Inverse spline-interpolation for numerically solving a piston problem with isothermic gas compression and constant velocity

F. Criado \(^a\), T. Gachechiladze \(^b\), H. Meladze \(^b\)

\(^a\) Department of Mathematics, Malaga University, Spain
\(^b\) Department of Applied Mathematics, Tbilisi State University, Georgia

To cite this Article: Criado, F., Gachechiladze, T. and Meladze, H., 'Inverse spline-interpolation for numerically solving a piston problem with isothermic gas compression and constant velocity', International Journal of Computer Mathematics, 71:1, 83 - 95

To link to this article: DOI: 10.1080/00207169908804794
URL: http://dx.doi.org/10.1080/00207169908804794
INVERSE SPLINE-INTERPOLATION FOR NUMERICALLY SOLVING A PISTON PROBLEM WITH ISOTHERMIC GAS COMPRESSION AND CONSTANT VELOCITY

F. CRIADO a,*, T. GACHECHILADEZ b and H. MELADZE b

a Department of Mathematics, Malaga University, Spain;
b Department of Applied Mathematics, Tbilisi State University, Georgia

(Received 25 September 1997; In final form 11 November 1997)

The application of an inverse spline-interpolation method for solving non-linear algebraic equation systems, obtained when changing the differential equations of gas dynamics by the corresponding difference scheme, is considered. The iteration process constructed on account of inverse spline-interpolation with the aid of p-linear splines for solving a non-linear algebraic equation system may be considered as a generalisation of a known method used for solving equation \( F(x) = 0 \), based on the contraction principle. This process is presented in such a way as to facilitate its realization on abstract parallel processing systems (APPS).

Keywords: Splines; interpolation; computer applications to physics


* Corresponding author.
1. SOLVING A NON-LINEAR ALGEBRAIC EQUATION SYSTEM BY INVERSE SPLINE-INTERPOLATION

Let us consider the following non-linear algebraic equation system:

\[
\begin{aligned}
G_1(x_1, \ldots, x_p) &= 0, \\
G_2(x_1, \ldots, x_p) &= 0, \\
& \quad \vdots \\
G_p(x_1, \ldots, x_p) &= 0.
\end{aligned}
\]  
\tag{1.1}

Let it be known that there is one solution to system (1.1) in a \(p\)-dimensional region \(D\). Let us consider the following expressions:

\[
\begin{aligned}
\phi_1(x_1, \ldots, x_p; u_1, \ldots, u_p) &= u_1 - G_1(x_1, \ldots, x_p) = 0, \\
& \quad \vdots \\
\phi_p(x_1, \ldots, x_p; u_1, \ldots, u_p) &= u_p - G_p(x_1, \ldots, x_p) = 0.
\end{aligned}
\]  
\tag{1.2}

We assume that existing conditions for non-explicit functions are fulfilled [1] in a \(2p\)-dimensional parallelepiped with centre in \((x_1^0, \ldots, x_p^0; u_1^0, \ldots, u_p^0)\). Then relations (1.2) define \(x_1, \ldots, x_p\) as unique functions of \(u_1, \ldots, u_p\):

\[x_1 = \phi_1(u_1, \ldots, u_p), \ldots, x_p = \phi_p(u_1, \ldots, u_p).\]  
\tag{1.3}

As an approximation to an exact solution \((x_1^*, \ldots, x_p^*)\) of system (1.1), we can take \((\bar{x}_1, \ldots, \bar{x}_p)\), which are calculated as shown below:

\[
\begin{aligned}
\bar{x}_1 &= S_{\varphi_1}(0, \ldots, 0), \\
& \quad \vdots \\
\bar{x}_p &= S_{\varphi_p}(0, \ldots, 0).
\end{aligned}
\]  
\tag{1.4}

where \(S_{\varphi_1}, \ldots, S_{\varphi_p}\) are \(p\)-linear spline-interpolating values of \(\varphi_1, \ldots, \varphi_p\) functions at the knots of net \(\Delta u\), respectively. We can write the expression for a \(p\)-linear spline such as [2]:

