Chapter 12

Solving optimal control problems described by PDEs

by Czesław Bajer[1], Andrzej Myśliński[2], Antoni Żochowski[2], Bartłomiej Dyniewicz[1] and Dominik Pisarski[1]

12.1 Introduction

The computational methods used in practice do not allow an accurate representation of the phenomena occurring in the processes described. Despite the effort and engagement of the knowledge from many disciplines we can not describe all the phenomena which occur. For this reason, we try to separate them from each other and focus our attention on one single phenomenon. Often we are forced to go back and accept the solution of simpler problems, which significantly deviates from our initial expectations. We make simplifications every step of the calculation.

We must chose between statics and dynamics, linear or nonlinear description, and finally take a solution method which satisfy our requirements. Solution methods applied to problems described by partial differential equations are treated in numerous papers. The efficiency and accuracy of solutions are main two features that are taken into account in numerical modelling. Numerical methods applied to such problems can be divided into two main groups:

- methods based on the discretization of the differential equation (for example central difference method),
- methods based on the discretization of the spatial and time domain of the problem (finite element method, space-time finite element method).

Moreover, one method can be applied to space while the other to time.

While many of simplifications are intuitive and the degree of approximation is assessed in a rather arbitrary manner, the degree of approximate validity of these mathematical methods can usually be estimated well. Hence, there are a large variety of computational tools. Mathematical methods that lead to numerical schemes can be divided into three groups:

Strong form – description of the equations of motion is represented by a system of differential equations in space and time, supplemented by boundary and initial

[1] Institute of Fundamental Research in Technology, Polish Academy of Sciences, 02-106 Warsaw
[2] Systems Research Institute, Polish Academy of Sciences, 01-447 Warsaw

conditions. Some computational methods reduce a strong system to a system of algebraic equations.
Weak form – presents a problem in the form of a weighted integral equation. This leads to a description of averaging over the field considered.
Variational form – presents a problem in the form of functional whose stationarity conditions lead to a weak or strong form. The transition from one form to another is done using the rules of the variational calculus.

Strong forms of description of phenomena have been used for a long time. Newton's second law is an example. Solutions of the tasks described by strong forms consist of a direct discretization of the differential equation, such as the finite difference method. Suitable differences replace the differential quotients. The basic defect is the difficulty of its application for non-rectangular or non-circular areas, and for problems which take into account the boundary conditions.

Therefore, formulations based on weak forms and variational forms have gained greater popularity. The advantages of such a course of action are as follows:

- unification of procedures in various theoretical and engineering problems; functionals are scalars, and as such do not depend on the reference system, which in turn facilitates appropriate transformations,
- weak forms and variational forms are the basis for effective computer methods,
- with variational forms and weak forms the basic principles of mechanics, such as energy conservation, conservation of mass, conservation of momentum and angular momentum can easily be expressed,
- the error estimation is facilitated, and so is the determination of the stability and the convergence of the numerical method employed.

Variational methods of solutions dominate methods based on strong forms. The latter are slowly beginning to address the conservation problems involved with historic sites.

12.1.1 General information on classes of problems

Partial differential equations which describe problems of mathematical physics can be divided into three main groups:

- elliptic differential equations that describe stationary problems, for example

$$\frac{\partial^2 u(x,y)}{\partial x^2} + \frac{\partial^2 u(x,y)}{\partial y^2} = f(x,y) \tag{12.1}$$

- parabolic differential equations that contain first time derivative and describe problems of diffusion, heat transmission, etc., for example

$$\frac{\partial^2 u(x)}{\partial x^2} - \frac{1}{a^2}\frac{\partial u(x)}{\partial t} = f(x,t) \tag{12.2}$$

- hiperbolic differential equations that contain second time derivative and describe dynamic problems, for example

$$\frac{\partial^2 u(x)}{\partial x^2} - \frac{1}{c^2}\frac{\partial u^2(x)}{\partial t^2} = f(x,t) \tag{12.3}$$

All three groups are important from the viewpoint of engineering practice and research. Treatment of physical phenomena in areas of complex shape is a characteristic feature. In such areas known analytical methods leading to closed solutions fail. In the group of approximate methods the finite element method gained a dominant position, because of simplicity in engineering practice and a more intuitive discretization of the domain.

A shortcoming of the finite difference method is an effort of the imposition of constraints (boundary conditions). More difficult is also the interpretation of results, and above all the determination of the parameters derived from state parameters. In addition, basic variant of the finite difference method required a certain regularity of the test domain. Ideally, when it consists of rectangular subdomains. The generalization of the method for arbitrary areas developed by Orkisz [192, 261], gave a powerful computational tool. Unfortunately, computational methods based on discretization of the differential equation lost with the finite element method, easier in algorithmization.

Computational capabilities of the methods, used in practice and allowing for effective modification of the task, thus optimizing the solution or control of selected parameters, result from the description method of the output. Due to the long experience of the authors related to programming and numerical solving of tasks, at this stage we reject all the techniques associated with the discretization of differential equations, and leave the finite element method as a tool to enable a relatively easy modification of the mathematical model.

The important stage is the input of data. In the finite element method, regardless of the type of problem, five groups of information are required. They provide fundamental information on the object, boundary conditions, and externally applied load:

1. coordinates of nodal points in the mesh,
2. mesh topology, i.e. numbers of nodes in elements,
3. material data,
4. number of bodes with imposed boundary conditions and type of constraints,
5. external load (for example external forces, heat flux, etc.).

Computer packages for the numerical simulations are complex because phenomena analyzed are complex. Creation of a computer program intended for a single, definite task takes 1–5 person-years (it is not applicable to the scope of simple tasks performed in a limited range as part of work of graduate students). Relatively universal commercial packages consume the equivalent of tens to hundreds of person-years of work. And so they lack capacity to analyze many important problems, such as for example elasticity in moving the external load [107, 19, 20].

Due to the nature of the input data, we can separate the tasks in which we can modify only selected structural parameters. In this group of optimization problems we adjust strength parameters (mainly cross-sections) to minimize the cost of construction, while ensuring the desired strength. School example is the selection of the truss height and the type of its geometrical scheme, to cover the given gap. Mostly we do this for problems described by elliptic differential equations.

Another task of border of control and optimization is described by the parabolic equation the heating process in designated area to a certain temperature. In the process of hardening, melting, casting, injection molding, etc. we want to achieve the goal of spending the minimum of external energy. Moreover, the heating or cooling process is also clearly defined in advance. We cool the object evenly and avoid damaging residual internal stress. Important is the performance of the process in time. These tasks are relatively well recognized experimentally by trial and error way. It does not change the fact that more and more complex objects, heat treated, require new trials.

The most difficult are the tasks described in hyperbolic equations. These are the tasks of structural dynamics, vibration problems, wave problems. Here, on the one hand side we describe the construction with the system of elliptic-parabolic differential equations, on the other side the term associated with time brings the hyperbolic nature. These are, like the previous group's tasks, the problems of evolutionary nature. It is impossible to predict the effect of the process, without observation of the entire path of transition to the final stage. Tasks of this group are computationally very expensive with high risk associated with the instability of the solution.

Here we touch an essential feature of all three groups of tasks: nonlinearity of equations describing the problem. In the simplest case this can be taken into account the friction phenomenon or the contact of a portion of the edge to another part of the structure or outer constraints. Energy functionals, by which we describe the tasks are usually non-convex and we can not prove the existence and uniqueness of solutions. For this reason, the process is discretized in time and we assume that in time intervals variable parameters are constant. In this way we get piecewise linear equations, whose features and properties we know. Unfortunately, we can not give a global solution. For this reason the impact on the solution by changing the parameters is extremely difficult. It is easy to exceed the limitations imposed by stability criteria. Moreover, most of linearized tasks criteria are difficult to identify.

12.1.1.0.1 Available software

Let us compare here control problems described by ordinary differential equations with control problems applied to partial differential equations. The fundamental feature that differs both groups is the computational complexity and cost of a single solution of the engineering problem, in practice – the objective function.

In the case of ODE the control problem takes about half of the total computational cost. The control problem in PDE takes few percent or less of the total cost. The remaining incurred effort, i.e. the majority, is spent for performing engineering computations.

The following facts must be underlined:

1. the solver of PDE systems for linear elliptic problems is relatively simple and can be programmed by skillful users in reasonably time,
2. the graphical input and output for complex domains is time consuming and depends on the computer architecture; it is usually considered as a separated problem,
3. the mesh generation is also complex even in the case of 2-dimensional domains, although numerous efficient procedures exist; in the case of 3-dimensional objects it can be a real challenge; in this case the generation of of a mesh must be supplemented with efficient way of assignment of different material parameters to each finite element.

Then the calculation part can be elaborated individually or the available open-source package can be applied. Individual software gives the opportunity of interference when the control procedure is elaborated. The possibility of using public domain packages are extensive. There are probably over a hundred items. Some of them are described in terms of the purpose and basic features. Most proposed are as follows:

Faep – complex finite element analysis with nonlinear problems,
Felt – in Fortran, applied to frames, 2D and 3D solids, statics [150], thermal problems and dynamics,
FElib v. 3 and v. 4, written in Fortran,
FEC – library in C,
FELyX – library in C,
FEM2DLib – library in Fortran,
FEMLab2d – interactive FE package to 2D problems,
FEMLib – object oriented library in C++,
FEMOctave – library package to system Octave,
FEMSET – package to bar and frame analysis,
Free Finite Element Package – library in C,
FreeFEM, FreeFEM+, FreeFEM++, FreeFEM3D – library for adaptive Finite Element Method,
GeoCrack – fluid flow in porous media,
KFEM – adaptation of FreeFEM to graphical Kde desktop environment),
MiniFEM – package for 2D problems,
ModFe – water flow in soil,
SEM2DPACK – in Fortran, for analysis of seismic wave propagation.

The vast majority of the packages were made as students and doctoral students works. They are usually organized as a collection of procedures or semi-compiled

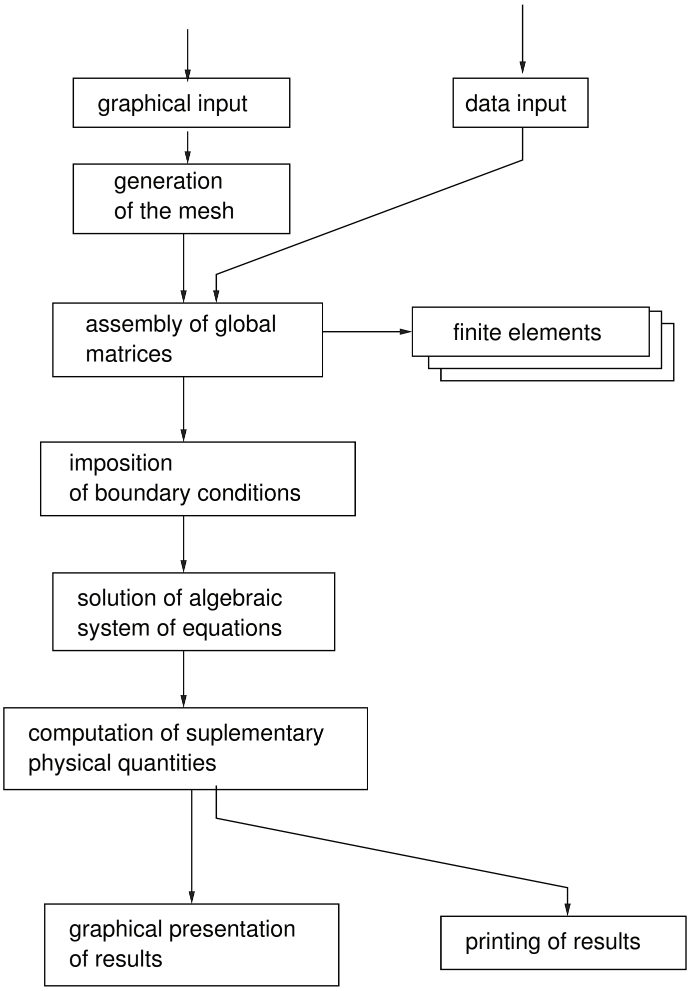

Fig. 12.1 Scheme of the computational process performed by the finite element method.

libraries. In the hands of their creators they can be relatively efficient. Unfortunately, they have the following disadvantages, practically excluding them from wider use:

- they are written for individual purpose, arising in an evolutionary way from the author's acquired skills and knowledge of the subject, and often so are not useful in a wider use,
- they do not have sufficient documentation and therefore the possible use must be preceded by a time-consuming procedures for the diagnosis of contents; this cost is comparable to writing our own procedures from the beginning,

- procedures contain errors, of which the author are not aware, or deliberately left them abandoning the testing stage and corrections; the use of them forces others to adjust to the awkward style, programming (the division of tasks procedures, characteristic parameters of objects, etc.) often can not be compatible, especially if we use modules from different sources.

