An. Şt. Univ. Ovidius Constanţa Vol. 11(1), 2003, 179–186

A NOTE ON THE DANILOVSKAIA'S PROBLEM

Răzvan Răducanu

To Professor Silviu Sburlan, at his 60's anniversary

Abstract

The present paper describes a thermal shock problem on a semispace within the frame of linear thermoelasticity. The analytical solution is obtained and two type of a finite difference numerical algorithm to solve the problem are also described. The solutions are discussed.

1.Introduction

The present paper presents the famous problem of thermal shock on a semi space, called Danilovskaia's problem. So, we'll consider the equations of linear thermoelasticity to predict the way in which the heat will propagate through the elastic semi space. We will implement two kinds of finite difference method: an explicit one and an implicit one. The results are plotted and the differences between these two methods are discussed across the paper.

2.Basic equations

Let's consider the Cartesian frame $Ox_1x_2x_3$ the three dimensional Euclidian space. Suppose that the semi space $D = \{x = (x_1, x_2, x_3)|x_1 > 0$ is occupied by an isotropic and homogenous medium, as in the figure 1.

As in [3], the basic equations of the linear thermoelasticity are:

• the motion equations:

$$t_{ji,j} + \rho_0 f_i = \rho_0 ii_i \text{ on } D \times I \tag{2.1}$$

• the energy equation:

$$\rho_0 \theta_0 \dot{\eta} = \rho_0 s + q_{i,i} \text{ on } D \times I \tag{2.2}$$

Key Words: thermal shock, thermoelasticity, Danilovskaia's problem, numerical methods

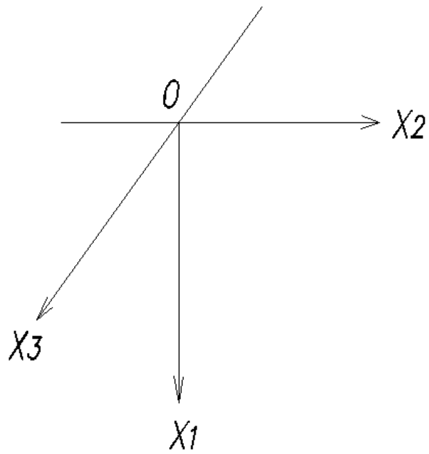

Figure 1: The frame

- the constitutive equations:

$$t_{ij} = 2\mu\varepsilon_{ij} + \lambda\varepsilon_{kk}\delta_{ij} - \beta\theta\delta_{ij} \tag{2.3}$$

$$\rho_0\eta = \beta\varepsilon_{ij} + a\theta \tag{2.4}$$

$$q_i = k\theta_i \tag{2.5}$$

- the strain-displacement relations:

$$\varepsilon_{ij} = u_{i,j} + u_{j,i} \quad on \ \ D \times I \tag{2.6}$$

where, $I = [0, \infty)$ is the time interval, u_i are the components of the displacement vector, t_{ij} are the components of the stress tensor, ε_{ij} are the components of the strain tensor, f_i are the components of the specific body force, s is the specific heat supplied, η is the entropy density on mass unit, θ is the temperature measured from a constant reference temperature θ_0,λ, μ, β, k are constants, characteristic of the material. We'll attach the following boundary conditions:

$$u_i = \overline{u_i}, \theta = \bar{\theta} \text{ on } \ \Gamma \times I \tag{2.7}$$

$$t_{ji}n_j = \bar{t}_i, q_i n_i = \bar{q} \ \ on \ \ \Gamma \times I \tag{2.8}$$

where $\bar{u}_i, \bar{t}_i, \bar{\theta}, \bar{q}$ are continuous functions given on the boundary, $\Gamma = \partial D$ and the following initial conditions:

$$u_i(x,0) = a_i(x),\ u_i(x,0) = b_i(x),\ x \in \overline{D} \tag{2.9}$$

$$\eta(x,0) = \eta_0(x),\ x \in \overline{D}, \tag{2.10}$$

where $a_i(x), b_i(x), \eta(x)$ are given continuous functions.