\[
S_{\varphi}(u_1, \ldots, u_p) = \frac{1}{h_1 \ldots h_p} \left[ \varphi(u_{i_1}, \ldots, u_{i_p})(u_1 - u_{i_1 - 1}) \ldots (u_p - u_{i_p - 1}) + \right.
\]
\[
\left. + \ldots \varphi(u_{i_k}, \ldots, u_{i_k - 1}, \ldots, u_{i_p})(u_1 - u_{i_1 - 1}) \right)
\]
\[
\left. + (u_k - u) \right)
\]
\[
+ \ldots (u_p - u_{i_p - 1}) + 
\]
\[
+ \ldots \varphi(u_{i_1 - 1}, \ldots, u_{i_p - 1})(u_{i_1 - 1} - u_1) \ldots (u_{i_p} - u_p), \]  
\tag{1.5}
where \((u_1, \ldots, u_p) \in \{u_{i_1-1} \leq u_1 \leq u_{i_1}, \ldots, u_{i_p-1} \leq u_p \leq u_{i_p}\}\). From (1.5) we can see that for calculating an approximate solution \((\bar{x}_1, \ldots, \bar{x}_p)\) to system (1.1) by formulae (1.4) it is necessary to know the values of non-explicit functions \(\varphi_1, \ldots, \varphi_p\) at the fixed knots of the net. These values are calculated using the following formulae:

\[
\bar{x}_j = \varphi_j(u_1, \ldots, u_p) = \varphi_j(u_1^0, \ldots, u_p^0) - (u_k - u_k^0) J^{-1}(x_1^0, \ldots, x_p^0) \times \frac{\mathcal{D}(\phi_1, \ldots, \phi_p)}{\mathcal{D}(x_1^0, \ldots, x_p^0)} j = 1, \ldots, p
\]

where \((x_1^0, \ldots, x_p^0)\) is a given point and correspondingly

\[
u^0_1 = G_1(x_1^0, \ldots, x_p^0), \ldots, u^0_p = G_p(x_1^0, \ldots, x_p^0),
\]

\[
J(x_1, \ldots, x_p) = \frac{\mathcal{D}(\phi_1, \ldots, \phi_p)}{\mathcal{D}(x_1, \ldots, x_p)} \neq 0.
\]

The deduction of formulae (1.6) is given in Chapter 2.

Let us construct the iteration process based on the inverse spline-interpolation idea for solving system (1.1). Let us consider a point \((x_1^0, \ldots, x_p^0) \in \mathcal{D}\) and let cube \((-u_1^0 \leq u_1 \leq u_1^0, \ldots, -u_p^0 \leq u_p \leq u_p^0\) with centre in the origin of the co-ordinate system be contained in the definition region of non-explicit functions. On such a cube we construct splines \(S_{\varphi_1}^0, \ldots, S_{\varphi_p}^0\). Let us assume that

\[
\bar{x}_1^1 = S_{\varphi_1}^0(0, \ldots, 0), \ldots, \bar{x}_p^1 = S_{\varphi_p}^0(0, \ldots, 0).
\]

Correspondingly we have:

\[
u_1^1 = G_1(\bar{x}_1^1, \ldots, \bar{x}_p^1), \ldots, u_p^1 = G_p(\bar{x}_1^1, \ldots, \bar{x}_p^1).
\]

On cube \((-u_1^1 \leq u_1 \leq u_1^1, \ldots, -u_p^1 \leq u_p \leq u_p^1\) we again construct splines \(S_{\varphi_1}^1, \ldots, S_{\varphi_p}^1\) etc. At \(n\)-th step of this process we shall have

\[
\bar{x}_1^{n+1} = S_{\varphi_1}^n(0, \ldots, 0), \ldots, \bar{x}_p^{n+1} = S_{\varphi_p}^n(0, \ldots, 0). \quad n = 0, 1, \ldots
\]

Using (1.5) and (1.6) we can explicitly express this process in the following way:

\[
\bar{x}_j^{n+1} = \bar{x}_j^n + \sum_{k=1}^{p} u_k^n J^{-1}(\bar{x}_1^n, \ldots, \bar{x}_p^n) \frac{\mathcal{D}(\phi_1, \ldots, \phi_p)}{\mathcal{D}(\bar{x}_1^n, \ldots, \bar{x}_p^n)}, \quad j = 1, \ldots, p.
\]
Let us now consider the following functions:

\[ f_j(x_1, \ldots, x_p) = x_j - \lambda_j \sum_{\beta=1}^{p} \Gamma_{\alpha \beta}(x_1, \ldots, x_p) G_{\beta}(x_1, \ldots, x_p), \quad j = 1, \ldots, p \]  