For this reason it is extremely difficult to adapt for own needs publicly available programs. Their use is usually limited to the repetition of tasks similar to the test examples, delivered together with source procedures. On the other side commercial codes are well documented and allow solve various problems. The variety of problems that can be treated by commercial software usually exhausts assumed aim. There is a reason why commercial software packages are expensive, and their implementation – laborious. Experience shows that the time devoted to write raw code represents 5–10% of the total time spent on creating a finished, efficient, and well documented computer program. Therefore, the average statistical efficiency of programmers is 5 standard code instructions par day. It is not surprising that the creators of publicly available sources of codes do not take the effort to pass tests and documentation stage.

12.1.1.0.2 Bar structure vs. plane structures and solids

Control problems applied to bar structures (ie. trusses, grids, frames) can be divided into two groups:

- optimization of the geometry (lengths of elements, cross-sectional areas of bars),
- control of vibrations of structures:
  - characteristic modes of vibrations,
  - vibrations under oscillatory load applied to selected points,
  - vibrations under a moving non-inertial and inertial load,
  - coupling of vibrations between two structures.

2- and 3-dimensional problems additionally enable the control of heat conduction, diffusion and filtration in porous media. Nowadays bar structures do not cause difficulties.

12.1.1.0.3 Use of the available software

We can not count on making full use of public domain software. Worth consideration are therefore the following solutions:

1. use of the existing two-dimensional mesh generator and creation of the own solver for tasks of two-dimensional unsteady heat flow and elasticity,
2. use of the existing two-dimensional mesh generator program and available finite element solver for respective tasks,
3. stay in the research area of bars and preparing data in a form.

Because of the given objective of the project we take into account only the packages that allow the calculation in batch mode. We exclude programs that run graphically and issuing the results graphically. In the case of graphical service is not easy to run the code on the remote machine in browser mode. In the case of 2-D problems we propose the procedure as in Figure 12.2. In discretization it is sufficient to use

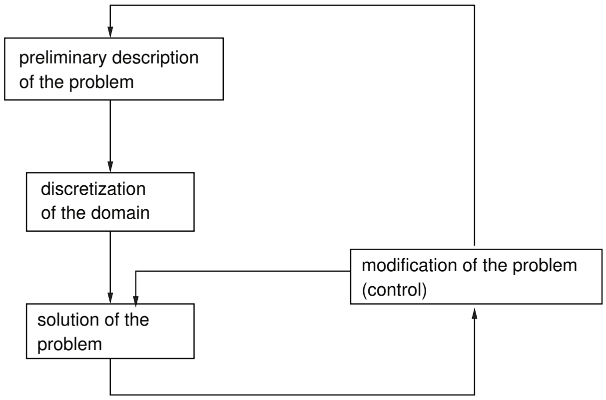

Fig. 12.2 Scheme of the solution of 2D problems.

the program triangle (Figure 1.4). It works by adapting created triangle mesh to given contour lines. In different subdomains we can get different mesh densities. The degree of elongation of triangles is also restricted. Numerous set of parameters allows to obtain the desired result.

Inconsistent numbering of nodes is a drawback of the creation performed with triangle. Some numbers have no equivalents in the resulting table of nodal numbers and their coordinates. Before using the resulting array the renumbering of nodes numbers must be performed with supplementary procedure. The program is useful in two dimensional problems since the mesh generation is time consuming and elaboration of own mesh generators for arbitrary contours is complex.

12.1.2 Active control of mechanical structures described by partial differential equations

Lightweight structures have been intensively investigated in recent years. Optimal design with low weight is insufficient in the case of dynamic behaviour of the structure. There are some approaches to the decrease of the vibration level.

- Passive vibration absorbers — the idea is performed by means of dynamic vibration absorbers as a set of additional masses flexibly attached to the main system.

Both transverse and rotational types of absorbers can be applied. The out of phase vibration of dynamic absorbers results in elimination of vibration of the structure.

- Active control — the control forces are generated by electric actuators imposed on the tendons or on a stiff cantilever fixed to the end of the beam [135, 293]. The actuators can generate both transverse control force and bending moment. It enables to control the predominant lowest vibration modes of the beam whereas the piezo-electric actuators are used to control higher ones. The possibility of unstable behaviour in the case of improper design is the disadvantage of such a solution.
- Semi-active damping — the control of the damping of viscous dampers installed in the structure [259]. The span of the beam is supported by viscous dampers. They can be attached to rigid foundation or hanged on a system of tendons. The damping properties are changed according to the position of the travelling load or other, more complex observation of the beam response. Such a control is always stable.

The following are the possible applications of optimal controls for the selected dynamic systems. We will present the concepts of the use of methods of optimal controls of discrete systems to some of the proposed tasks. Each of them is the unique solutions and certainly the attempt of implementation will be attractive. The tasks are grouped and dealt with separately, because each of them requires a different approach in the formulation of the problem and the optimization.

We can give the following examples of problems:

1. The class of active controls of continuous systems subjected to travelling load can contain particular tasks:

   a. Optimal trajectories of passing load (straightness).
   b. Minimization of deflection of structures under a moving load.
   c. Minimization of deflections of a structure to preserve minimum of the energy expense.
   d. Optimal control of the vehicle suspension.

2. The class of active controls of continuous systems under a given load, subjected at a set of stationary points contains the following tasks:

   a. Optimal reduction of structural vibrations.
   b. Optimal transition of the system from one dynamical state to another.

3. The class of active controls of continuous systems subjected to small disturbance can contain:

   a. Stabilization of the system with possible use of optimal Lyapunov functions.

4. Transport problem:

   a. Reduction of vibrations of guideways, bridges, and railway tracks.
   b. Multiple-beam systems that absorbe vibrations [271].
   c. Strings and cables supporting moving loads.

12.1.2.0.4 Examples of problems

Ad. 1a. The perspective of applications to various production processes where precise movement in a straight line is much essential, the implementation of the rectilinear trajectory of passage can be attractive. Active and semi-active control of beams or plates elements can significantly correct the quality of such trajectories.

State of the system would be defined by the coordinates of displacements from the equilibrium position. The moving load can be accepted as a force of constant value travelling along a beam or plate with constant velocity. Any spatial distribution of the actuators would be an external control system. Cost function can be expressed by the integral of the deflections module calculated by the trajectory of the motion. Can also be minimized maximum deflections of the trajectory curve. Such a problem would fall to the category of controls in an open system, and control quality index would be calculated at time corresponding to the length of travel time. The control functions operate on symmetric compact set. As a result of proper discretization the problem can be formulated according to the assumptions for Maximum Principle. For the numerical solutions used to algorithms for nonlinear optimal control with appropriate integration of the system with coupled variable. The result would be the optimal control courses of the actuators applied to the system. Appropriate definition of penalty function which fulfil the assumption of a limited set of values of control functions.

The method of discretization significantly affects the computational aspects of the problem. If you take the modal coordinate space as the state space it should be adequately define the system load vector, which is a nonlinear operator of the additional state variable. It is de facto the time coordinate. This is a classic trick that is used in the transition from the non-autonomous system to the autonomous system. The linear non-autonomous problem transform to the non-linear autonomous problem, it can be directly applied the Pontriagin Maximum Principle. When integral indicator of quality specified only the state of a system, as previously mentioned, the form of a Hamiltonian system can be expected that the optimal control will take a bang-bang form. Suitable numerical methods can be used here to find the optimal switching times.

Ad. 1b. From the viewpoint of reduction strength of the structure deflections at specific points in the structure may be attractive. The development of transport, including high speed trains entails the construction of new, more durable structures or strengthen existing ones. Increasing requirements for speed and weight of vehicles require renovation of bridges, overpasses, railway and subway tracks. Actuators apply to the structure can significantly reduce the amplitude of the vibrations caused by moving loads. Indicator of quality control can be expressed a maximum deflection or have the integral form of the displacements calculated at selected points of the structure. Another approach suggested by the standards of construction vibrations would require to determine the quality controls by the integral of the velocity in the most sensitive points of the structure. Minimization of quality functionals so defined, would allow for finding controls, which much improve the life of the structure. Computational technique can be based on the formulation of the Pontriagin Maximum Principle.

Ad. 1c. This is a standard example of the minimum-square problem. Quality functional includes the energy spent on control. This energy is expressed by the corresponding quadratic form. In this optimal control problem formulation we obtain control functions dependent on the state variables - the closed-loop control. Solution of the Riccati equation would provide the terms of the control in a closed form.

Ad. 1d. The problem would be to find a control function responsible for the functioning of the actuator in a vehicle suspension travelling along a construction. This is a problem so-called moving oscillator. Its practical significance need for a comfortable passage through structures such as viaducts, bridges or plate is explained. The system can be expressed in the form of discrete modal space. The dynamics is described by a linear system of equations of motion for time-dependent coefficients. Assuming the quadratic performance index control can be expressed as a subsidiary of state variables. The disadvantage of this approach is that the matrix K in Riccati equation is time dependent and must be calculated separately for each position of the oscillator.

Ad. 2a. In many cases, the work of machines associated with excitation of vibrations in the foundation. Wherever there can not be applied directly to the effective isolation of contact with the ground and machine, you can consider the problem of active vibration damping using actuators to support the whole system, ie the foundation with the machine. Therefore can be considered plate subjected to an excitations at fixed points. Excitations may have set of known progress. The main class here would be the harmonic excitations. The plate may be simply supported on a fixed foundation and/or based on on a set of controllable actuators. Optimal control of such a system can be understood as an optimal reduction of vibration amplitudes at selected points of the plate. Appropriate for this task would be discrete model developed using finite element method. Operation of excitations and controls would be described by the vectors with nonzero components in locations corresponding position in space. Indicator of quality control would have the integral form of displacement or velocity of selected components of the state. On the occasion could be included the components determining the energy expenditure in control. After switching to an autonomous system we obtain nonlinear optimal problem formulated according to the direct application of Maximum Principle.

Ad. 2b. Here can be considered to carry out a continuous system from one state to another in minimum time. Discrete model obtained by finite element method would be a state variables system subjected to controlled external forces. Solution of the time-optimal problem can get by using the Maximum Principle. Consider appropriate limitations should be here at the state and control. Another problem would be to carry the system from one state to another within a specified time, but with a minimum expenditure of energy. Should use the appropriate form performance index such as the quadratic form with controls. Direct integration of differential equations of state and coupled variables would result in an optimal courses for controlled forces.

Ad. 3a. The problem could concern the stability of the continuous system. Many structures in the course of their work is subjected to external excitations of unknown disturbances. The use of active controls can increase the area of stability or improve the speed of convergence to equilibrium. The problem would be to find a suitable Lyapunov function, assuming a priori the form of such a quadratic form. Rate of convergence, expressed by the appropriate norm would be functional, which the extremum seeking. Controls may bang-bang form and dependent of Lyapunov function. The result is a control program in an open system. The necessary for calculations would be appropriate numerical procedure determines the optimal Lyapunov function for each point in time. The problem can also develop on the case when the disturbances are expressed by certain functions of time.

Ad. 4a, 4b. Semi-active control, of the issues so far, are the group most up to date, and the methods of solving them are not yet widely known. The share of these problems in the project would certainly be an important novel element, trying to implement them would be an interesting experiment.

The forces generated by the active dampers give the bilinear term in differential equations that is linear due to the state and control function. Suitable numerical algorithms for the optimization of such controls may be used. When the cost is determined by the integral state, a Hamilitonian form usually conclude that the character of optimal controls are bang-bang type. In the case of moving loads are sufficiently effective control with only one switching, the problem can be reduced quite easily so the non-linear programming. Consider two approaches to the continuous system discretization. FEM model provides a direct possibility to construct the observer and the closed-loop control. In this case for the purpose of use of existing algorithms for integration of coupled equations can be also formulate the problem for the Maximum Principle. Discrete model in modal space can be reduced to an autonomous nonlinear system.

Ad. 4c. The problem of the moving oscillator can also be seen in the category of semi-active control. The essence of this strategy is the control parameters of the vehicle suspension, in order to passage realized comfort trajectory, ie the smallest accelerations in the vertical direction. In such cases, heuristic methods sky-hook, ground-hook are usually adapted. The challenge here would be an appropriate formulation and solution of optimal control problem.