Thus the boundary value problem is to find $u_i(x,t), \theta(x,t)$ which satisfy (2.1)-(2.6), the boundary conditions (2.7) and (2.8) and the initial conditions (2.9) and (2.10). After some elementary computations, we obtain the following equations:

$$\mu u_{i,jj} + (\lambda + \mu)u_{j,ji} - \beta\theta_i + \rho_0 f_i = \rho_0 \ddot{u}_i \quad \text{in } D \times I \tag{2.11}$$

$$k\theta_{j,ii} - \theta_0\beta u_{r,r} - a\theta_0 \dot{\theta} = -\rho_0 s \ \text{ in } D \times I. \tag{2.12}$$

To these equations we'll attach the corresponding initial and boundary conditions, that follow immediately from (2.9) and (2.10).

In the rest of the paper, as in [3] we'll suppose that $f_i = 0,\ s = 0$, and the boundary surface $x_1 = 0$ is traction free. We will also suppose that the thermal field that acts on does not depend on position.

Thus our problem becomes: find $u_i(x,t)$ and $\theta(x,t)$ that satisfy the equations:

$$\mu u_{i,jj} + (\lambda + \mu)u_{j,ji} - \beta\theta_i = \rho_0 ii_i \text{ in } \ D \times I \tag{2.13}$$

$$k\theta_{j,ii} - \theta_0\beta u_{r,r} - a\theta_0 \dot{\theta} = 0 \text{ in } \ D \times I. \tag{2.14}$$

And the corresponding initial conditions are:

$$u_i(x,0) = 0,\ \dot{u}_i\ (x,0),\ \theta(x,0) = 0, x\ x \in D \tag{2.15}$$

and the boundary conditions are:

$$t_{ji}n_j = 0, \theta = l(t) \text{ on } x_1 = 0, t > 0, \tag{2.16}$$

$$u_i, u_{i,j}, \theta, \theta_i \to 0, x \to \infty. \tag{2.17}$$

In the following, invoking the domain symmetry we'll search for the solution of the form:

$$u_1 = u_1(x,t),\ u_2 = u_3 = 0,\ \theta = \theta(x_1,t). \tag{2.18}$$

In the following we will use the notations:

$$x = \frac{cc_1}{k}x_1, \tau = \frac{cc_1^2}{k}t, u = \frac{cc_1}{k}u_1, T = \frac{1}{T_0\theta}, \tag{2.19}$$

where

$$c_1 = \sqrt{\frac{\lambda + 2\mu}{\rho_0}}. \tag{2.20}$$

In these new variables, our problem reads: find $u(x,\tau)$ and $T(x,\tau)$ that satisfy the equations:

$$\left(\frac{\partial^2}{\partial x^2} - \frac{\partial^2}{\partial \tau^2}\right) u - A\frac{\partial T}{\partial x} = 0 \tag{2.21}$$

$$\left(\frac{\partial^2}{\partial x^2} - \frac{\partial}{\partial \tau}\right) T - \frac{\beta}{c}\frac{\partial^2 u}{\partial \tau \partial x} = 0, \tag{2.22}$$

where

$$A = \frac{\beta\theta_0}{\rho_0 c_1^2}. \tag{2.23}$$

The attached initial conditions are:

$$u(x,0) = 0, \frac{\partial u}{\partial \tau}(x,0) = 0, T(x,0) = 0, x > 0. \tag{2.24}$$

The corresponding boundary conditions are:

$$t_{11} = 0, T = f(\tau) \text{ on } x, \tau > 0, \tag{2.25}$$

$$u, \frac{\partial u}{\partial x}, T, \frac{\partial T}{\partial x} \to 0 (x \to \infty). \tag{2.26}$$

Danilovskaia studied the uncoupled case of the problem exposed here, supposing:

$$f(\tau) = T^* H(\tau), \tag{2.27}$$

where $H(\tau)$ is the Heaveside function. So, let us consider this specific case in the following. We'll take:

$$\left(\frac{\partial^2}{\partial x^2} - \frac{\partial}{\partial \tau}\right) T = 0. \tag{2.28}$$

