SESSION F

Wednesday, October 21, 1992

1:00 p.m.

Distributed Computing

Papers:

1. The Coming Era of Full Spectrum Computing
   - Joe Requa (LLNL)

2. Heterogeneous Computing Environments
   - Ray Cline (SNL)

3. Research in Models for Massively Parallel Computing
   - Norman Morse (LANL)

4. Multi-Computer Efficiency Estimation on Evolutionary Problems in Computational Physics
   - Ivan Sofronov (Arzamas)

5. Focus of Study of the Mathematical Department
   - Valentin Kuropatenko (Chelyabinsk)

SESSION F

Distributed Computing

The Coming Era of Full Spectrum Computing

Joseph E. Requa

October, 1992

Abstract

The creation of the Personal Computer (PC) and Workstation market sparked a revolution in computing technology. Profits from mass marketing of PCs and workstations provided unprecedented capital for research and development in the computer field. That capital financed a radical improvement in computer and communication technology. Processors, memories, storage devices and communications facilities have all seen dramatically enhanced performance at reduced cost. Massively Parallel Processing (MPP) systems, built of these new commodity components, show great promise for the future. Hardware advances have been matched by an explosion in the quality, quantity and capability of software. The need for interoperability in networked environments has forced the computing industry to standardize on UNIX™ and open software to provide compatibility and portability of software. New software technologies provide for distributed computing, utilizing a heterogeneous set of resources to solve a single problem. These changes will usher in a new era of computing, based on a paradigm shift from a network computing environment to a Full Spectrum Computing (FSC) environment.

By the author's reckoning, The FSC era will be the fourth era of supercomputing. This talk will briefly examine the two past eras and the current era to illustrate the nature of technology changes required to cause a change in computing eras. The current set of technology changes, which presage a change of computing eras, will be summarized. The nature of those changes, combined with the needs of computing customers will be used to project the nature of computing in the FSC era. This nature will be illustrated by developing an architectural model for the FSC computer center of the future. The model consists of a network of distributed heterogeneous resources with a coordinating set of software services to provide virtually unlimited computing power to the end user. The computing center of the future will provide access to a wide variety of computational resources in a cost effective, responsive and transparent manner. Computing will be done in an environment of plentiful resources rather than in the current environment of scarce resources.

This work was performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under contract No. W-7405-Eng-48.

SESSION F

Distributed Computing

Heterogeneous Computing Environments*

R. E. Cline, Jr. and R. E. Palmer
Sandia National Laboratories

In simplest terms, distributed computing is the application of a wide variety of resources to solve a given problem. Resources may include workstations, vector and parallel supercomputers, storage systems, and special purpose computers such as visualization servers. An heterogeneous distributed computing environment can offer benefits ranging from new capabilities (through specialized machines and software) and expanded capacity (through the opportunistic use of spare compute cycles), to improved convenience and productivity (through the use of software tools that provide value to the end user). Distributed computing is well suited for the emerging technologies of collaborative engineering and agile manufacturing. In these applications distributed simulation tools, engineering databases, CAD facilities, and production facilities are coupled together to optimize the design, analysis, and manufacturing cycle. This application extends the definition of distributed computing to include special purpose computers that perform process control or monitoring.

The heterogeneous distributed computing approach can best be seen as a "processor view of the world." The complete distributed system is composed of all available processors, whether they reside in massively parallel processors, multiple CPU supercomputers, or a heterogeneous set of networked workstations. These systems require data translation into platform independent representations as part of the communication process. This approach permits development of applications that exploit the full spectrum of available computational resources. The capabilities of heterogeneous environments cannot be separated from the characteristics of the network that connects the individual components. In this presentation we will review current projects that are aimed at developing heterogeneous networking and computing environments for scientific and engineering applications.

---

* Work supported by United States Department of Energy contract #DE-AC04-76DP00789

SESSION F

Distributed Computing

RESEARCH IN MODELS FOR MASSIVELY PARALLEL COMPUTING

Norman R. Morse

Technical Staff Member
Computer Network Engineering
Los Alamos National Laboratory

Supercomputing in the DoE has been traditionally performed on von Neumann vector pipeline machines. In the past few years, a belief has developed that advances in supercomputing capability in the future must come from massively parallel architectures. When considering such architectures, many questions remain which are active areas of research. Examples of research subjects for massively parallel computing paradigms are communication structures, memory structures, node processor architectures, latency hiding, algorithm development, switching schemes, communication bandwidth and support software. A brief discussion of the Los Alamos Clustered Workstation Project will be given including proposed solutions to the problems revealed by the massively parallel research effort.

