladjev V.Z., Shishakov M.L., Trohova T.A. Educational computer laboratory of the engineer // Proc. of the 8-th Byelorussian mathematical conference, vol. 3.- Minsk: BSU, 2000, pp. 154-162.*
117. *Aladjev V.Z., Shishakov M.L., Trohova T.A. Applied aspects of the theory of homogeneous structures // Proc. of the 8-th Byelorussian mathematical conference, vol. 4.- Minsk: BSU, 2000*
118. *Aladjev V.Z., Shishakov M.L., Trohova T.A. Modelling in the classical homogeneous structures // Proc. of Internat. Conf. on Mathematical Modelling (MKMM-2000).- Herson, Ukraine, 2000*
119. *Aladjev V.Z., Shishakov M.L., Trohova T.A. Modeling in mathematical package Maple V // Proc. of Internat. Conf. on Mathematical Modelling (MKMM-2000).- Herson, Ukraine, 2000*
120. *Aladjev V.Z., Shishakov M.L., Trokhova T.A. Workstation for solution of systems of differential equations // 3rd Int. Conf. "Differential Equations and Applications".- St.-Petersburg, 2000*
121. *Aladjev V.Z., Shishakov M.L., Trokhova T.A. Computer laboratory for engineering researches // International Conference ACA-2000.- St.-Petersburg, 2000*
122. *Aladjev V.Z., Bogdevicius M.A., Hunt U.J. A workstation for mathematicians // Lithuanian conference TRANSPORT-2000.- Vilnius: Technika Press, 2000, pp. 274-282.*
123. *Aladjev V.Z. Interactive Course of General Theory of Statistics.*- Tallinn: The Intern. Academy of Noosphere, The Baltic Branch, 2000, 300 p. + 2 diskettes for Windows, ISBN 9985-60-866-6
124. *Aladjev V.Z., Bogdevicius M.A. Solution of Physical and Technical, and Mathematical Problems with Package Maple V.*- Vilnius: Technika Press, 1999, 686 p., ISBN 9986-05-398-6