Thus, applying the Laplace transform to the equations (2.21) and (2.22), after some computations we obtain the following analytical solution of our problem:

$$\theta(x,t) = \theta_0 \mathrm{erfc}\left\{\frac{z}{2\sqrt{\tau}}\right\}, \tag{2.29}$$

where τ is given by (2.19) and erfc is the complementary error function given by:

$$\operatorname{erfc}\{y\} = \frac{2}{\sqrt{\pi}} \int\limits_{y}^{\infty} e^{-s^2} ds. \tag{2.30}$$

In the same manner one can obtain an analytical expression for $u, x, t)$. We won't write it explicitly here.

3.Numerical implementation

In this section we will implement the finite difference method ([2], [4]) for solving our problem. First we will implement an explicit finite difference method and second an implicit finite difference method. For both these methods we will consider a spatial mesh and a mesh in time. To set these, we can write:

% L the interval length
% T the time limit
% n number of subintervals for x
% m number of subintervals for t
$h = L/n; k = T/m;$

Next, we will denote the solution at a grid point $T(x_i, t_i)$ by T_{ij}. In order to implement the explicit finite difference method, we will replace the space derivative by a finite difference formula at the $j - th$ time step and the time derivative by a forward difference formula. Thus, we can write the following system of equations for T at the grid points:

$$\frac{1}{k}\left[T_{i,j+1} - T_{i,j}\right] = \frac{1}{h^2}\left[T_{i-1,j} - 2T_{i,j} + T_{i+1,j}\right]. \tag{2.31}$$

If we employ the following notation:

$$r = \frac{k}{h^2} \tag{2.32}$$

the equation (2.31) becomes:

$$T_{i,j+1} = rT_{i-1,j} + (1-2r)T_{i,j} + rT_{i+1}, \text{ for } i = 1, ..., n-1. \tag{2.33}$$

Now the idea is to find the solution step by step, because the initial solution is known. The problem with this explicit method is to consider the meshes in such a way to ensure stability. If we'll take those meshes such as $0 < r \leq \frac{1}{2}$ the stability is ensured, as it is mentioned in []. To find the solution within this algorithm at the first time step, we should use the following MATLAB code:

$r = a * k/h\hat{\ }2; rr = 1 - 2 * r;$
$u(1,1) = r * y0 + rr * y(1) + r * y(2);$
$u(2 : n-2, 1) = r * y(1 : n-3)\prime + rr * y(2 : n-2)\prime + r * y(3 : n-1)\prime;$
$u(n-1, 1) = r * y(n-2) + rr * y(n-1) + r * L;$

For the next steps of the algorithm a similar code may be implemented. Within this explicit frame, we can draw the following surface.

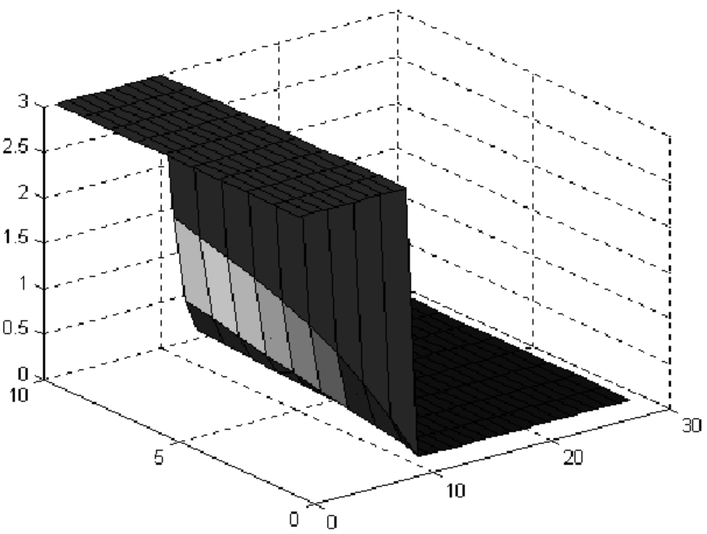

Figure 2: Explicit case

In this figure, the space is expanding from left to right and the time is progressing down the page.