(1.9)

where \( \lambda_j, j = 1, \ldots, p \) are constants and values \( \Gamma_{\alpha \beta} \) satisfy condition:

\[ \det \| \Gamma_{\alpha \beta} \| \neq 0, \quad \alpha, \beta = 1, \ldots, p \]  

(1.10)

in this case, the solution to system

\[ x_j = f_j(x_1, \ldots, x_p), \quad j = 1, \ldots, p \]  

(1.11)

is equivalent to the solution to system (1.1). In [3] it is proven that if we conveniently choose constants \( \lambda_1, \ldots, \lambda_p \) then the iteration process

\[ x_j^{n+1} = f_j(x_1^n, \ldots, x_p^n), \quad j = 1, \ldots, p \]  

(1.12)

will converge. If we apply this to (1.8) and take the following expression for \( \Gamma_{\alpha \beta} \)

\[ \Gamma_{\alpha \beta} = \frac{\partial \phi_{\alpha}}{\partial u_{\beta}} = J^{-1}(x_1, \ldots, x_p) \frac{D(\phi_1, \ldots, \phi_p)}{D(x_1, \ldots, x_{\alpha-1}, u_{\beta}, x_{\alpha+1}, \ldots, x_p)}, \]  

(1.13)

then condition (1.10) will be fulfilled because it is a consequence of the existence condition of non-explicit functions.

Hence, the iteration process for solving system (1.1) can be written in the following way:

\[ x_j^{n+1} = x_j^n + \lambda_j \sum_{k=1}^{p} u_k^n J^{-1}(x_1^n, \ldots, x_p^n) \frac{D(\phi_1, \ldots, \phi_p)}{D(x_1^n, \ldots, x_{j-1}^n, u_k^n, x_j^{n+1}, \ldots, x_p^n)}, \quad j = 1, \ldots, p. \]  

(1.14)

The solution to system (1.1) is reduced to solving the equivalent system (1.11), which in operator representation can be written as

\[ x = \mathbb{T}x \]  

(1.15)

and the corresponding iteration process (1.12) is

\[ x^{n+1} = \mathbb{T}x^n, \quad n = 0, 1, 2, \ldots \]  

(1.16)
For investigating the convergence of this process we can therefore use the $\mathcal{T}$ operator's fixed point Theorem [3] and [4].

The estimation of parameter $\lambda = \max_{1 \leq \alpha \leq p} \lambda_{\alpha}$ (see (1.14)) in [3] is obtained in the following way:

$$0 < \lambda < \frac{2}{p}$$

Likewise, it is shown that the convergence rate of process (1.14) is no greater than 2.

**2. CALCULATING NON-EXPlicit FUNCTION VALUES**

Let $x_0^0, \ldots, x_p^0 \in \mathcal{D}$ and corresponding values, in accordance with Eq. (1.1), are given in the following way:

$$x_i^0 = G_i(x_1^0, \ldots, x_p^0), \ldots, x_p^0 = G_p(x_1^0, \ldots, x_p^0).$$

Let point $(u_1^0, \ldots, u_p^0)$ belong to the existence region of non-explicit functions. In this case, we have:

$$x_1^0 = \varphi_1(u_1^0, \ldots, u_p^0), \ldots, x_p^0 = \varphi_p(u_1^0, \ldots, u_p^0).$$

Let us calculate the values of functions $\varphi_1, \ldots, \varphi_p$ at point $(u_1^0, \ldots, u_{k-1}^0, u_k, u_{k+1}^0, \ldots, u_p^0)$, which also belong to the existence region of non-explicit functions. We have:

$$x_1 = \varphi_1(u_1^0, \ldots, u_{k-1}^0, u_k, u_{k+1}^0, \ldots, u_p^0), \ldots,$$

$$x_p = \varphi_p(u_1^0, \ldots, u_{k-1}^0, u_k, u_{k+1}^0, \ldots, u_p^0).$$

Considering $x_1, \ldots, x_p$ as functions of one variable and using the finite increment formula, we can write

$$x_j = \varphi_j(u_1^0, \ldots, u_p^0) + (u_k - u_j^0) \varphi'_{j_{m_k}}(u_1^0, \ldots, u_{k-1}^0, u_k, u_{k+1}^0, \ldots, u_p^0),\quad j = 1, \ldots, p,$$

where $\bar{u}_k^0 = u_k^0 + \theta_j(u_k - u_j^0),\; |\theta_j| < 1, j = 1, \ldots, p$.