125. *Aladjev V.Z., Bogdevicius M.A. Maple 6: Solution of Engineering and Physical, Statistical and Mathematical Problems.* - Moscow: Laboratory of Base Knowledge, 2001, 800 p. + CD
126. *Aladjev V.Z. Computer Algebra// Alpha*, no. 1 (April), 2001: The lecture given in *Institute of Modern Knowledge and the Grodno State University, October - November 2000*, Grodno, Byelorussia; lecture text is presented on CD [127].
127. *Aladjev V.Z., Shishakov M.L. A Workstation for Mathematicians.* - Moscow: Laboratory of Base Knowledge, 2000, 751 p. + CD, ISBN 5-93208-052-3 (in Russian with extended English summary)
128. *Bogdevicius M.A. Simulation of Dynamic Processes in Hydraulic, Pneumatic and Mechanical Drivers and Their Elements/* Research Report no. 541.- Vilnius: Technika, 2000
129. *Maple 6: Installation Guide.*- Waterloo: Waterloo Maple Inc., 2000, ISBN 1-894511-02-6
130. *Heal K. et al. Maple 6: Learning Guide.*- Waterloo: Waterloo Maple Inc., 2000
131. *Monagan M. et al. Maple 6: Programming Guide.*- Waterloo: Waterloo Maple Inc., 2000
132. *Tan K.S. et al. Symbolic C++: An Introduction to Computer Algebra Using Object-Oriented Programming.*- Berlin: Springer, 2000, 672 p., ISBN 1-85233-260-3
133. *Gerhard J. et al. MuPAD Tutorial: A Version and Platform Independent Introduction.*- Paderborn: Springer, 2000, 362 p., ISBN 3-540-67546-9
134. *Aladjev V.Z., Vaganov V.A., Shishakov M.L., Hunt U.J. A Workstation for Mathematicians.*- Gomel-Tallinn-Moscow: International Academy of Noosphere, 1999, 605 p.
135. *Ben-Israel A., Gilbert R. Computer-Supported Calculus*, in 2 volumes.- Berlin: Springer, 2000
136. *Oracle WebDB. Version 2.0 Evaluation.*- Redwood Shores: Oracle Corp., CD, 1999
137. *Computer Science. Newsletter*, no 1-3.- Berlin: Springer, 2000, textbooks@springer.de, http://www.springer.de/customers/textbook_inspect.html
138. *Bettini C. et al. Temporal Databases with Multiple Granularities.*- Milan: Springer, 2000, 200 p.
139. *Feuerlicht G. Object-Relational Database Management.*- Sydney: Springer, 2000, 350 p.
140. *Kuper G., Libkin L., Paredaens J. Constraint Databases.*- Berlin: Springer, 2000, 427 p.
141. *Dajtbegov D. Software for Processing of Statistical Data.*- Moscow: Finance and Statistics, 1984
142. *Rouanet H. et al. New Ways in Statistical Methodology.*- Paris, 1998, 276 p.
143. *McPherson G. Statistics in Scientific Investigation: Its Basis, Application and Interpretation.*- Berlin: Springer-Verlag, 1990, 666 p. ISBN 3-540-97137-8
144. *Mokhtari M. MatLab 5.2 & 5.3 and Simulink 2&3 for Engineers.*- Paris: Springer, 2000, 730 p.
145. *Redfern D., Campbell C. The MatLab 5 Handbook.*- Waterloo: Sprinder, 1998, 488 p.
146. *SAS Institute Publications.*- Cary: SAS Institute Inc., 1994
147. *SAS/STAT User`s Guide. Vol. 1-2.*- Cary: SAS Institute Inc., 1990
148. *STATISTICA.*- Tulsa: StatSoft Inc., 1994
149. *Härdle W., Klinke S., Müller M. XploRe - Academic Edition: The Interactive Statistical Computing Environment.*- Berlin: Springer-Verlag, 2000, 526 p. (CD with XploRe-Learning Guide)
150. *S-Plus: Academic Edition Version.*- Cambridge: Mathsoft Inc., Springer, 2000, 560 p.
151. *Berenson M. et al. Intermediate Statistical Methods and Applications: Computer Package Approach.*- New Jersey: Prentice-Hall, 1983
152. *Dixon W.J. et al. BMDP Biomedical Computer Programs.*- Berkeley, 1981
153. *Voelki K.E., Gerber S.B. Using SPSS for Windows.*- Berlin: Springer-Verlag, 1999, 228 p.
154. *Brockwell P.J., Davis R.A. ITSM for Windows: A User`s Guide to Time Series Modelling and Forecasting.*- Berlin: Springer-Verlag, 1994
155. *Polasek W. EDA - Explorative Datenanalyse.*- Berlin: Springer-Verlag, 1994
156. *Karian Z.A., Dudewicz E.J. Modern Statistical Systems and GPSS Simulation.*- N.Y.: Springer-Verlag, 1998, 560 p., ISBN 0-8493-3927-7.
157. *The Seventh International Conference Computational Finance 2000.*- London: London Business