Control of temperature distribution in the systems describe by partial differential equations.

Problems may include warm-up bars, plates and three-dimensional solids. The optimal control problem can be formulated in the following form: choose a strategy for providing heat to the system in order to carry it from the state A to state B in a short time and using only the small portion of energy. Quality indicator will be to an appropriate integral of the sum of one and the quadratic form corresponding to the input of energy. Control operate on a compact set. As a result we obtain the optimal trajectories of the inputs and time required for the implementation of a fixed problem. Discrete model could be provided by finite element method.

12.1.3 Literature review

Most of the observable natural and physical processes are modelled by using Partial Differential Equations (PDE). For only few of them the solution can be represented in a closed analytical form. Therefore, the key for conveying PDE models into real applications lies in the development of numerical algorithms that result in approximated solutions. The rate of complexity grows even higher for the optimization problems, where constraints are given as PDEs. For these problems crucial, before choosing the relevant numerical method, is to resolve the issue of existence and uniqueness of solutions. This is the reason why most of the optimization problems are formulated in such a way that the objective function is convex and the set of admissible decision parameters is compact. As we will show later, the convex problems are also more convenient for numerical treatment due to the fact, that in case of quadratic objective function, the insertion of the adjoint state, used for gradient evaluation, comes very natural and the adjoint PDE reflects the original system.

The study of optimal control goes back to 1950s. In that time two important advances were made. One was Dynamic Programming, founded by Richard Bellman [30]. Dynamic Programming is a procedure that reduces the search for an optimal control function to finding the solution of a partial differential equation (the Hamilton–Jacobi–Bellman Equation) [336]. The other was the Pontryagin Maximum Principle [138], a set of necessary conditions for a control function to be optimal. Based on these theories numerous computational techniques were developed in the 1960s and 1970s [173]. With the exception of simplest cases, however, it is impossible to express controls in an explicit feedback form. Large parts of the theory taken from of finite-dimensional optimal control was successfully applied to the systems governed by PDEs.

There is a good number of works on the various aspects of the optimization under PDE constraints. For the comprehensive study a reader is refereed to the monographs by Lions [210] and Troltzsch [317]. The books contain deep investigations on existence of linear and semilinear PDEs, existence of optimal controls, necessary optimality conditions and adjoint equations, second-order sufficient optimality conditions s well as introduction to numerical methods. The up-to-date quadratic optimal control theory for PDEs over a finite or infinite time horizon, and related differential (integral) and algebraic Riccati equations can be found in the two-volumes book by Lasiecka and Triggiani [197]. Interesting work on the problems of shape optimization of nonlinear PDEs is published in the monograph by Mysliński [248].

Reader interested, in particular, application of PDEs control is encouraged to study the problems of parametric optimization in mechanical systems. Intensive researches on the semi-active control of systems represented by PDE have opened a lot of unsolved problems. One of them occurs if the cost functional is limited to a fixed period of time. The switching scheme for control is given in implicit form, and it depends on state and adjoint state variables. Solving the Two-Point Boundary Value Problem is time consuming and in general difficult to solve in the case of a multidimensional problem for vibrating system. Another open problem that occurs in the case of systems described by PDE is a stability and control of a switched system. The asymptotic stability of a switched system can be proven in the simplest cases only. The extensive research on these problems was treated in terms of the Lie algebra, and it was done by D. Liberzon et al. in the following works [244], [245].

12.1.4 A few motivating examples

Among the variety of different optimal control problems several of them seem to be typical, in particular, those that consist of linear PDEs and quadratic objective functions. We present here a few examples of boundary and distributed control. We start with a typical academic problem. Then we present some more complex problems that are related to real engineering issues.

12.1.4.1 Boundary control: optimal stationary heating

We consider a body that occupies some two or three-dimensional spatial domain. The body can be heated or cooled and the heat source, which is a control parameter, is applied to its boundary. The goal is to find a control $u(x)$ such that the resulting stationary temperature distribution $y(x,u)$ is as close as possible to some specified distribution $y_d(x)$. The model used in this example is a simplified under the assumption that the heat conduction parameter is constant. The optimal control problem is written as follows

$$\begin{aligned} &\min \quad \left[J = \frac{1}{2}\int_{\Omega} (y(x,u) - y_d(x))^2 \,\mathrm{d}x + \frac{\alpha}{2}\int_{\Gamma} u^2 \,\mathrm{d}s(x) \right], \\ &\text{subject to the constraints} \quad -\Delta y = 0 \quad \text{on} \quad \Omega\,, \\ &\qquad \frac{\partial y}{\partial n} = \lambda(u-y) \quad \text{on} \quad \Gamma\,, \\ &\qquad u \in U\,. \end{aligned} \tag{12.4}$$

The state equation in (12.4) is of elliptic type and the problem is called a linear-quadratic elliptic boundary control problem.

12.1.4.2 Distributed control: optimal vibrations

Here we consider a vibrating object subjected to force distributed on its surface. The state of such a system is transversal displacement $y(x,u,t)$ and the control input $u(x,t)$ is force density acting in the vertical direction. The objective of control is the excite the desired vibrations, here denoted by $y_d(x,t)$. We assume a finite time horizon. The optimization problem can be written in the following form

$$\min \quad \left[J = \frac{1}{2}\int_0^T \int_\Omega (y(x,u,t) - y_d(x,t))^2 \,\mathrm{d}x\,\mathrm{d}t \;.+ \right.$$
$$\left. \frac{\alpha}{2}\int_0^T \int_\Omega (u(x,t))^2 \,\mathrm{d}x\,\mathrm{d}t \right], \tag{12.5}$$

$$\text{subject to the constraints} \quad \begin{aligned} \tfrac{\partial^2 y}{\partial t^2} - \Delta y = u \quad &\text{on} \quad \Omega\,, \\ y(t=0) = y_0 \quad &\text{on} \quad \Omega\,, \\ \tfrac{\partial y}{\partial t}_{|t=0} = y_1 \quad &\text{on} \quad \Omega \\ y = 0 \quad &\text{on} \quad \Gamma \\ u \in U\,. & \end{aligned} \tag{12.6, 12.7}$$

This problem is referred as linear quadratic hyperbolic control problem with distributed control. The treatment of these type of problems is much more difficult, due to the weaker smoothing properties of the associated solutions [317].

12.1.4.3 Identification of a source of pollution

The following example was presented in [184]. It shows how an optimal control problem can be used to identify the source of a pollution flow. The author considered a river or a lake with polluted water. The state of the system reflects pollution concentration, and it is denoted by $y(x,t)$. The location of pollution source is denoted by $a \in \Omega$ and it is a parameter to be determined. $s(t)$ is the flow rate of pollution and it is assumed to be known while y_{obs} is the measured state. The optimization problem is as follows

$$\begin{aligned} \min \quad & \left[J = \int_0^T \int_\Omega (y(x,t) - y_{obs})^2 \,\mathrm{d}\Omega\,\mathrm{d}t\right], \\ \text{subject to the constraints} \quad & \frac{\partial y}{\partial t} - \Delta y + V\cdot\nabla y + \sigma y = s(t)\delta_a \quad \text{on} \quad \Omega\times[0,T]\,, \\ & \frac{\partial y}{\partial t} = 0 \quad \text{on} \quad \Gamma\times[0,T]\,, \\ & y(x,y) = y_0 \quad \text{on} \quad \Omega\,. \end{aligned} \tag{12.8}$$

The state equation is of parabolic type. The control parameter is a two or three-dimensional vector of space coordinates.

12.1.4.4 Optimization of moving load trajectories

The current trend to lightening structures requires new and more efficient methods to decrease the vibration levels. In large-scale engineering structures like bridges or viaducts that span gaps, beams must resist loads due to heavy and fast vehicles. The construction of new bridges of sufficiently higher load carrying capacity is usually limited by costs. Moreover, static strengthening can be restricted for technological reasons. Existing old weak structures can be reinforced by supplementary supports with magneto or electro-rheological dampers controlled externally (see the Figure 12.3).

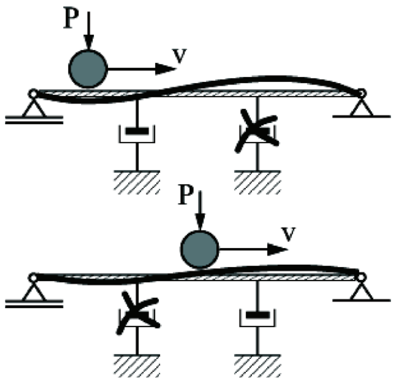

Fig. 12.3 Optimization of moving load trajectory.

The optimization problem can be stated as follows. For the semi-active dampers find a control policy such that the moving load trajectory is close to the straight line. Formally, this can be written as following

$$\min \quad \int_0^{t_f} [w(vt,t)]^2 \, \mathrm{d}t \tag{12.9}$$

subject to the constraints

$$EI\frac{\partial^4 w(x,t)}{\partial x^4} + \mu\,\frac{\partial^2 w(x,t)}{\partial t^2} = -\sum_{i=1}^{m} u_i(t)\,\frac{\partial w(x,t)}{\partial t}\,\delta(x-a_i) + P\,\delta(x-vt)$$
$$w(x=0,t)=0\,, \quad w(x=l,t)=0\,, \quad \left(\frac{\partial^2 w(x,t)}{\partial x^2}\right)_{|x=0,x=l} = 0$$
$$w(x,t=0)=0\,, \quad \dot{w}(x,t=0)=0\,,$$

where $w(vt,t)$ is the vertical deflection of the Euler-Bernoulli beam under the travelling load and the decision parameter u is the vector of damping coefficients.

The idea of straight-line passage is based on the principle of a two-sided lever. The first part of the beam which is subjected to a moving load is supported by semi-active damper placed on the rigid base. The last damper is active while the second is passive. At this stage, a part of the beam is turned around its centre of gravity, levering the right hand part with a passive damper attached. The temporal increment of displacements on the right hand part of the beam enables us to exploit it during the second stage of passage.

To the potential application of semi-active damping methods, we can also include the robotic systems, in particular the linear guideways. The straight or precisely controlled trajectory of a moving object is essential in some technological processes such as cutting (flame, plasma, laser, textile, waterjet, glass cutting) or bonding (glueing, welding, soldering). Other especially suited areas of application for linear guideway systems are large-format plotters and scanners for various industries as well as devices in medical and semiconductor technologies. New solutions can accelerate procedures and decrease the mass and size of guideways supporting carriages.

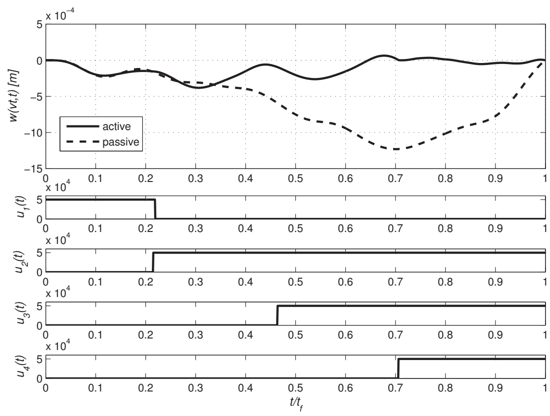

Fig. 12.4 Extremal deflection trajectory and controls.

Extensive studies on the problem are presented in the papers [21], [22]. Here we recall only the major result: For a wide range of system parameters there exists at least one semi-active switching control method such that it outperforms the best passive (uncontrolled) case. The near optimal solution requires a finite number of switchings for every control. The optimization was performed on the finite-dimensional model by introducing a relevant adjoint state. The numerical example is presented in the Figure 12.4. Solid line indicates the optimal trajectory that strongly outperforms passive case.

12.1.5 Convex problems with PDE constraints: solving methodology

Most of the numerical algorithms developed for optimization of partial differential equations are dedicated to convex problems, in particular, for the problems, where the objective function is quadratic. There are few reasons for this. The major is that the convex problem has a unique solution, and therefore, the gradient methods can be used. Moreover, the quadratic functions enable us to derive simple formulas for the gradients by introducing adjoint state. It is also worth mentioning that for such a case, the partial differential equation for the adjoint state reflects the properties of to the state equation. It this section we present in brief a methodology for solving a convex quadratic optimal control problem for the partial differential equation of elliptic type.