For $\varphi'_{j_{m_k}}, j = 1, \ldots, p$ let us use the following known formulae:

$$\varphi'_{j_{m_k}}(u_1, \ldots, u_p) = -J^{-1} \frac{\partial(\phi_1, \ldots, \phi_p)}{\partial(x_1, x_j, \ldots, x_{j_{m_k}}, \ldots, x_p)},\quad J = \frac{\partial(\phi_1, \ldots, \phi_p)}{\partial(x_1, \ldots, x_p)}.$$
Using these expressions in (2.1), we obtain

\[ x_j = \varphi_j(u_1^0, \ldots, u_p^0) - (u_k - u_k^0) \frac{j^{-1} \frac{D(\phi_1, \ldots, \phi_p)}{D(x_1, \ldots, x_{j-1}, \tilde{x}_2, \ldots, \tilde{x}_p)}}{D(x_1, \ldots, x_{j-1}, \tilde{x}_2, \ldots, \tilde{x}_p)}, \quad j = 1, \ldots, p \]  

(2.3)

In these formulae, points \((\tilde{x}_1, \ldots, \tilde{x}_p), j = 1, \ldots, p\) correspond to points \((u_1^0, \ldots, u_{k-1}^0, u_k^0, u_{k+1}^0, \ldots, u_p^0), j = 1, \ldots, p\). We can evidently write the following:

\[ G_k(x_1^0, \ldots, x_p^0) = u_k^0 \quad \text{and} \quad G_k(\tilde{x}_1, \ldots, \tilde{x}_p) = u_k^0, \quad k = 1, \ldots, p \]  

(2.4)

or

\[
\begin{align*}
\Delta G_1 &= 0 \\
\Delta G_k &= G_k(x_1^0, \ldots, x_p^0) - G_k(x_1^0, \ldots, x_p^0) = \frac{\partial G_k(x_1^0, \ldots, x_p^0)}{\partial x_1} \Delta x_1 + \\
&+ \ldots + \frac{\partial G_k(x_1^0, \ldots, x_p^0)}{\partial x_p} \Delta x_p + 0(\|\Delta x\|) = u_k - u_k^0, \\
\Delta G_p &= 0.
\end{align*}
\]  

(2.5)

Where \(x_1, \ldots, x_p\) are defined from formulae (2.3).

Introducing these expressions into (2.5), we can write:

\[
\begin{align*}
\frac{\partial G_k}{\partial x_1}(x_1^0, \ldots, x_p^0)(u_k - u_k^0) \frac{1}{D(x_1, \ldots, x_{j-1}, \tilde{x}_2, \ldots, \tilde{x}_p)} + \ldots + \\
&+ \frac{\partial G_k}{\partial x_p}(x_1^0, \ldots, x_p^0)(u_k - u_k^0) \frac{p}{D(x_1, \ldots, x_{j-1}, \tilde{x}_2, \ldots, \tilde{x}_p)} \\
&= -(u_k - u_k^0), \quad k = 1, \ldots, p
\end{align*}
\]  

(2.6)

Considering values

\[ \frac{j^{-1} \frac{D(\phi_1, \ldots, \phi_p)}{D(x_1, \ldots, x_{j-1}, \tilde{x}_2, \ldots, \tilde{x}_p)}}{D(x_1, \ldots, x_{j-1}, \tilde{x}_2, \ldots, \tilde{x}_p)}, \quad j = 1, \ldots, p \]

as unknown, we define them using system (2.6): the determinant of this system is not equal to zero owing to the existence assumption of non-explicit
functions. Taking into account that