School, June 2000; <http://www.lbs.ac.uk/cf2000>

158. **Tanner M.A.** *Tools for Statistical Inference.*- Berlin-Heidelberg: Springer-Verlag, 1994
159. **Kobayashi M.** *Mathematica: An Introduction to Statistics and Probability.*- Tokyo: Toppan, 1994, 232 p., ISBN 4-8101-8932-5 (includes a diskette)
160. *Teabevihik*. No. 4.- Tartu: Eesti Statistikeselts, 1994 (in Estonian)
161. **CRC Standard Mathematical Tables and Formulae** // Ed. **D. Zwillinger.**- Berlin: Springer, 1995, 832 p., ISBN 0-8493-2479-3
162. **Devore J.** *Probability and Statistics for Engineering and the Sciences.* 4th edition.- N.Y., 1995
163. **McCall R.** *Fundamental Statistics for the Behavioral Sciences.* 5th ed.- N.Y.: Harcourt Brace, 1990
164. **Malliavin P.** *Stochastic Analysis.*- Paris: Springer, 1997, 343 p., ISBN 3-540-57024-1
165. **Moeschlin O. et al.** *Experimental Stochastics.*- Berlin-London: Springer, 1998, CD, 206 p.
166. **Simonoff J.S.** *Smoothing Methods in Statistics.*- New York: Springer, 1998, 338 p.
167. **Whittle P.** *Probability via Expectation.*- Cambridge: Springer, 2000, 352 p.
168. **COMPSTAT 2000 - Proceedings in Computational Statistics** // Eds. **J. Bethlehem et al.**- The Netherlands: Springer, 2000, 540 p., ISBN 3-7908-1326-5
169. **Kil'dishev G.S.** *General Theory of Statistics.*- Moscow: Statistics Press, 1980
170. **McPherson G.** *Statistics in Scientific Investigation: Its Basis, Application and Interpretation.*- Berlin-Heidelberg: Springer-Verlag, 1990, 666 p., ISBN 3-540-97137-8
171. **Bertoin J. et al.** *Lectures on Probability Theory and Statistics.*- Paris: Springer-Verlag, 1999
172. **Mittelhammer R.** *Mathematical Statistics for Economics and Business.*- N.Y.: Springer-Verlag, 1999
173. **Simonoff J.S.** *Smoothing Methods in Statistics.*- New York: Springer, 1998, 338 p.
174. **Aladjev V.Z.** *Interactive Encyclopaedia of Cellular Automata.*- Tallinn: The International Academy of Noosphere, The Baltic Branch, 2006 (in preparation).
175. **Lakin G.F.** *Biometrics.*- Moscow: Higher School Press, 1990, 352 p.
176. **Kudrjavtsev V.A., Demidovich B.P.** *Brief Course of Higher Mathematics.*- M.: Nauka Press, 1989
177. **Fattahi A.** *Maple V Calculus Labs.*- London: Brooks/Cole, 1998, 276 p.
178. **MAPLETECH: Special Issue on Industrial Mathematics and Industrial Applications of Maple.**- Allentown: Air Products and Chemicals, Inc., 1998
179. **Clegg F.** *Simple Statistics: A Course Book for the Social Sciences.*- Cambridge: Camb. University Press, 1990, 200 p., ISBN 0-521-28802-9
180. **Coombes K. et al.** *Differential Equations with Maple.*- New York: John Wiley & Sons, 1996
181. **Ellis W. et al.** *Maple V Flight Manual: Tutorial for Calculus, Linear Algebra and Differential Equations.*- London: Brooks/Cole Publishing Co., 1996, 376 p.
182. **Barrow D. et al.** *Solving ODE with Maple.*- London: Brooks/Cole Pub Co., 1997
183. **Greene R.** *Classical Mechanics with Maple.*- New York - Heidelberg: Springer-Verlag, 1995
184. **Adams S., Tocii C.** *Maple Talk.*- New York: Prentice-Hall, 1996, 346 p.
185. **Chiang K., Mann H.** *Maple V for Physicists.*- Berlin: Springer-Verlag, 1996, 256 p.
186. **Cohen J., Hagin F.** *Calculus Explorations with Maple V.*- New York: Prentice-Hall, 1995
187. *Computational statistics* // Eds. **W. Härdle et al.**, 2002, ISSN 0943-4062
188. *Journal of Classification*, ISSN 0176-4268, ISSN 1432-1343 (electronic)
189. **Capinski M., Zastawniak T.** *Probability Through Problems.*- London: Springer, 2001, 300 p.
190. **Emery M. et al.** *Lectures on Probability Theory and Statistics.*- Paris: Springer, 2000, 349 p.
191. **Halloran M., Berry D.** *Statistical Models in Epidemiology, the Environment and Clinical Trials.*- Berlin: Springer, 2000, 274 p., ISBN 0-387-98224-2
192. **Jacod J., Protter P.** *Probability Essentials.*- Paris: Springer, 2000, 200 p., ISBN 0-387-98649-9
193. **Steele M.J.** *Stochastic Calculus and Financial Applications.*- Philadelphia: Springer, 2000, 335 p.