In order to implement the implicit finite difference method, we will replace the space derivative by a centered difference formula at the forward time step $j+1$ and the time derivative by a forward difference formula. We can write the following system of equations for T at the grid points:

$$\frac{1}{k}\left[T_{i,j+1} - T_{i,j}\right] = \frac{1}{h^2}\left[T_{i-1,j} - 2T_{i,j} + T_{i+1,j}\right]. \tag{2.31}$$

The equation (2.31) can be written:

$$T_{i,j+1} = rT_{i-1,j} + (1-2r)T_{i,j} + rT_{i+1}, \text{ for } i = 1, ..., n-1. \tag{2.33}$$

The advantage of this algorithm consists in the fact that it is unconditionally stable. To implement it, one should consider an algorithm of the following type for the time steps 2 to m.

$\%w0t$ is the initial condition function from (2.27)

$w(1:n-1,1) = x(1:n-1)\prime$; for $j = 2:m$

$cc(1) = r * w0t(j) + w(1, j-1);$

$cc(2:n-2) = w(2:n-2, j-1);$

$cc(n-1) = r * wat(j) + w(n-1, j-1);$

$x = LU\_tridiag\_solve(aa, dd, bb, cc); w(1:n-1, j) = x(1:n-1)\prime;$

end.

In the above code the function LU_tridiag_solve solves a tridiagonal system using LU factorization. The LU_tridiag_solve function can be written:

$functionx = LU\_tridiag\_solve(a, d, b, r)$

$n = length(d);$

$z(1) = r(1);$

for $i = 2:n$

$z(i) = r(i) - b(i) * z(i-1);$

end

$x(n) = z(n)/d(n);$

for $i = n-1 : -1 : 1$

$x(i) = (z(i) - a(i) * x(i+1))/d(i);$

end

Implementing the implicit algorithm MATLAB code, one can draw the figure 3. In this figure, we used the same convention as in the previous picture.

Observing the graphical results obtained by these two kinds of finite difference method, one could notice the difference between these two methods. In a future paper, an error analysis will be made.

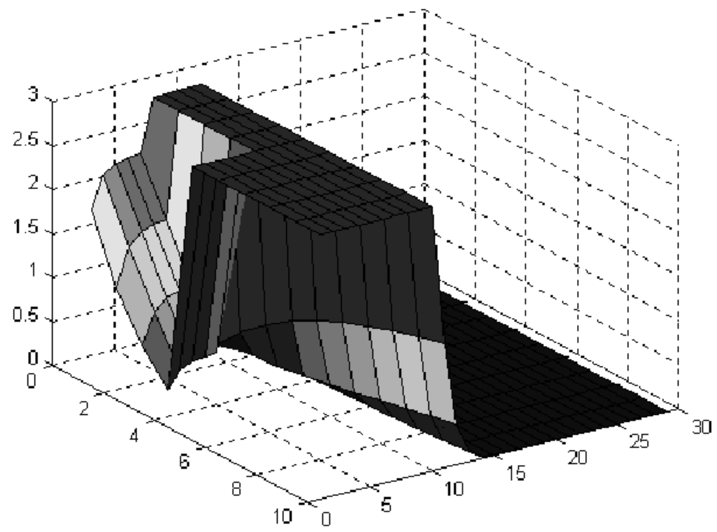

Figure 3: Implicit case

References

[1] Ghinea M., Fireteanu V., *MATLAB calcul numeric, grafica, aplicatii.*

[2] Hornbeck, R.W., *Numerical methods,* Prentice Hall, Englewood Cliffs,NJ, 1996.

[3] Iesan D., *Teoria termoelasticitatii*, Ed. Acad. R.S.R., Bucureşti, 1979.

[4] Jaeger, J.C., *An Introduction to Applied Mathematics*, Claredon Press, Oxford, U.K., 1951.

Al. I. Cuza University,
Department of Applied Mathematics,
6600- Iassy,
Romania