\[ \frac{D(\phi_1, \ldots, \phi_{k-1}, \phi_{k+1}, \ldots, \phi_n)}{D(x_0, \ldots, x_{j-1}, x_j, x_{j+1}, \ldots, x_p)} = (-1)^{j-k} \frac{D(\phi_1, \ldots, \phi_p)}{D(x_0, \ldots, x_{j-1}, x_k, x_{j+1}, \ldots, x_p)}, \]

we can finally write

\[ \frac{j_j \frac{D(\phi_1, \ldots, \phi_p)}{D(x_0, \ldots, x_{j-1}, x_k, x_{j+1}, \ldots, x_p)} - J^{-1}(x_0, \ldots, x_p) \frac{D(\phi_1, \ldots, \phi_p)}{D(x_0, \ldots, x_{j-1}, x_k, x_{j+1}, \ldots, x_p)}}{j \frac{D(\phi_1, \ldots, \phi_p)}{D(x_0, \ldots, x_{j-1}, x_k, x_{j+1}, \ldots, x_p)}}, \quad j = 1, \ldots, p. \] (2.7)

Substituting these expressions into (2.3), we obtain

\[ x_j = \varphi_j(u_1^0, \ldots, u_p^0) - (u_k - u_k^0) J^{-1}(x_0^0, \ldots, x_p^0) \times \]

\[ \frac{D(\phi_1, \ldots, \phi_p)}{D(x_0^0, \ldots, x_{j-1}^0, u_k^0, x_{j+1}^0, \ldots, x_p^0)}, \quad j = 1, \ldots, p. \] (2.8)

### 3. PARALLEL ALGORITHM OF A NUMERICAL SOLUTION OF DIFFERENCE EQUATIONS OF GAS DYNAMICS

Let us consider a problem in which a piston is moving into gas with a constant velocity (isothermic case) [5]. The structure of the resulting shock wave in an ideal gas is described by the following differential equation system:

\[ \frac{\partial \eta}{\partial t} - \frac{\partial \nu}{\partial S} \quad \frac{\partial \nu}{\partial t} = - \frac{\partial g}{\partial S}, \quad g = \rho + \omega, \quad \omega = \frac{\nu}{\eta} \frac{\partial \nu}{\partial S}, \quad p = \frac{c^2}{\eta} \] (3.1)

Where \( c > 0 \) is a sound isothermic velocity, \( \nu \) a suitable pseudoviscosity coefficient, \( \eta \) a specific volume, \( \nu \) the movement velocity of gas particles, \( p \) pressure, \( \omega \) the linear pseudo-viscosity and \( S \) the Lagrange mass variable. The initial conditions for all unknown functions are the following:

\[ \eta(S, 0) = \eta_0, \quad \rho(S, 0) = \rho_0, \quad \nu(S, 0) = \nu_0 \] (3.2)

and the boundary condition is

\[ \nu(0, t) = \nu_0. \] (3.3)
The non-explicit completely conservative uniform difference scheme for equations (3.1) has the following form [5]:

\[ v_i = -g_i^{(c)}, \quad \eta_i = \psi_i^{(0,5)}, \quad \omega = -\frac{\nu}{\eta} v_S \quad (3.4) \]

Let us consider the case \( \sigma = 1 \). (3.4) is a non-linear algebraic equation system. Using the known designations of the difference scheme theory [5], we can write system (3.4) in the following expanded form:

\[ f_{1,i}^{j+1} = v_i^{j+1} - v_i^j + \frac{\tau c^2}{h} \left( \frac{1}{\eta_i^{j+1}} - \frac{1}{\eta_i^{j+1}} \right) - \frac{\nu \tau}{h^2} \left[ \frac{1}{\eta_i^{j+1}} (v_{i+1}^{j+1} - v_i^{j+1}) - \frac{1}{\eta_i^{j-1}} (v_{i-1}^{j+1} - v_i^{j-1}) \right] = 0, \]

\[ f_{2,i}^{j+1} = \eta_i^{j+1} - \eta_i^j - \frac{\tau}{2h} \left[ (v_{i+1}^{j+1} - v_i^{j+1}) + (v_{i-1}^{j+1} - v_i^j) \right] = 0, \]

\[ i = 1, \ldots, M, \quad j = 0, 1, \ldots, \]