194. *Journal of Nonparametric Statistics*, ISSN 1048-5252
195. *Journal of Statistical Computation and Simulation*, ISSN 0094-9655
196. *Statistics: An Journal of Theoretical and Applied Statistics*, ISSN 0233-1888
197. *Stochastics and Stochastics Reports*, ISSN 1045-1129
198. **Pecquet L.** *A First Course in Magma - the Computer Algebra System.*- Paris-Berlin: Springer, 2001
199. *Computer Algebra in Scientific Computing - CASC-2000* / Eds. **V.G. Ganzha et al.**- Munchen: Springer, 2001, 439 p., ISBN 3-540-41040-6
200. **Gatermann K.** *Computer Algebra Methods for Equivariant Dynamical Systems.*- Berlin: Springer, 2000, 153 p., ISBN 3-540-67161-7
201. **Grabmeier J. et al.** *Handbook of Computer Algebra: Foundations, Application, Systems.*- Heidelberg: Springer, 2001, 400 p., ISBN 3-540-65466-6
202. **Langtangen H.P. et al.** *Advances in Software Tools for Scientific Computing.*- Oslo-Berlin: Springer, 2000, 360 p., ISBN 3-540-66557-9
203. **Scott B.** *Maple for Environmental Sciences.*- Melburn-Berlin: Springer, 2001, 360 p.
204. **Jain S.K. et al.** *Basic Linear Algebra with MatLab.*- Pittsburgh-Heidelberg: Springer, 2001, 305 p.
205. *MathCAD 2000 Professional: The Worldwide Standard for Technical Design: Student Version.*- Cambridge: Mathsoft Inc., 1999, 337 p. + CD, ISBN 3-540-14859-0
206. *Encyclopedia of Biostatistics* // Eds. **P. Armitage, T. Colton**, in 6 vol.- Oxford-Boston-New-York: Wiley, 1998, 4898 p., ISBN 0471-97576-1
207. **Aladjev V.Z.** *Computer algebra* // *Alpha*, no. 1 (April), 2001
208. **Aladjev V.Z., Bogdevicius M.A.** *Maple 6: Solution of Mathematical, Statistical and Engineering-Physical Problems.*- Tallinn-Vilnius: Vilnius Technical University, 2001, 942 p. + CD
209. *Informatiology problems of mankind in XXI century.*- Moscow: Informatiology Press, 2000
210. **Aladjev V.Z.** *Modern computer algebra for simulation of transport systems* // Proc. Int. Conf. **TRANSBALTICA-2001.**- Vilnius: Technika Press, 2001
211. **Abell M.** *Differential Equations with Maple V.*- London: Academic Press, 1999
212. **Articolo G.** *Partial Differential Equations and Boundary Value Problems with Maple V.*- N.Y.: AP Professional, 1998, ISBN 012-064475-4
213. **Davenport J et al.** *Computer Algebra.*- N.Y.: Academic Press, 1993
214. **Hawley W.** *Foundations of Statistics.*- N.Y.: Saunders College Publishing, 1996
215. **Kirk R.** *Statistics: An Introduction.*- N.Y.: Saunders College Publishing, 1999
216. **Ross. S.** *Introduction to Probability and Statistics for Engineers and Scientists.*- N.Y.: Academic Press, 1999, ISBN 012-598472-3
217. **Roussas G.** *A Course in Mathematical Statistics.*- N.Y.: Academic Press, 1997
218. **Freund R., Wilson W.** *Regression Analysis.*- N.Y.: Academic Press, 1998
219. **Hays W.** *Statistics.*- N.Y.: Saunders College Publishing, 1993
220. **Newmark J.** *Statistics and Probability in Modern Life.*- N.Y.: Saunders College Publishing, 1998
221. **Betounes D.** *Differential Equations: Theory and Applications with Maple.*- N.Y.: Springer, 2001
222. **Cornill J.-M., Testud P.** *An Introduction to Maple V.*- Paris: Springer, 2001, 470 p.
223. **Sampath S.** *Sampling Theory and Methods.*- Alpha Science International Ltd., 2001, 184 p.
224. **Stahel W.A.** *Statistische Datenanalyse.*- Zurich: ETH Zentrum, 2000, 375 p., ISBN 3-528-26653-8
225. **Bunday B.** *Basic optimization methods.* School of mathematical Sciences, Univ. of Brodfard, 1984
226. **Murauskas G., Shioshys V.** *Nonlinear programming.* Vilnius university, Vilnius, 1991, 131 p.
227. **Luenberger D.G.** *Linear and nonlinear programming.* 2nd. Addison-Wesly, Massachusetts, 1984
228. **Polak E.** *Computational Methods in Optimization: A Unified Approach*, Academic Press, N.Y., 1971
229. **Walsh G.R.** *Methods of Optimization.* John Wiley & Sons, 1975