Let us consider the following optimization problem

$$
\begin{aligned}
&\min && \left[J = \frac{1}{2}\|y(x,u) - y_d(x)\|^2_{L^2(\Omega)} + \frac{\alpha}{2}\|u\|^2_{L^2(\Omega)}\right], \\
&\text{subject to the constraints} && Ay(x,u) = f(x) + u \quad \text{on} \quad \Omega\,, \\
& && By(x,u) = g(x) \quad \text{on} \quad \Gamma\,.
\end{aligned}
\tag{12.10}
$$

Here x, y and y_d stand for coordinate vector, state and desired state, respectively. The decision (or control) parameter to be optimized is denoted by $u \in U$. The domain and its boundary are Ω and Γ (see the Figure 12.5). The operators A (differential) and B (differential or algebraic) determine the major PDE and the boundary conditions. Finally f and g are the functions of x or constants. Let us assume that v is the global

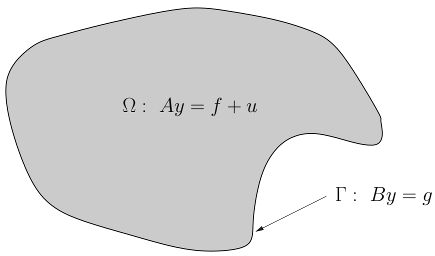

Fig. 12.5 State and boundary equations.

minimizer for the problem (12.10). Then for every $u \in \mathscr{U}$ the following condition is satisfied

$$
\left\langle \frac{dJ(y,u)}{du}, v - u \right\rangle \geq 0\,,
\tag{12.11}
$$

where $\langle\cdot\rangle$ is the inner product in Hilbert space. The term $\frac{dJ(y,u)}{du}$ stands for the derivative of objective function with respect to control and will appear later as the gradient

$\nabla_u J$ for finite-dimensional model approximation. First we tend to reformulate the condition (12.11) such that the derivative is given in explicit form. After differentiation of J we rewrite (12.11) as follows

$$\langle y(x,u) - y_d(x), y(x,v) - y(x,u) \rangle + \alpha \langle u, v-u \rangle \geq 0\,, \tag{12.12}$$

We introduce the adjoint state p that solves the following equations

$$\begin{aligned} A^* p &= y - y_d \quad \text{on} \quad \Omega\,, \\ Cp &= 0 \quad \text{on} \quad \Gamma\,, \end{aligned} \tag{12.13}$$

where A^* is the operator adjoint to A. By using (12.13) we can rewrite the first term in (12.12) as follows

$$\langle y(x,u) - y_d(x), y(x,v) - y(x,u) \rangle_{L^2(\Omega)} = \langle A^* p, y(x,v) - y(x,u) \rangle_{L^2(\Omega)}\,. \tag{12.14}$$

Based on Green's theorem we have

$$\langle Ay, p \rangle_{L^2(\Omega)} - \langle y, A^* p \rangle_{L^2(\Omega)} = \langle y, Cp \rangle_{L^2(\Gamma)} - \langle By, p \rangle_{L^2(\Gamma)} \tag{12.15}$$

and therefore (12.14) can be written as follows

$$\begin{aligned} &\langle y(x,u) - y_d(x), y(x,v) - y(x,u) \rangle_{L^2(\Omega)} = \langle p, A\,(y(x,v) - y(x,u)) \rangle_{L^2(\Omega)} + \\ &+ \langle p, B\,(y(x,v) - y(x,u)) \rangle_{L^2(\Gamma)} - \langle Cp, y(x,v) - y(x,u) \rangle_{L^2(\Gamma)}\,. \end{aligned} \tag{12.16}$$

Using the boundary condition $p = 0$ on Γ we finally get

$$\begin{aligned} \langle y(x,u) - y_d(x), y(x,v) - y(x,u) \rangle_{L^2(\Omega)} &= \langle p, f + v - f - u \rangle_{L^2(\Omega)} = \\ &= \langle p, v-u \rangle_{L^2(\Omega)}\,. \end{aligned} \tag{12.17}$$

The condition (12.11) can now be written in the following form

$$\langle p + \alpha u, v - u \rangle \geq 0\,. \tag{12.18}$$

From (12.18) we can see that the gradient of objective function with respect to control is a simple linear function of adjoint state and control

$$\frac{dJ(y,u)}{du} = p + \alpha u\,. \tag{12.19}$$

This formula allows us to build the computational algorithm based on steepest descent method. The algorithm uses projection $\mathscr{P}$ on the set of admissible controls U. If for instance this set is bounded by two values u_{min} and u_{max} then the projection can be computed as follows

$$\mathscr{P}(\cdot) = \max\{u_{min}, \min\{u_{max}, \cdot\}\}\,. \tag{12.20}$$

In order to familiarize a reader with the presented methodology we present a simple example below. The physical object is a stretched membrane subjected to some external pressure. Let us assume that we can generate an additional pressure that acts on some specified area (see the Figure 12.6) of the membrane. The idea is to find a distribution for this additional pressure such that the resulting total membrane deflection stays as close as possible to the the common value which we denote by y_d. Adopting the following notation: x_1, x_2–spatial coordinates, y–membrane deflection, f–external pressure, u–controlled pressure, T–membrane tension, the optimization problem can be written as follows

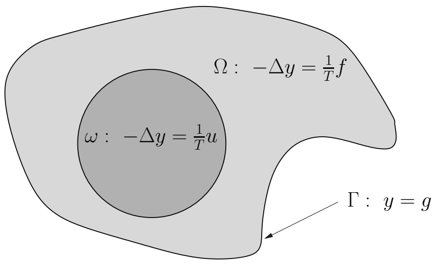

Fig. 12.6 State and boundary equations for the membrane example.

$$\min \quad \left[J = \frac{1}{2} \int_\Omega (y(x_1,x_2,u) - y_d)^2 \, \mathrm{d}x_1 \mathrm{d}x_2 + \frac{\alpha}{2} \int_\omega u^2 \, \mathrm{d}x_1 \mathrm{d}x_2 \right] ,$$

subject to the constraints

$$\begin{aligned} -\Delta y(x_1,x_2,u) &= \frac{1}{T} f(x_1,x_2) \quad \text{on} \quad \Omega \, , \\ -\Delta y(x_1,x_2,u) &= \frac{1}{T} u \quad \text{on} \quad \omega \, , \\ y(x_1,x_2,u) &= g(x_1,x_2) \quad \text{on} \quad \Gamma \, . \end{aligned} \tag{12.21}$$

We introduce the adjoint state equation as follows

$$\begin{aligned} -\Delta p &= \frac{1}{T}(y - y_d) \quad \text{on} \quad \Omega \, , \\ p &= 0 \quad \text{on} \quad \Gamma \, . \end{aligned} \tag{12.22}$$

12.2 Mesh generators

There are several efficient mesh generators in 2 or 3-dimensional domains. It is not our goal to test or evaluate them. We intend to show the real efficiency in our control problems. Two-dimensional domains can be partitioned into finite elements with the internal Felt mesh generator corduroy or external triangle. Three-dimensional domains can be split by both corduroy or external gmsh [146]. The last one is universal and can be used to meshing and visualization of data in 2 and 3 dimensional domains.

12.2.1 Internal mesh generator: corduroy

The in-built Felt mesh generator allows us to spread a mesh over the given contour. The boundary line can contain holes. The contour is described by a counter clockwise polygon while holes by clockwise polygons. Two examples are given below. The first one gives the mesh depicted in Figure 12.7. The second one describes the mesh with two holes (Figure 12.8).

```
start-node = 1
start-element = 1
triangular mesh
element-type = ctg
boundary = [
(0,50)
(20,10)
(40,0)
(40,50)
(55,70)
(95,70)
(115,50)
(115,0)
(135,10)
(155,50)
(155,90)
(115,160)
(170,404)
(122,404)
(67,160)
(0,90)
]
end

triangular mesh
boundary = [
(0,0)
(20,0)
(20,10)
(0,10)
```

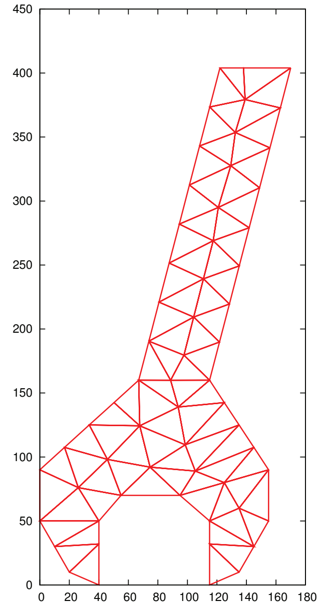

Fig. 12.7 Single computational pass.

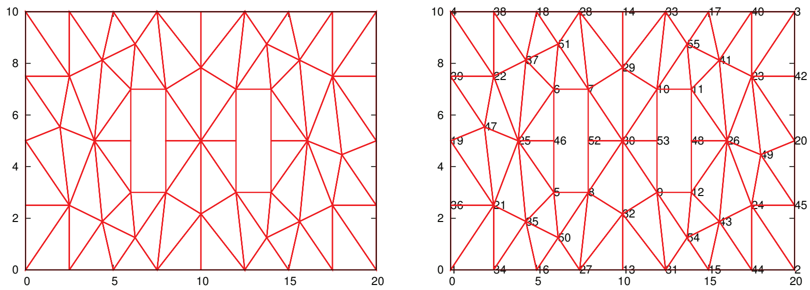

Fig. 12.8 Generated mesh without and with nodal numbers.

```
]
hole = [
(6,3)
(6,7)
(8,7)
(8,3)
]
hole = [
```

```
(12,3)
(12,7)
(14,7)
(14,3)
]
end
```

The automatic mesh point generation together with quadrilateral elements can be produced from the file given below. The resulting mesh is depicted in Figure 12.9.

```
start-node    = 1
start-element = 1

quadrilateral grid
start    = (0,0)
end      = (10,3)
x-number = 20
y-number = 6
x-rule   = linear
y-rule   = linear

end
```

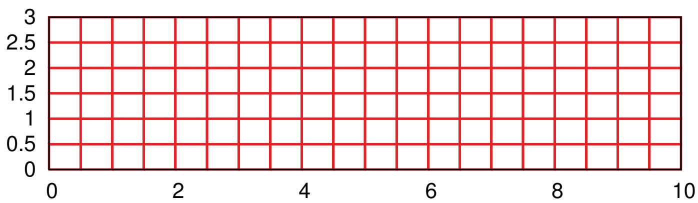

Fig. 12.9 Example of the quadrilateral linear mesh.

The next example results in the mesh shown in Figure 12.10.

```
start-node    = 1
start-element = 1

quadrilateral grid
start    = (0,0)
end      = (10,3)
x-number = 20
y-number = 6
x-rule   = logarithmic
y-rule   = reverse-logarithmic
```

```
end
```

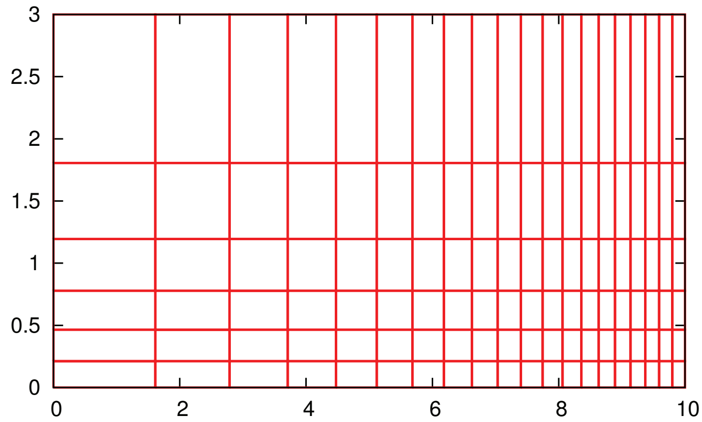

Fig. 12.10 Example of the logarithmic mesh.

12.2.2 External mesh generator: triangle

The stand-alone program triangle allows us to create a complex mesh based on various data delivered. It was written by Jonathan Richard Shevchuk. Triangle generates:

- exact Delaunay triangular mesh,
- constrained Delaunay triangular mesh,
- conforming Delaunay triangular mesh,
- Voronoi diagrams,
- high-quality triangular meshes.

The command can be supplemented with several options, which allows us to control the meshing process: mesh density, regularity, maximum triangle area constraint, regional attributes, etc.