where \( h \) is a net step at axis \( S \) and \( \tau \) the net step time. Conditions (3.2) and (3.3) will be written in the following way: \( \eta_i^0 = \eta_0, \quad \rho_i^0 = \rho_0, \quad v_i^0 = v_0, \quad v_i^0 = v_0 \) respectively. System (3.5) has a form (1.1). For writing it in an equivalent form (1.11), we choose values \( \Gamma_\alpha \beta \) from uncoupled system (3.5). Uncoupling is performed in the following way: substitute all the quantities with a subscript different from \( i \) and superscript \( j + 1 \) by their values from the \( j \)-th layer, i.e., we shall consider quantities \( v_i^{j+1}, \eta_i^{j+1}, i = 1, \ldots, M \) as unknowns. So, as a result of uncoupling the system (3.5) with \( 2M \) unknowns we obtain \( M \) systems, all with two unknowns. In this case, matrix \( ||\Gamma_\alpha \beta|| \) has a cell structure and non-zero cells are located on the diagonal. Therefore, \( \det ||\Gamma_\alpha \beta|| \) is equal to the product of each cell's determinants \( ||\Gamma_\alpha^{(i)}||, \quad i = 1, \ldots, M, \)

\[ \det ||\Gamma_\alpha^{(i)}|| = \left| \begin{array}{cc} \frac{\partial v_i^{j+1}}{\partial f_{1,i}^{j+1}} & \frac{\partial v_i^{j+1}}{\partial f_{2,i}^{j+1}} \\ \frac{\partial \eta_i^{j+1}}{\partial f_{1,i}^{j+1}} & \frac{\partial \eta_i^{j+1}}{\partial f_{2,i}^{j+1}} \end{array} \right|, \quad (3.6) \]
where

\[
\frac{\partial v_{i+1}^{j+1}}{\partial f_{i,j}^{j+1}} = - \left[ 1 + \frac{\nu \tau}{h^2} \left( \frac{1}{\eta_i^{j+1}} + \frac{1}{\eta_{i+1}^{j+1}} \right) + \frac{\tau^2}{2h^2} \frac{1}{\eta_i^{j+1}} \left( c^2 + \frac{\nu}{h} (v_i^{j+1} - v_{i+1}^{j+1}) \right) \right]^{-1},
\]

\[
\frac{\partial v_{i+1}^{j+1}}{\partial f_{i,j}^{j+1}} = - \frac{\tau}{2h} \frac{\partial v_{i+1}^{j+1}}{\partial f_{i,j}^{j+1}},
\]

\[
\frac{\partial \eta_{i+1}^{j+1}}{\partial f_{i,j}^{j+1}} = \frac{\tau}{h} \left( c^2 + \frac{\nu}{h} (v_i^{j+1} - v_{i+1}^{j+1}) \right) \frac{\partial v_{i+1}^{j+1}}{\partial f_{i,j}^{j+1}},
\]

\[
\frac{\partial \eta_{i+1}^{j+1}}{\partial f_{i,j}^{j+1}} = \left( 1 + \frac{\nu \tau}{h^2} \left( \frac{1}{\eta_i^{j+1}} + \frac{1}{\eta_{i+1}^{j+1}} \right) \right) \frac{\partial v_{i+1}^{j+1}}{\partial f_{i,j}^{j+1}}.
\]

(3.7)

As \( \det \| \Gamma(0, \alpha, \beta) \| \neq 0 \) in the definition region of non-explicit functions, then \( \det \| \Gamma \alpha, \beta \| \) will also not be equal to zero. We can therefore use the general formulae (1.14). An iteration process based on these formulae for solving system (3.5) will have the following form:

\[
\begin{align*}
\eta_{i+1}^{j+1} & = \eta_{i+1}^{j+1} + \lambda_2 \times \\
& \frac{\tau}{(\eta_{i+1}^{j+1} + 1/\eta_{i+1}^{j+1})} + \frac{\tau^2}{2h^2} \frac{1}{\eta_i^{j+1}} \left( c^2 + \frac{\nu}{h} (v_i^{j+1} - v_{i+1}^{j+1}) \right) \]
\end{align*}
\]