230. **Goldberg D.E.** *Genetic Algorithms in Search, Optimization, and Machine Learning.* Addison-Wisley Publishing Company, INC. Reading, 1989, 412 p.
231. **Michalewicz Z.** *Genetics Algorithms+Data Structures=Evolution Programs.* Warsaw, 1996, 432 p.
232. **Redfern M., Betounes D.** *Mathematical Computing: An introduction to Programming Using Maple.*- Berlin: Springer, 2002, 400 p. + CD, ISBN 0-387-95331-0
233. **Grabmeier J. et al.** *Handbook of Computer-Algebra: Foundations, Applications, Systems.*- Berlin: Springer, 2002, 400 p. + CD, ISBN 3-540-65466-6
234. **Aladjev V.Z., Bogdevicius M.A.** *Special Questions of Operation in Environment of the Mathematical Maple Package.*- Tallinn-Vilnius: *International Academy of Noosphere & Vilnius Gediminas Technical University*, 2001, 215 p. + diskette, ISBN 9985-9277-2-9
235. **Aladjev V.Z., Vaganov V.A., Grishin E.P.** *Additional functional facilities of the Mathematical Maple Package of releases 6 and 7.*- Tallinn: *International Academy of Noosphere*, 2002, 312 p. + diskette
236. **Aladjev V.Z.** *Effective Operation in the Maple Package.*- Moscow: *Laboratory of Base Knowledge*, 2002, 350 p. + CD (in Russian with extended English summary).
237. **Aladjev V.** *Computer Algebra System Maple: A New Software Library / ICCS*, Melbourne, 2003
238. **Aladjev V.** *Systems of Computer Algebra: New Software ToolBox for Maple / Int. Conf. ACAT03*, Tokyo, 2003
239. **Aladjev V.** *Computer Algebra Systems: A New Software Toolbox for Maple.*- Palo Alto: Fultus Publishing, 2004, ISBN 1-59682-000-4
240. **Aladjev V.** *Computer Algebra Systems: A New Software Toolbox for Maple.*- Palo Alto: Fultus Publishing, 2004, ISBN 1-59682-015-2, Adobe Acrobat eBook (pdf)
241. **Aladjev V. et al.** *Electronic Library of Books and Software for Scientists, Experts, Teachers and Students in Natural and Social Sciences.*- Palo Alto: Fultus Publishing, 2005, CD
242. **Aladjev V.Z., Vaganov V.A.** *Systems of Computer Algebra: A New Software Toolbox for Maple.*- Tallinn: *International Academy of Noosphere*, 2003, 270 p., ISBN 9985-9277-6-1
243. **Aladjev V.Z., Bogdevicius M.A., Vaganov V.A.** *Systems of Computer Algebra: A New Software Toolbox for Maple. Second edition.*- Tallinn: *International Academy of Noosphere*, 2004, 462 p.
244. **Aladjev V.Z., Bogdevicius M.A.** *Computer algebra system Maple: A new software toolbox // 4th Int. Conf. TRANSBALTICA-03*, Technics Press, April 2003, Vilnius, pp. 458-466.
245. **Aladjev V.Z.** *Computer Algebra System Maple: A New Software Library // Int. Conf. "Computer Algebra Systems and Their Applications", CASA-2003*, Saint-Petersburg, Russia, 2003.
246. **Aladjev V., Bogdevicius M., Vaganov V.** *Systems of Computer Algebra: A New Software Toolbox for Maple // Intern. Conf. on Software Engin. Res. and Practice, SERP'04*, 2004, Las Vegas
247. **Aladjev V.** *Computer Algebra System Maple: A New Software Library / ICCS*, Melbourne, 2003
248. **Aladjev V.Z., Haritonov V.N.** *General Theory of Statistics.*- Palo Alto: Fultus Corporation, 2004
249. **Maple 8 Learning Guide.**- Toronto: Waterloo Maple Inc., 2002, 308 p.
250. **Maple 8 Introductory Programming Guide.**- Toronto: Waterloo Maple Inc., 2002, 380 p.
251. **Maple 8 Advanced Programming Guide.**- Toronto: Waterloo Maple Inc., 2002, 382 p.
252. **DeMarco P. et al.** *Maple Advanced Programming Guide.*- Waterloo Maple Inc., 2005
253. **DeMarco P. et al.** *Maple Introductory Programming Guide.*- Waterloo Maple Inc., 2005
254. **Abell M., Braselton J.** *Maple by Example*, 3rd Edition.- Berlin: Springer, 2005
255. **Corless R.** *Symbolic Recipes: Scientific Computing with Maple.*- Waterloo Maple Inc., 2005
256. **Adams P. et al.** *Introduction To Mathematics With Maple.*- Waterloo Maple Inc., 2004
257. **Maple 9.5 Getting Started Guide.**- Waterloo Maple Inc., 2004
258. **Enns R., McGuire G.** *Computer Algebra Recipes + CD with Maple Softcover.*- Berlin: Springer, 2006
259. <http://writers.fultus.com/aladjev/source/UserLib6789.zip>
260. <http://writers.fultus.com/aladjev/source/Archive.zip>