12.3 Elasticity problem

As a test example we will consider a hip joint implant. Nowadays, there are over 100 million people aged over 50 in the USA and Europe. One of the consequences of aging is deterioration of bone quality; also, gradual loss of bone mass is observed, starting from 30 – 40 years. One of the frequent consequences of aging of the skeletal system are degenerative changes often occurring in the hip and the knee, which may lead to even serious handicap.

Human joints like hip, knee, elbow, etc. have very complicated geometry. To analyze stress states existing in such joints before and after arthroplasty it is necessary to apply numerical methods like FEM. Usually on the basis of computer tomography images the three–dimensional meshes of FE are elaborated. The next problem is to model the properties of bones including possibly their anisotropy, and in the case of implants, the behaviour of the system bone–cement–implant or bone–implant for cementless protheses.

The aim of the present example is to show some elements of optimal choice of the implanted stem geometry. They mainly deal with stress and displacement analysis in the hip joint, after arthroplasty. In the same way both cement and cementless prostheses can be analyzed. The influence of stem dimensions is discussed. An optimality criterion ensures that stress distribution is optimal comparing with healthy hip joint. Moreover, the stress distribution in the neighbourhood of the prosthesis can be neither extremely low (to avoid stress shielding) nor extremely elevated (to prevent decohesion).

12.3.1 Stress distribution analysis in healthy and diseased bones

Both 2 and 3 dimensional analysis carried on with the finite element method shows the distribution of stresses in the bone. Nowadays even complex form of the bone or implant–bone couple does not exhibit considerable difficulties. Even simple radiology enables to prepare the real geometry in a particular case (Figure 12.12). Tomography allows almost automatic passage from the human body to the finite element geometric model (Figure 12.11). 2-D model of the bone is extremely simplified. It enablesus us only to exhibit the strategy that can guide the engineer through the optimization details. In practical use 3-D model should be elaborated. The investigation should give the answer about the placement of the implant in the trabecular bone. Bone rebuilding ability should be also taken in the optimization process. More, the X-ray pictures show the density change in parts of a bone and in the same way the lower relative load carrying capacity. The external contour can be simply determined in most parts. The difficulty appears in the contact region between the acetabulum located in the pelvis and the head of the bone. The internal contour determines the places where the density (i.e. X-ray transmission) has a predefined low value. Since bone density and bone mineralization change continuously and the bone is laced with voids, the internal contour is much more complex. If the contour line separate voids or liquid–like media, the precision has minor role. However, in the trabecular bone, where we have mostly the lattice structure, rigid enough with low average material density, the proper geometrical and material data are important.

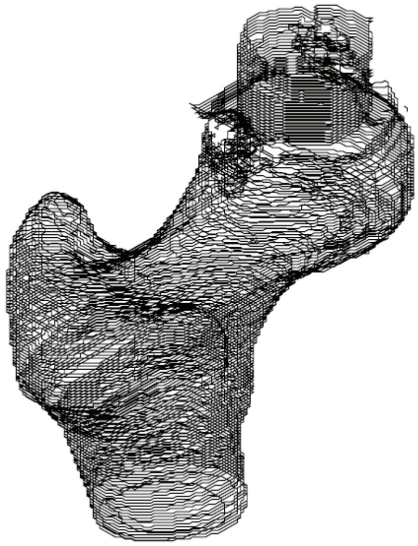

Fig. 12.11 Contour lines of the hip joint obtained by the tomography.

Fig. 12.12 Hip joint implant.

Although for researches the realistic and precise model is not so important, in the case of diseased bones both the geometry and material properties may differ considerably from the typical ones.

There are two main reasons of prosthesis loosening:

- biological, strongly depended on the individual predispositions, which should be investigated on the microscopic level; wear of the implant, caused both by the biological factors and mechanical factors (for example friction), migration of wear debris to the contact surface of bone, cement and steel, slacken rebuilding of the bone,
- mechanical, when the stress limits are exceed; it results in the fatigue, especially in the case of cyclic load, crack propagation and material fragmentation.

12.3.2 Optimization problem

The following data must be prepared for the 2-dimensional modelling:

- bone and implant contours geometry,
- position of the implant in the bone,
- rigidity and strength distribution in the bone,
- destructive temperature limitations for cement prostheses,
- heat generation in cemented regions,
- geometrical constraints to avoid penetrations by steel stem and ensure tissue nutrition.

The full study of the problem is complex and exceeds the scope of the chapter. We will only show the simple geometrical optimization.

The boundary conditions in the form of zero values of the solution (displacements) was applied to the lower edge of the area. The head of the prosthesis is loaded with concentrated force. Three sub-areas were selected: the bone of fine and coarse mesh and area of the endoprosthesis. The relevant material constants were introduced as for bone and steel. The objective function was assumed as the following

$$\min[\max(\sigma_{\max} - \sigma_{\min})], \tag{12.23}$$

where $\sigma_{\max}$ and σ_{min} are maximal and minimal principal stresses in finite elements, respectively.

We can apply here a lot of possible functions, depending on the aim, that can be defined in term of nodal or elemental parameters. For example it can be:

- minimum of the greatest principal stress,
- minimum of the greatest shear stress,
- minimum of the stress diversity in the domain,
- minimum of the surface of the contact between the bone and the steel stem, with the given level of stresses,
- minimum of the surface of endoprothesis,

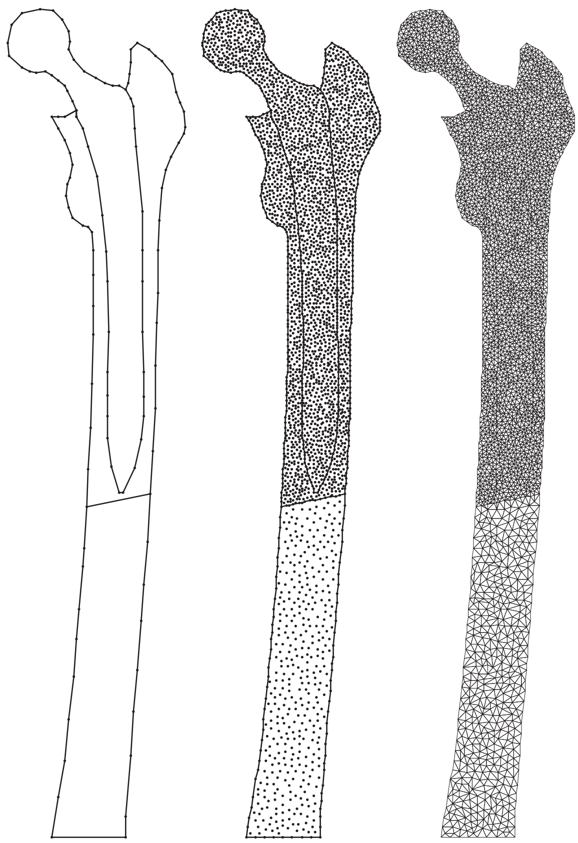

Fig. 12.13 Contours of bone and implant, nodes distribution and finite elements in the domain.

- mane other derived from the wide mechanical results obtained from a single finite element solution.

The contour of the stem prosthesis is altered. It is parameterized with four parameters $x_1,\ldots,x_4$. The restrictions are as follows:

$$0 \le x_i \le 1, \quad i = 1,\ldots,4 \tag{12.24}$$

$$x_1,x_2,x_3,x_4 \ge 0,8 \tag{12.25}$$

$$0 \le x_1^2 + x_2^2 + x_3^2 + x_4^2 \le 12\ . \tag{12.26}$$

The starting point was assumed at $x_i = 0.5$, $i = 1,\ldots,4$. Final stresses are depicted in Figure 12.14.

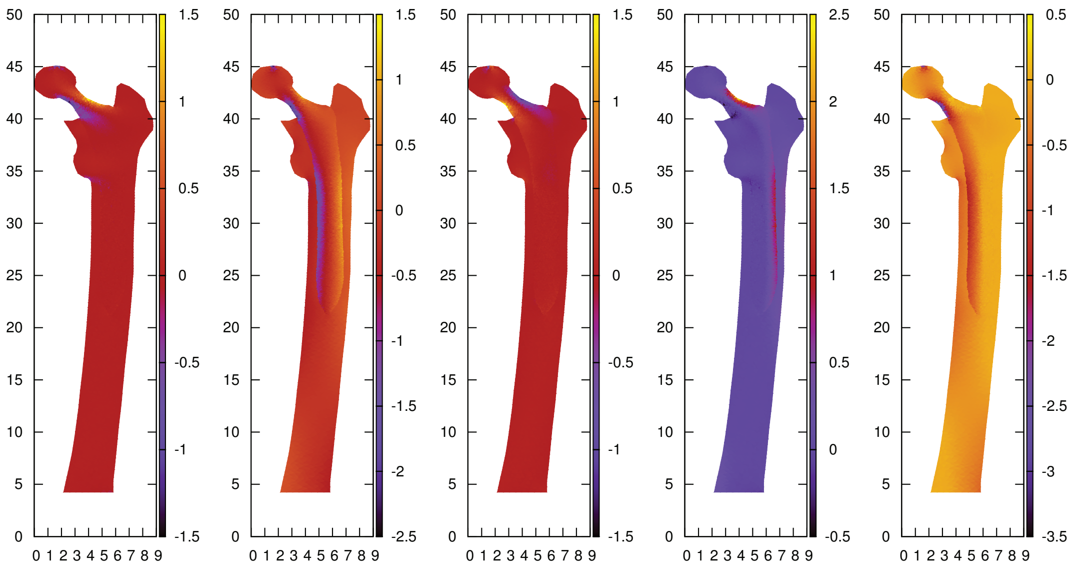

Fig. 12.14 Resulting stresses σ_x, σ_y, τ_{xy}, σ_{max}, σ_{min}.

12.4 Thermal problems

We consider here a heat transfer problem, reduced to the stationary state, described by the equation

$$\nabla T(x,y) = q(x,y) \ , \tag{12.27}$$

with boundary conditions

$$T(x,y) = \tilde{T}, \ (x,y) \in \Gamma_T, \tag{12.28}$$

and

$$p(x,y) = \tilde{p}, \ (x,y) \in \Gamma_p, \tag{12.29}$$

We can impose boundary conditions of a different type:

- specified heat flux
$$\dot{q} = -k\frac{\mathrm{d}T}{\mathrm{d}n} \tag{12.30}$$
- insulation
$$\frac{\mathrm{d}T}{\mathrm{d}n} = 0 \tag{12.31}$$
- convection
$$h(T_\infty - T_0) = -k\frac{\mathrm{d}T}{\mathrm{d}n} \tag{12.32}$$
- radiation
$$\varepsilon\sigma(T_{sur}^4 - T_0^4) = -k\frac{\mathrm{d}T}{\mathrm{d}n} \tag{12.33}$$

12.4.1 Geometry definition

There are two possible ways of the definition of two-dimensional contours:

- use of built-in mesh generator in Felt [150] package,
- use of external mesh generator triangle.

12.4.2 Structure of the computational process

12.4.3 Single computational pass

The scheme in Figure 12.15 presents computational stages in temperature control problem. Below segments of the solution are described:

task.cdr is the basic definition of the geometry for the mesh generator,
corduroy generates the simple input file for the Felt program,

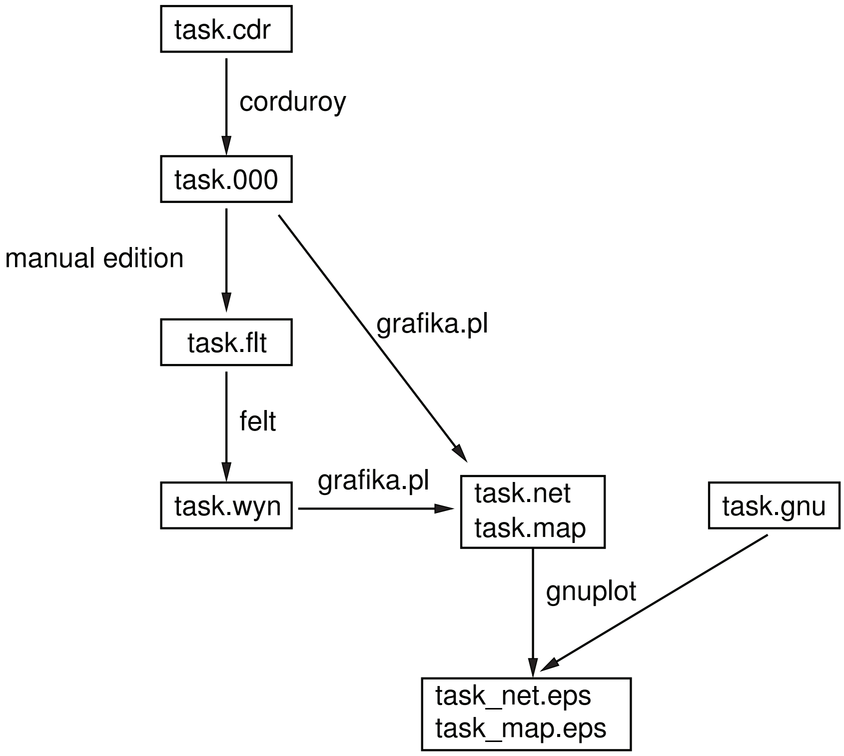

Fig. 12.15 Single computational pass.

task.000 is the basic task file that must be manually supplemented with remaining problem data,
task.flt is the final Felt input file,
grafika.pl is the graphical output procedure that allows us to verify the geometry contained in both 'task.000' and 'task.flt'; it produces the mesh in gnuplot format task.net; if is used to results, it produces a table of parameters task.map that can be transformed into coloured map of solution parameters,
task.wyn contains results of single step of computations,
task.gnu is a gnuplot file that is used to generate final PostScript graphical files task_net.eps and task_map.eps.