\[
\begin{align*}
\eta_{i+1}^{j+1} & = \eta_{i+1}^{j+1} + \lambda_2 \times \\
& \frac{\tau}{(\eta_{i+1}^{j+1} + 1/\eta_{i+1}^{j+1})} + \frac{\tau^2}{2h^2} \frac{1}{\eta_i^{j+1}} \left( c^2 + \frac{\nu}{h} (v_i^{j+1} - v_{i+1}^{j+1}) \right) \]
\end{align*}
\]

\[
\begin{align*}
\eta_{i+1}^{j+1} & = \eta_{i+1}^{j+1} + \lambda_2 \times \\
& \frac{\tau}{(\eta_{i+1}^{j+1} + 1/\eta_{i+1}^{j+1})} + \frac{\tau^2}{2h^2} \frac{1}{\eta_i^{j+1}} \left( c^2 + \frac{\nu}{h} (v_i^{j+1} - v_{i+1}^{j+1}) \right) \]
\end{align*}
\]

(3.8)

And for \( i = 1 \) we shall have:

\[
\begin{align*}
\frac{\eta_{i+1}^{j+1}}{\eta_{i+1}^{j+1}} & = \eta_{i+1}^{j+1} + \lambda_1 \times \\
& \frac{\tau}{(\eta_{i+1}^{j+1} + 1/\eta_{i+1}^{j+1})} + \frac{\tau^2}{2h^2} \frac{1}{\eta_i^{j+1}} \left( c^2 + \frac{\nu}{h} (v_i^{j+1} - v_{i+1}^{j+1}) \right) \]
\end{align*}
\]

(3.8)
This process is presented in a form which is directly applicable in parallel processing.

Iterations were performed for each time layer: we took values of net functions from the previous layer as the initial approximation when passing to the next layer. The test calculations were performed on the condition that the gas finite mass \( 0 \leq S \leq M \) was bounded from the left side by a piston moving at a constant velocity.

The initial state of the gas was

\[ \eta(S, 0) = \eta_0 = 1, \quad p(S, 0) = p_0 = 0,25, \quad \nu(S, 0) = \nu_0 = 0 \]

The value of a sound isothermic velocity was \( c = 0,75 \).

Let us finally deduce the maximal values for \( X_1 \) and \( X_2 \) for which the iteration process is convergent. For these values, the number of iterations necessary to attain a given precision is minimal.

For \( \tau = 0,001 \) and \( \tau = 0,01 \): \( \lambda_1 = \lambda_2 = 1 \); for \( \tau = 0,25 \): \( \lambda_1 = \lambda_2 = 0,95 \); and for \( \tau = 0,4 \): \( \lambda_1 = \lambda_2 = 0,45 \).

Let us consider the realisation of the iteration process (3.8), (3.9) on abstracta parallel processing system (APPs) [7].

In our case, the parallel calculation of values \( \eta_{i,j}^{(n)} \) and \( \nu_i^{(n)} \), \( i = 1, \ldots, (M - 1) \) on piston co-ordinate knots is performed. Hence, the number of parallel processors (pp) necessary will be \( P = M - 1 \) (i.e., we calculate the
values of the two aforementioned quantities on each pp). According to formulae (3.8), (3.9), the number of binary operations necessary for calculating unknown quantities on one pp will be $m = 44 + 46 = 90$. 
Thus, the computation time on a sequential computer will be

\[ t_1 = 90 \cdot (M - 1) \quad (3.10) \]

and we estimate the computation time on APPS by the Munro and
Peterson's Theorem [7, 8], such as
\[
I_p \geq \begin{cases} 
\left[ \log_2 (M - 1) \right] + \left[ \frac{91 - 2^{\log_2 (M - 1)}}{M - 1} \right], & 2^{\log_2 (M - 1)} \leq 90, \\
7, & \text{in the other cases,}
\end{cases}
\]
where \([x]\) denotes the smallest integer greater or equal to \(x\).

The estimation, based on formulae (3.10), (3.11), of the highperformance rising coefficient \(\sigma\) [7] is the following:
\[
\sigma = \frac{I_1}{I_p} \geq \begin{cases} 
\frac{90(M - 1)}{\left[ \log_2 (M - 1) \right] + \left[ \frac{91 - 2^{\log_2 (M - 1)}}{M - 1} \right]}, & 2^{\log_2 (M - 1)} \leq 90, \\
12, & \text{in the other cases.}
\end{cases}
\]

References