The information about the writers

Aladjev Victor

Aladjev V.Z. was born on June 14, 1942 in the town of *Grodno (Western Byelorussia)*. After successful completion of the secondary school (*Grodno*) in 1959 he matriculated on the first course of physical and mathematical faculty of *Grodno State University*, and in 1962 was transferred to the Department of Mathematics of *Tartu State University (Estonia)*. In 1966 he successfully finished *Tartu State University* the speciality of *Mathematician*. In 1969 he started post-graduate studies of the Estonian Academy of Science the speciality of *Probability Theory and Mathematical Statistics*, and successfully finished in 1972 two specialities "*Theoretical Cybernetics*" and "*Technical Cybernetics*". The doctoral degree of the mathematician was assigned to him for the monography "*Mathematical Theory of Homogeneous Structures and their Applications*".

From 1972 to 1990 he had a respectable positions in a number of project-technological and research organizations of *Tallinn (Estonia)*. In May 1991 he established scientific company *VASCO Ltd.* and was a director; in December 1992 he became vice-president of the joint venture *Salcombe Eesti Ltd.* In 1999 he was appointed the President of the *Tallinn Research Group (TRG)*, the scientific findings of which received international recognition, first of all, in the field of activities in mathematical theory of *Homogeneous Structures (Cellular Automata)*.

Aladjev V.Z. is the author of more than 350 scientific and technological publications (*including 65 monographies, books and collections of the articles*), published in the former *USSR, Estonia, Russia, Byelorussia, the Ukraine, Lithuania, GDR, Germany, Czechoslovakia, Hungary, Japan, USA* and *Holland*. Since 1972 he is the reviewer and associate editor of the international mathematical journal "*Zentralblatt fur Mathematik*" and since 1980 a member of *IAMM (The International Association for Mathematical Modelling, USA)*. He founded the *Estonian* school of mathematical theory of *Homogeneous Structures (Cellular Automata)* which received international recognition and covered the fundamentals of the new section of modern mathematical cybernetics.

Since October 1993 **Aladjev V.Z.** has been elected as member of Working Group *IFIP (International Federation for Information Processing, USA)* studying the mathematical theory of homogeneous structures and its applications. **Aladjev V.Z.** participates as a member of the organizing committee and/or a guest lecturer in many international scientific forums in mathematics and cybernetics. In April 1994 **Aladjev V.Z.** was elected as an *academician* of the *Russian Academy of Cosmonautics* in the section of *Fundamental Researches*; in September 1994 he was elected as an *academician* of the *Russian Academy of Noosphere* in the section of *Information Science*. In September 1995 **Aladjev V.Z.** was elected as an *academician* of the *Russian Academy of Natural Sciences* in the section of *Noosphere Knowledge and Technologies* and in June 1998 as an *honorary academician* of the *Russian Ecological Academy*.