This set of procedures is specially immersed in the Ipopt optimization procedure.

12.4.4 Examples of structures

We consider the mesh as shown in Figure 12.8.

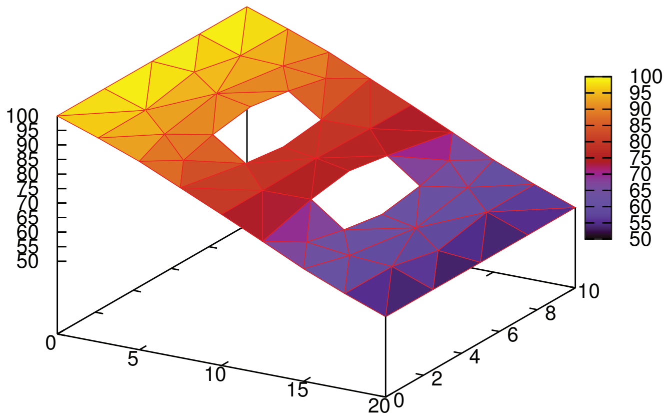

Fig. 12.16 Resulting map of temperature.

12.4.4.1 Control problem

Thermal elements only have one degree of freedom per node. It is the nodal temperature. Prescribed and initial temperatures are assigned using constraints, Tx=, and Ty=, respectively. Heat sources are specified as an external load placed on the right-hand-side vector of the resulting system of algebraic equations, using nodal forces (Fx=). For uniform heat sources, simply apply the same value to all nodes.

Let us consider a rectangular plate, insulated at the top and at the bottom. The left edge is held at a fixed temperature of 100 degrees and the other three are exposed to a free stream temperature of 50 degrees. The control function J determines the difference between the given temperature distribution $\tilde{T}(x,y)$ and actually computed one $T(x,y)$:

$$J = \frac{1}{2}\int_{\Omega} \big(T(x,y) - \tilde{T}(x,y)\big)^2 \,\mathrm{d}x\,\mathrm{d}y\ . \tag{12.34}$$

A table of nodal temperatures at thermal equilibrium is given as results. The input file for Felt [150] taken from the Felt Web pages has the following form:

```
problem description
```

```
title="Heat Transfer"
nodes=5 elements=4 analysis=static-thermal

nodes
1  x=0 y=0 constraint=insulated
2 x=2 y=0 constraint=free
3 x=2 y=2
4 x=0 y=2 constraint=insulated
5 x=1 y=1 constraint=free

ctg elements
1 nodes=[1,2,5] material=steel
2 nodes=[1,5,4]
3 nodes=[4,5,3]
4 nodes=[2,3,5] load=convection

material properties
steel    t=1  kx=25 ky=25

distributed loads
convection values=(1,20) (2,50)

constraints
insulated Tx=100
free Tx=u

end
```

We must prepare several insertions. They contain informations to be taken into account in the control program:

- The number of variables under control

```
n=8;
```

- Lower and upper bounds of all variables

```
x_L[i]=-2.0;
x_U[i]= 2.0;
```

The file can also give assignments to each variable. In such a case it has a form:

```
x_L[0]=-2.0;
x_U[0]= 2.0;
x_L[1]=-2.0;
x_U[1]= 2.0;
x_L[2]=-2.0;
x_U[2]= 2.0;
x_L[3]=-2.0;
```

```
x_U[3]= 2.0;
x_L[4]=-2.0;
x_U[4]= 2.0;
x_L[5]=-2.0;
x_U[5]= 2.0;
x_L[6]=-2.0;
x_U[6]= 2.0;
x_L[7]=-2.0;
x_U[7]= 2.0;
```

- Initial values

```
x[0]=0.1;
x[1]=0.1;
x[2]=0.1;
x[3]=0.1;
x[4]=0.1;
x[5]=0.1;
x[6]=0.1;
x[7]=0.1;
```

Above the following parameters must be supplied: t – thickness, kx – conductivity in steel in *x* direction, ky – conductivity in *y* direction,

For a ctg element the definition values=(1,20) (2,50) specifies a convection coefficient (the load is named here 'convection') of 20, a free stream temperature of 50, and that the convection acts on the edge defined by local element node numbers 1 and 2 (i.e.. by global nodes 2 and 3).

12.5 Control of vibrations

Although the idea of passive vibration absorbers was well known in the machine industry and visible implementation could even be found in sport equipment as bows, sky or tennis rackets, the application to civil engineering structures had minor importance. The reason was that absorbers can not accumulate vibration for a longer time. That is why in the global count vibration absorbers (passive dampers) were not as promising as active dampers. Problems with actively controlled vibration were investigated in [136, 134, 270]. The beam was subjected in the middle of the span by the force generated by actuators. Open loop and closed loop control were tested. However, the system of actuators seems to be hardly performed, since large forces are required in real cases.

More efficient displacement decrease of the moving vehicle or beam structure can be performed by the use of semi-active or active dampers. From the practical point of view the arrangement in which we propose to change the damping parameters is much more efficient than actuators. What is more, the dampers do not require power supply as high as in the case of actuators [232].

12.5.1 Membrane

12.5.1.1 Definition of the problem

Let us consider a square domain Ω_1 that contains subdomain Ω_2 (Figure 12.17. We

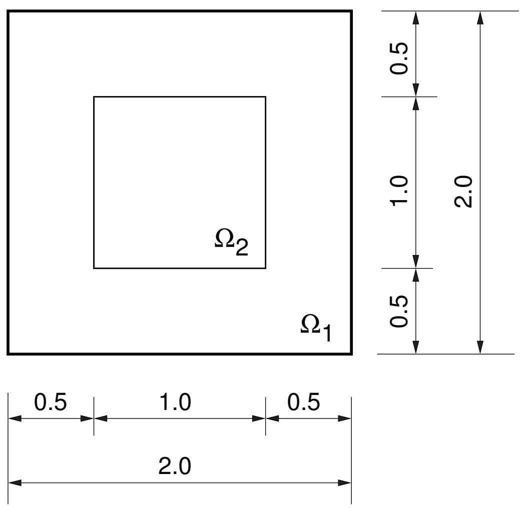

Fig. 12.17 Domain Ω_1 and subdomain Ω_2 of the test problem.

assume small displacements, i.e. significantly smaller than dimensions in x and y. The following equation describes the transverse displacements:

$$k\Delta w(x,y) = p(x,y), \quad (x,y) \in \Omega_1 \ . \tag{12.35}$$

k is the rigidity of the membrane. It corresponds to the forces that tense the membrane surface. k represents the force per unit length in the plane (x,y). $p(x,y)$ is the transverse force per unit area of the membrane. The boundary $\partial\Omega_1$ is fixed:

$$w(x,y) = 0 \ \text{ on } \ \partial\Omega_1 \ . \tag{12.36}$$

The deflection of the subdomain Ω_2 is controlled. We impose the constraints

$$w(x,y) = \tilde{w}(x,y), \quad (x,y) \in \Omega_2 \ . \tag{12.37}$$

Our problem can be written then in the form

$$k\Delta w(x,y) = p(x,y) + u(x,y), \quad (x,y) \in \Omega_1 \ . \tag{12.38}$$

$p(x,y)$ is defined on Ω_1, $u(x,y)$ is defined on Ω_2.

12.5.1.2 Finite element solution

We use the coarse triangular mesh as depicted in Figure 12.18. The description of

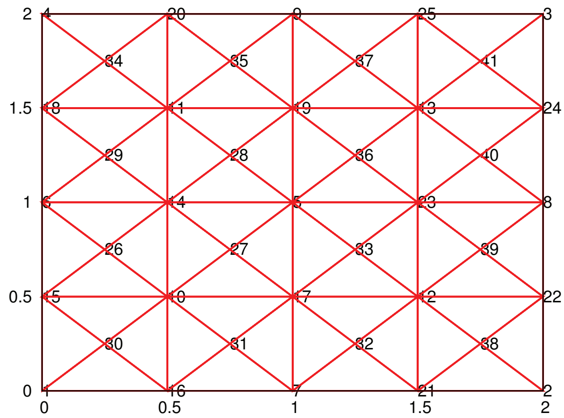

Fig. 12.18 Mesh assumed in analysis.

the mesh geometry can be introduced in different ways: manually, by using the mesh generator corduroy, triangle or other meshing tool. We simply used the following corduroy file:

```
triangular mesh
boundary = [
(0,0)
(2,0)
(2,2)
(0,2)
]
end
```

Then we complete the resulting file with appropriate problem definitions:

- forces placed to nodes being in Ω_2,
- material assigned to entire mesh or to selected finite elements,
- definitions of material parameters in the block 'material properties',
- definitions of forces in the block 'forces',
- definitions of boundary conditions in the block 'constraints'.

Below we show the exemplary model file.

```
problem description
title="membrane"
nodes=41 elements=64  analysis=static-thermal
```

```
nodes
1  x=0   y=0   z=0  constraint=insulated
2  x=2   y=0   z=0  constraint=insulated
3  x=2   y=2   z=0  constraint=insulated
4  x=0   y=2   z=0  constraint=insulated
5  x=1   y=1   z=0  constraint=free          force=force5
6  x=0   y=1   z=0  constraint=insulated
7  x=1   y=0   z=0  constraint=insulated
8  x=2   y=1   z=0  constraint=insulated
9  x=1   y=2   z=0  constraint=insulated
10  x=0.5   y=0.5   z=0 constraint=free      force=force10
11  x=0.5   y=1.5   z=0 constraint=free      force=force11
12  x=1.5   y=0.5   z=0 constraint=free      force=force12
13  x=1.5   y=1.5   z=0 constraint=free      force=force13
14  x=0.5   y=1   z=0  constraint=free       force=force14
15  x=0   y=0.5   z=0  constraint=insulated
16  x=0.5   y=0   z=0  constraint=insulated
17  x=1   y=0.5   z=0  constraint=free       force=force17
18  x=0   y=1.5   z=0  constraint=insulated
19  x=1   y=1.5   z=0  constraint=free       force=force19
20  x=0.5   y=2   z=0  constraint=insulated
21  x=1.5   y=0   z=0  constraint=insulated
22  x=2   y=0.5   z=0  constraint=insulated
23  x=1.5   y=1   z=0  constraint=free       force=force23
24  x=2   y=1.5   z=0  constraint=insulated
25  x=1.5   y=2   z=0  constraint=insulated
26  x=0.25   y=0.75   z=0   constraint=free
27  x=0.75   y=0.75   z=0   constraint=free  force=force27
28  x=0.75   y=1.25   z=0   constraint=free  force=force28
29  x=0.25   y=1.25   z=0   constraint=free
30  x=0.25   y=0.25   z=0   constraint=free
31  x=0.75   y=0.25   z=0   constraint=free
32  x=1.25   y=0.25   z=0   constraint=free
33  x=1.25   y=0.75   z=0   constraint=free  force=force33
34  x=0.25   y=1.75   z=0   constraint=free
35  x=0.75   y=1.75   z=0   constraint=free
36  x=1.25   y=1.25   z=0   constraint=free  force=force36
37  x=1.25   y=1.75   z=0   constraint=free
38  x=1.75   y=0.25   z=0   constraint=free
39  x=1.75   y=0.75   z=0   constraint=free
40  x=1.75   y=1.25   z=0   constraint=free
41  x=1.75   y=1.75   z=0   constraint=free

ctg elements
1    nodes=[23,13,36]    material=steel
2    nodes=[26,10,14]
3    nodes=[34,20,4]
4    nodes=[29,18,6]
5    nodes=[5,27,17]
6    nodes=[17,33,5]
7    nodes=[5,33,23]
8    nodes=[19,37,9]
9    nodes=[16,31,10]
```

```
10   nodes=[15,30,10]
11   nodes=[19,35,11]
12   nodes=[6,14,29]
13   nodes=[22,39,12]
14   nodes=[21,38,12]
15   nodes=[25,9,37]
16   nodes=[24,41,13]
17   nodes=[28,11,14]
18   nodes=[27,5,14]
19   nodes=[10,26,15]
20   nodes=[10,30,16]
21   nodes=[32,12,17]
22   nodes=[10,31,17]
23   nodes=[34,4,18]
24   nodes=[36,13,19]
25   nodes=[11,28,19]
26   nodes=[11,35,20]
27   nodes=[12,32,21]
28   nodes=[12,38,22]
29   nodes=[40,13,23]
30   nodes=[12,39,23]
31   nodes=[13,40,24]
32   nodes=[13,41,25]
33   nodes=[6,15,26]
34   nodes=[14,6,26]
35   nodes=[10,17,27]
36   nodes=[14,10,27]
37   nodes=[5,19,28]
38   nodes=[14,5,28]
39   nodes=[11,18,29]
40   nodes=[14,11,29]
41   nodes=[1,16,30]
42   nodes=[15,1,30]
43   nodes=[7,17,31]
44   nodes=[16,7,31]
45   nodes=[7,21,32]
46   nodes=[17,7,32]
47   nodes=[12,23,33]
48   nodes=[17,12,33]
49   nodes=[11,20,34]
50   nodes=[18,11,34]
51   nodes=[9,20,35]
52   nodes=[19,9,35]
53   nodes=[5,23,36]
54   nodes=[19,5,36]
55   nodes=[13,25,37]
56   nodes=[19,13,37]
57   nodes=[2,22,38]
58   nodes=[21,2,38]
59   nodes=[8,23,39]
60   nodes=[22,8,39]
61   nodes=[8,24,40]
62   nodes=[23,8,40]
63   nodes=[3,25,41]
```

```
64   nodes=[24,3,41]

material properties
steel  t=1  kx=25 ky=25

forces
force5  Fx=100.
force10 Fx=100.
force11 Fx=100.
force12 Fx=100.
force13 Fx=100.
force14 Fx=100.
force17 Fx=100.
force19 Fx=100.
force23 Fx=100.
force27 Fx=100.
force28 Fx=100.
force33 Fx=100.
force36 Fx=100.


constraints
insulated Tx=0
free Tx=u

end
```

12.5.1.3 Example 1

The membrane subjected to the point force $F = 10$ placed at the centre $x = 1, y = 1$ is depicted in Figure 12.19. We wish to subject the membrane to a set of additional forces (control forces) placed in the subdomain Ω_2 to decrease displacements and to get the flat diagram of displacements in $\Omega_2 : w(x,y) = 0.5625$ (Figure 12.20). The considered problem is represented by the DOML file given by Listing 12.1.

The Ipopt procedure allows us to obtain sufficiently accurate results if 10 iterations. 9 variables were controlled and the and computations with the accuracy $2 \cdot 10^{-5}$ reduced the objective function to 0.15. Below we include the technical report.

```
******************************************************************************
This program contains Ipopt, a library for large-scale nonlinear optimization.
 Ipopt is released as open source code under the Common Public License (CPL).
         For more information visit http://projects.coin-or.org/Ipopt
******************************************************************************

This is Ipopt version 3.6.0, running with linear solver ma27.

Number of nonzeros in equality constraint Jacobian...:        0
Number of nonzeros in inequality constraint Jacobian.:        0
Number of nonzeros in Lagrangian Hessian.............:        0

Total number of variables............................:       13
                     variables with only lower bounds:        0
                variables with lower and upper bounds:       13
                     variables with only upper bounds:        0
```

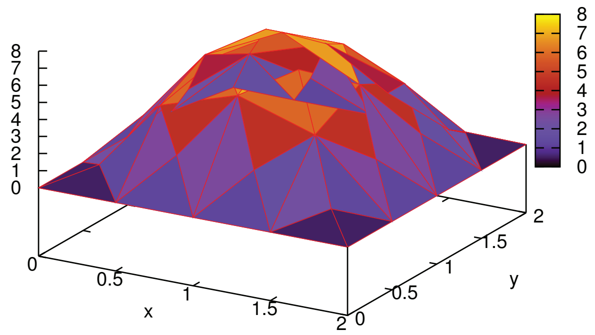

Fig. 12.19 Membrane displacements without control, under point forces $F = 100.0$ uniformly applied to Ω_1.

```
package Membrane

  optimization Membrane_opt(spaceDims = 2, domain = omega1,
                            objective = res)
    annotation(solver='pdesolve');

    Space x = space[1], y = space[2];
    Domain omega1(spaceDims=spaceDims,
               boundaryPoints=[0,0; 0,2; 2,2; 2,0]);
    Domain omega2(spaceDims=spaceDims,
               boundaryPoints=[.5,.5; .5,1.5; 1.5,1.5; 1.5,.5]);

    field Real w(free=true)  in omega1;
    field Real p = 100 in omega2;
    field input Real u  in omega2;
    parameter Real res;
    parameter Real k = 25;
  equation
    k*(der(w,x,x)+der(w,y,y)) = p + u;
    res = integral((w-0.5625)^2, omega2);
  constraint
    u = 0 in Boundary(omega1);
  end Membrane_opt;

end Membrane;
```

Listing 12.1 Membrane problem example written in DOML

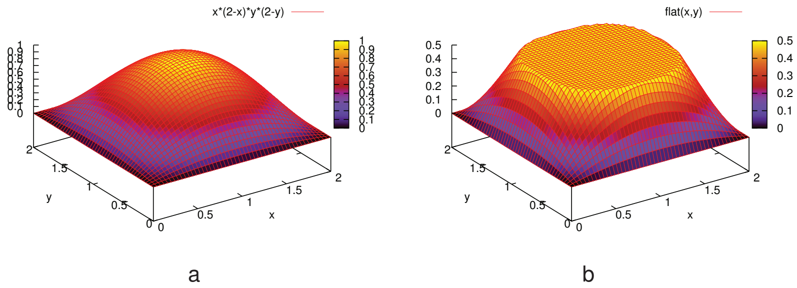

Fig. 12.20 Decreased displacements of the membrane (a) and required flattened displacements with control (b).

```
Total number of equality constraints.................:        0
Total number of inequality constraints...............:        0
        inequality constraints with only lower bounds:        0
   inequality constraints with lower and upper bounds:        0
        inequality constraints with only upper bounds:        0

iter    objective    inf_pr   inf_du lg(mu)  ||d||  lg(rg) alpha_du alpha_pr  ls
   0  4.6547998e+00 0.00e+00 3.75e-02   0.0 0.00e+00    -  0.00e+00 0.00e+00   0
   1  4.6547998e+00 0.00e+00 3.56e-02  -4.3 3.61e-02    -  9.89e-01 3.81e-06f 19
   2  4.5644531e+00 0.00e+00 3.70e-02  -0.4 1.02e-05    -  1.00e+00 1.00e+00f  1
   3  4.5644529e+00 0.00e+00 3.70e-02  -2.5 1.02e-05    -  9.98e-01 1.00e+00f  1
   4  4.5644449e+00 0.00e+00 3.70e-02  -2.5 1.02e-05    -  1.00e+00 1.00e+00f  1
   5  4.5644304e+00 0.00e+00 3.69e-02  -4.4 3.70e-02    -  1.00e+00 1.00e+00f  1
   6  4.5384396e+00 0.00e+00 3.62e-03  -5.1 9.09e+00    -  1.00e+00 1.00e+00f  1
   7  6.2430818e-01 0.00e+00 3.58e-03  -6.4 9.36e-01    -  1.00e+00 1.00e+00f  1
   8  5.9240639e-01 0.00e+00 1.46e-04  -6.9 2.25e+01    -  1.00e+00 1.00e+00f  1
   9  1.5195497e-01 0.00e+00 1.15e-04  -8.3 1.02e+00    -  1.00e+00 1.00e+00f  1
  10  1.4905032e-01 0.00e+00 4.42e-06  -9.7 3.12e+00    -  1.00e+00 1.00e+00f  1

Number of Iterations....: 10

                                   (scaled)                 (unscaled)
Objective...............:   1.4905032436874999e-01    1.4905032436874999e-01
Dual infeasibility......:   4.4235098695713256e-06    4.4235098695713256e-06
Constraint violation....:   0.0000000000000000e+00    0.0000000000000000e+00
Complementarity.........:   5.0422118675236915e-10    5.0422118675236915e-10
Overall NLP error.......:   4.4235098695713256e-06    4.4235098695713256e-06


Number of objective function evaluations             = 33
Number of objective gradient evaluations             = 11
Number of equality constraint evaluations            = 0
Number of inequality constraint evaluations          = 0
Number of equality constraint Jacobian evaluations   = 0
Number of inequality constraint Jacobian evaluations = 0
Number of Lagrangian Hessian evaluations             = 0
Total CPU secs in IPOPT (w/o function evaluations)   =      0.010
Total CPU secs in NLP function evaluations           =      0.000

EXIT: Optimal Solution Found.
```

Figure 12.21 depicts the deformed state under a set of control forces. In Table 12.1 computed nodal displacements are compared with assumed for the control. We notice that controlled variables have accurate values while remaining slightly differ. Table 12.2 shows the values of control variables in the problem.

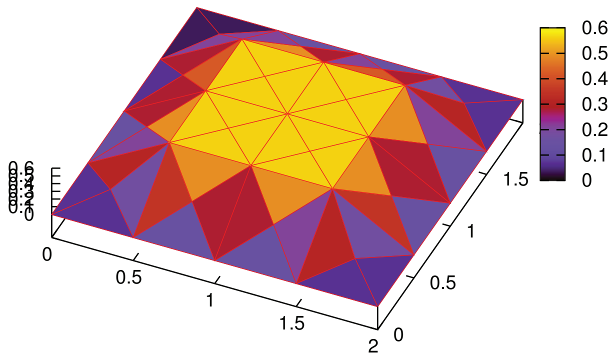

Fig. 12.21 Membrane after control.

Let us consider the same example discretized with the fine mesh (Figure 12.22). The total number of unknown equals 332. The mesh contains 612 triangular elements. The inner square contains 96 points which are controlled. The Ipopt procedure converges and after 790 steps the objective function reaches the value 2.157. Results are depicted in Figure 12.23. We can notice the flat surface if the inner square at the level 0.5625.

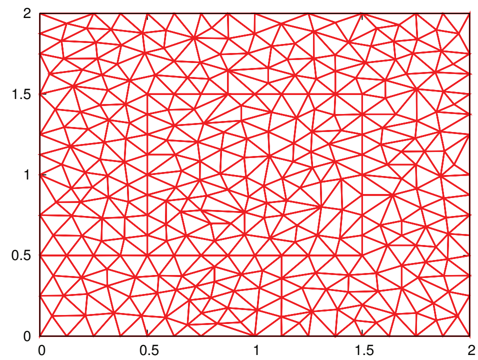

Fig. 12.22 Refined mesh of the example.