In November 1997 **Aladjev V.Z.** was elected as the *academician-secretary* of the *Baltic Branch* of the *Russian Academy of Noosphere*, integrating scientists and specialists of three *Baltic countries* and *Byelorussia*, who work in the field of a complex of scientific disciplines, which are included in the doctrine about Noosphere (*sphere of mind*) and areas, adjacent to it, including theoretical and applied questions of the homogeneous structures problems. As a result of reorganization of the

Russian Academy of Noosphere **Aladjev V.Z.** was elected as the *First vice-president* in December 1998. In December 1999 **Aladjev V.Z.** was elected as a *foreign member* of the *Russian Academy of Natural Sciences* (RANS) in the section of *Information Science and Cybernetics*.

Most considerable scientific results of **Aladjev V.Z.** are related to the mathematical theory of homogeneous structures (*Cellular Automata*) and its applications. The sphere of his scientific interests includes mathematics, information science, cybernetics, computing sciences, mathematical and theoretical biologies, physics, astronautics, cosmology, problems of universe and a number of other naturally-scientific areas of modern knowledge.

Bogdevicius Marijonas

Bogdevicius M.A. was born on January 2, 1958 in the city of *Vilnius (Lithuania)*. After successful completion of secondary school (*Vilnius*) in 1976 he entered the Mechanical Technological Faculty of *Vilnius Civil Engineering Institute* (now *Vilnius Gediminas Technical University*), which he finished with honours in 1981 in the speciality of "*Building and road machines and equipment*". During his studies **M. Bogdevicius** actively participated in research work in the field of computational mathematics and received the diploma of the first degree in the Republican conference of student research in 1980.

M. Bogdevicius started working for the *Vilnius Branch of Experimental Scientific Institute of Mechanical Engineering*. Then he started working for the faculty of Mechanical Technological of *Vilnius Civil Engineering Institute*. From 1983 to 1988 he was an assistant of the chair, 1988-1991 - a senior teacher, 1991 - 1995 - a associate professor. Since 1995 **M. Bogdevicius** is the Head of the Department of Transport Technological Equipment of Transport Engineering faculty of *Vilnius Gediminas Technical University*. In 1988 Bogdevicius defended a thesis on *Building and road machines* at *Moscow Automobile and Road Institute*. In 1990-1991 he worked on probation in *Stuttgart University (Germany)* in the Institute of Hydraulic Machines.

In 1993 in the order of nostrification the scientific degree of the doctor of engineering science was awarded to him. **M. Bogdevicius** has published more than 180 scientific and technical works; he is the author of over 25 inventions. A lot of his research is on dynamic processes in mechanical, pneumatic and hydraulic systems.

Being the Head of the Department of *Technological Transport Equipment* of *Vilnius Gediminas Technical University*, **Bogdevicius M.A.** reads the lectures on basic courses: "*Finite element method in the mechanics of continuous medium*", "*Simulation of technological transport equipment*", "*Optimization of transport machines*", "*Vehicles' dynamics*" etc. He defended thesis on "*Simulation of dynamic processes in hydraulic, pneumatic and mechanical drives and their elements*" and was conferred a degree of the habilitated doctor in engineering science in *Vilnius Gediminas Technical University* in August 2000.

In September 1998 **M. Bogdevicius** was elected as a full member of the *Tallinn Research Group (Estonia)* in the section of informatics. In December 1998 **M. Bogdevicius** was elected as a full member of the *Russian Academy of Noosphere* in the section of informatics and informational technologies, and in December 1999 - as a full member of the *International Academy of Noosphere* in the section of informational technologies. In April 2004 **M. Bogdevicius** was elected as a full member of the International Informatization Academy .

M. Bogdevicius's interests include such topics as: computation methods, methods of optimization, dynamics of mechanical, hydraulic and pneumatic systems, problem of safety of road motion as well as software for the solution of different technical and manufacturing problems. **M. Bogdevicius** pays a lot of attention to pedagogical activities as the Head of the Transport Technological Equipment at *Vilnius Gediminas Technical University*.