Table 12.1 Required and computed variables.

node	required	computed
1	0	0
2	0	0
3	0	0
4	0	0
5	0.56250000	0.56183
6	0	0
7	0	0
8	0	0
9	0	0
10	0.56250000	0.56212
11	0.56250000	0.56212
12	0.56250000	0.56212
13	0.56250000	0.56212
14	0.56250000	0.56221
15	0	0
16	0	0
17	0.56250000	0.56321
18	0	0
19	0.56250000	0.56321
20	0	0
21	0	0
22	0	0
23	0.56250000	0.56321
24	0	0
25	0	0
26	0.41015625	0.28133
27	0.56250000	0.56257
28	0.56250000	0.56257
29	0.41015625	0.28133
30	0.19140625	0.14053
31	0.41015625	0.28133
32	0.41015625	0.28133
33	0.56250000	0.56257
34	0.19140625	0.14053
35	0.41015625	0.28133
36	0.56250000	0.56257
37	0.41015625	0.28133
38	0.19140625	0.14053
39	0.41015625	0.28133
40	0.41015625	0.14053
41	0.19140625	1.0504

Table 12.2 Final values of control variables.

node	force
5	-0.040123
10	24.586242
11	24.586242
12	24.586242
13	24.586242
14	14.096711
17	14.096711
19	14.096711
23	14.096711
27	-0.002174
28	-0.002174
33	-0.002174
36	-0.002174

```
Total number of variables............................:       96
                     variables with only lower bounds:        0
                variables with lower and upper bounds:       96
                     variables with only upper bounds:        0
Total number of equality constraints.................:        0
Total number of inequality constraints...............:        0
        inequality constraints with only lower bounds:        0
   inequality constraints with lower and upper bounds:        0
        inequality constraints with only upper bounds:        0

iter    objective    inf_pr   inf_du lg(mu)  ||d||  lg(rg) alpha_du alpha_pr  ls
   0  1.5992910e+02 0.00e+00 6.68e-01  -1.0 0.00e+00    -  0.00e+00 0.00e+00   0
   1  1.5992910e+02 0.00e+00 3.35e-01  -1.0 6.43e-01    -  9.79e-01 7.45e-09f 28
   2  5.3582789e+01 0.00e+00 3.59e-01  -1.0 8.61e-09    -  1.00e+00 1.00e+00f  1
   3  5.3582789e+01 0.00e+00 3.59e-01  -1.7 8.61e-09    -  1.00e+00 5.00e-01f  2
   4  5.3582789e+01 0.00e+00 3.59e-01  -1.7 8.61e-09    -  1.00e+00 5.00e-01f  2
...
...
 789  2.1566350e+00 0.00e+00 2.20e-05  -6.0 8.90e-04    -  1.00e+00 2.78e-309h10
 790  2.1566350e+00 0.00e+00 9.32e-06  -6.0 8.90e-04    -  1.00e+00 1.00e+00f  1

Number of Iterations....: 790

                                   (scaled)                 (unscaled)
Objective...............:   2.1566349782056600e+00    2.1566349782056600e+00
Dual infeasibility......:   9.3169975635287168e-06    9.3169975635287168e-06
Constraint violation....:   0.0000000000000000e+00    0.0000000000000000e+00
Complementarity.........:   9.0909090909091433e-07    9.0909090909091433e-07
Overall NLP error.......:   9.3169975635287168e-06    9.3169975635287168e-06


Number of objective function evaluations             = 165930
Number of objective gradient evaluations             = 791
Number of equality constraint evaluations            = 0
Number of inequality constraint evaluations          = 0
Number of equality constraint Jacobian evaluations   = 0
Number of inequality constraint Jacobian evaluations = 0
Number of Lagrangian Hessian evaluations             = 0
Total CPU secs in IPOPT (w/o function evaluations)   =      9.830
Total CPU secs in NLP function evaluations           =     20.180
```

```
EXIT: Optimal Solution Found.
```

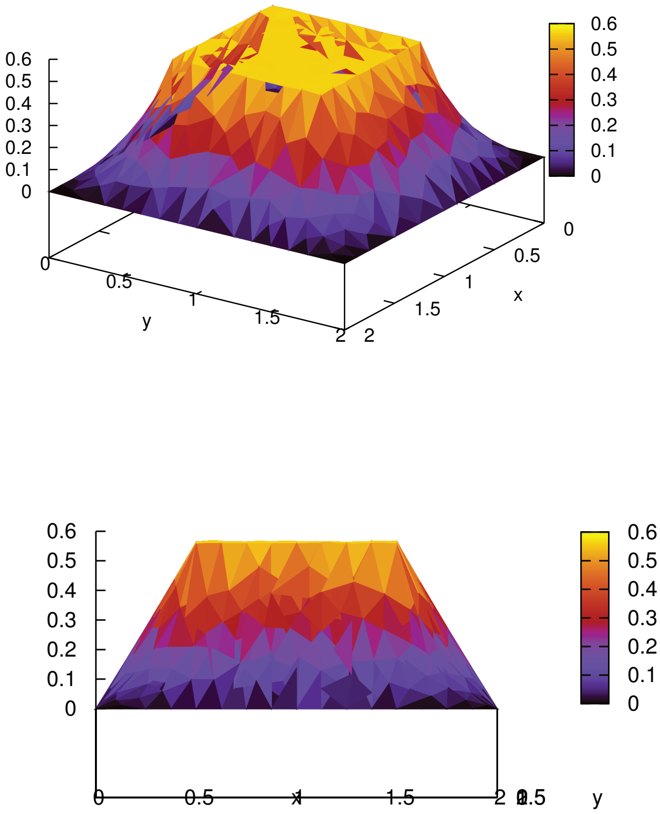

Fig. 12.23 Membrane after control – fine mesh.

The distribution of control function over Ω_2 is depicted in Figure 12.24.

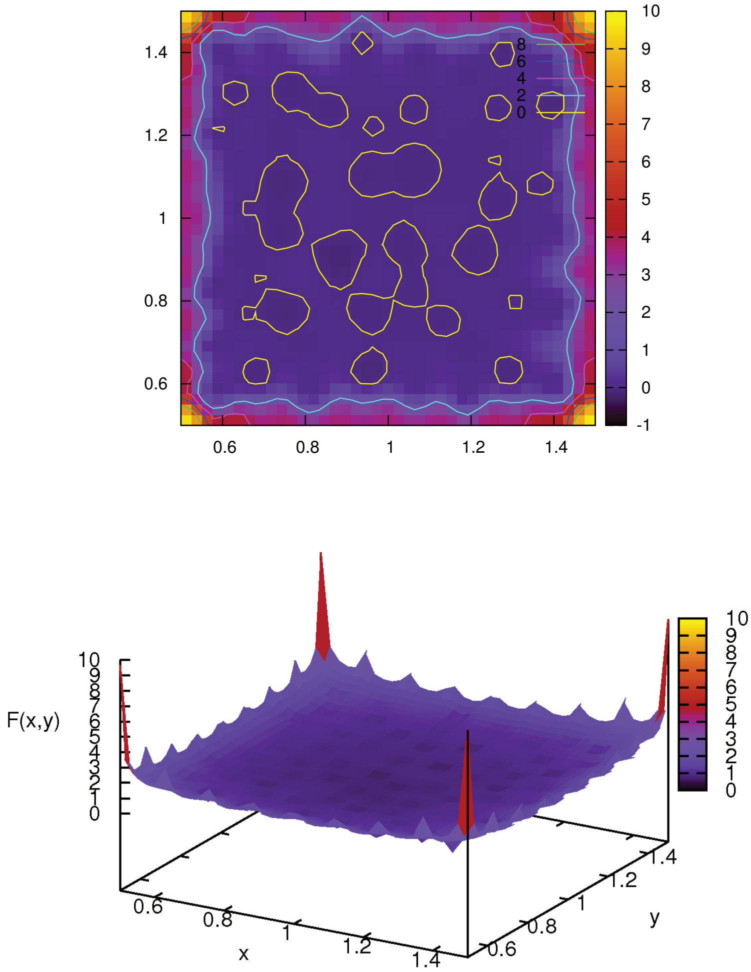

Fig. 12.24 Distribution of the control function.

12.5.1.4 Example 2

In the second example boundary conditions were altered. Instead of the boundary $\partial\Omega_1$ fixed, we fix only four corners: (0,0), (2,0), (2,2), and (0,2). First iterations improve the displacements distribution and the objective function decreases. The sufficiently good solution is depicted in the first diagram in Figure 12.25. The subdomain Ω_2 is flat, although more accurate solution is expected. Then the Ipopt procedure starts to diverge. Successive plots in Figure 12.25 show that the final form is far from perfect. Unfortunately, Ipopt announces that the optimal solution is found and the objective function is two or four ranges higher than at the beginning.

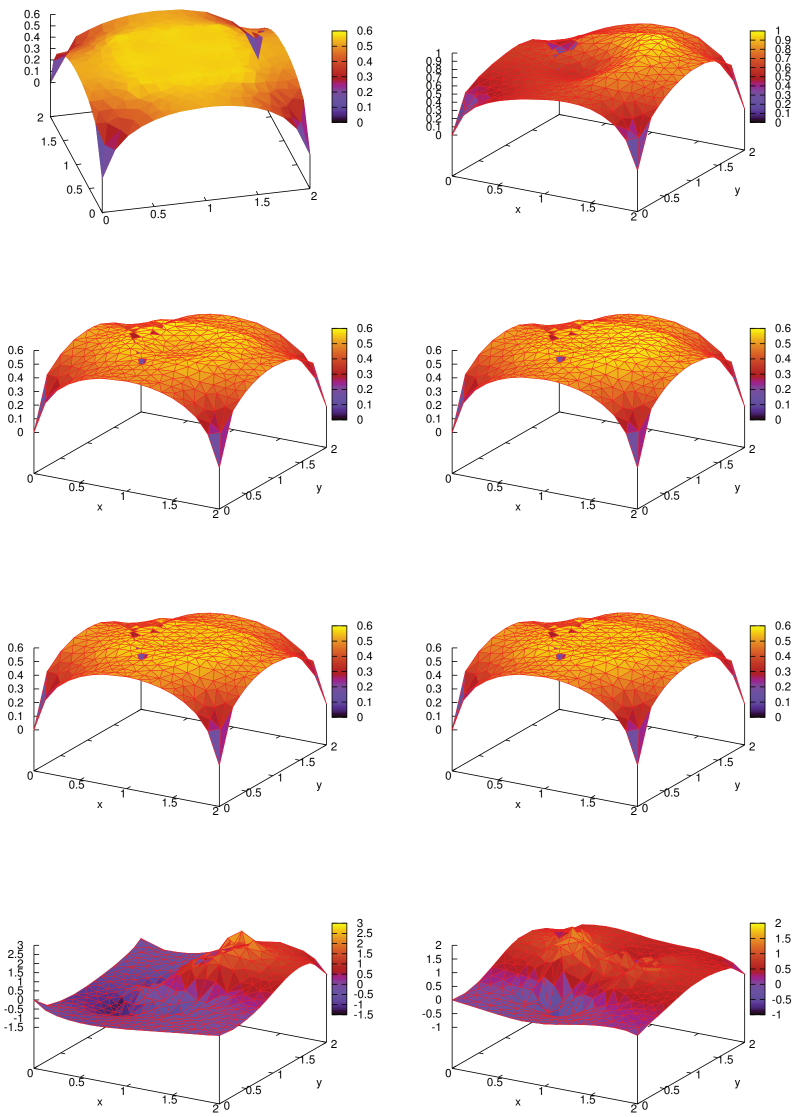

Fig. 12.25 Poor convergence of the control problem.

Table 12.3 Objective function in dependence on the accuracy.

accuracy	objective function	No. of steps
1e-2	1.459972e+01	10
5e-3	2.011944e+01	32
2e-3	1.724331e+01	27
1e-3	1.818643e+01	35
5e-4	1.2364652e+03	3000 (terminated)
1e-4	2.1803947e+02	3000 (terminated)

We see that increased accuracy results in significantly longer computations and, what is a little bit confusing, the final solution much more far from the accurate results than in the case of lower accuracy (Table 12.3). In the case of low tolerance parameter the maximum number of iterations was reached and computations were terminated. This is the disadvantage of the optimization procedure Ipopt. However, in this example we have 95 controlled variables. What is more, the optimized functional is not convex. The objective function in successive iterative steps is depicted in Figure 12.26.

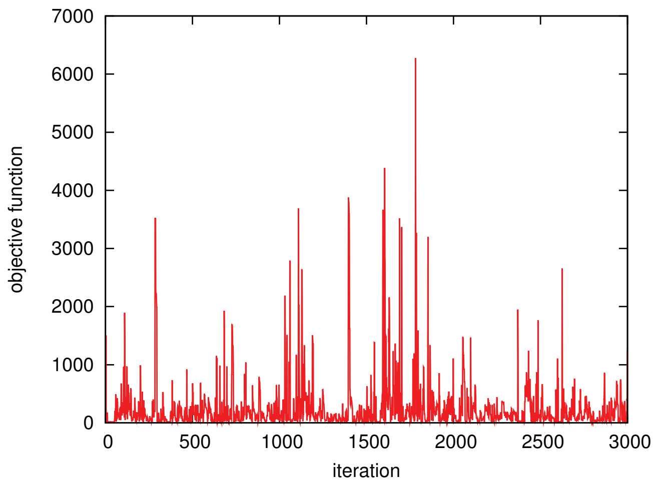

Fig. 12.26 Objective function vs. iterations.