A problem which has existed in computing since networks were invented is that the quality of service received by those using a computing resource is inversely proportional to the user's physical distance from the machine. The goal of the Casa Gigabit project is to remove this restriction by inventing gigabit speed networks with support protocols, switches, etc. which will operate at global distances. The Los Alamos effort directed at this goal will be described.

SESSION F

Distributed Computing

Multicomputer Efficiency Estimation on Evlutionary Problems in Computational Physics

Sofronov I.D.

The simolest model is examined for a multicomputer, that is a machine containing a great number of shared-memory processors.

A simple discretization is given for differential equations describing evolutionary problems of computational physics.

Utilization efficiency is estimated for multicomputers having various switch architectures. For irregular grid, the crossbar switch multiprocessor has a higher performance if the number of PEs is smaller than that of comoutational points. As the number of procesors increases, the hypercube multiprocessor turns out to we the most efficient.

MULTICOMPUTER EFFICIENCY ESTIMATION ON EVOLUTIONARY PROBLEMS IN COMPUTATIONAL PHYSICS

Sofronov I.D., Sofronova O.I. VNIIEF, 37 Mir Avenue. Arzamas-16, Nizhny Novgorod Region.

1. CLASS OF PROBLEMS

For current computers with a large number of processing elements, the highest performance is achieved on a relatively narrow problem class which rapidly decreases as this class expands. This circumstance is a reason for considering them to be more tailored for their specific class of problems compared to conventional single-processor computers.
Recently this has resulted in many papers devoted to efficiency analysis of complex architecture computers on various type problems using a broad range of algorithms (see, for example, [1]-[5]). This paper considers well-known enough problems which require the highest performance machines to be solved. We mean time-dependent problems in continuum mechanics, high temperature and pressure physics, neutron physics and those describing the propagation of various radiation types.

For these problems, the most general governing system of

equations has the following form:

$$A\frac{\partial}{\partial t}U + B\frac{\partial}{\partial w} + CU + \mathcal{D} = \int K(U(w)U(w))dw' \quad (1)$$

where Ω is a ρ-hypercube in which a given system is defined. U, D, A, B and C are the vectors of unknown functions and coefficients, K(U(w), U(w)) are the matrices of integral operation kernels. The dimension of these quantities is P. We assume that initial and boundary conditions are correctly set on the hypercube faces. The greatest fraction of computational time is required to solve a system of differential equations which is a partial case of (1)

$$A\frac{\partial}{\partial t}U + B\frac{\partial}{\partial w}U + CU + \mathcal{D} = 0 \quad (2)$$

In real applications, the equation coefficients are functions of coordinates, time and unknown quantities, U. We take a limited model case where all the coefficients are constant. In addition to (2) consider another partial case of (1), that is a system of integral equations of the form

$$A\frac{\partial}{\partial t}U + CU + \mathcal{D} = \int_{\Omega} K(U(w), U(w'))dw' \quad (3)$$

For a hypercube, Ω, consider randomly located N points. We assign numbers to each of them and examine an i-numbered point.

We call its nearest (or first order) neighbors those points which are more closely located to it compared to others. For example, it is convenient to take as nearest neighbors those points which have Dirichlet regions with the same boundary [2]. We call the neighbors of the nearest ones second order neighbors etc. Obviously, searching nearest neighbors in the case of absolutely unordered points may result in an overall sampling. To avoid this we provide a special table of nearest neighbors for each point which would include the number or other addresses of all the nearest neighbors.

If the points are assumed to move continuously during the calculations, then we can say that neighborhood order changes may not exceed unity for two close enough times.

This means that searching nearest neighbors in a new layer should be performed by examining first and second order neighbors in an old layer.

With the point ordering established searching neighbors at a new time is considerably simplified, if close times are considered. The point set examined has not a single algorithm for choosing neighbors. This algorithm differs for each point and depends on neighbor table which changes with respect to point locations. Such point set is frequently referred to as "irregular mesh".

As opposed to above considerations, we mean by "irregular mesh" a set of points which have a multidimensional array structure where, as we know, shift operators act over each direction at any internal point.

In a regular mesh, the overall set of N points may be thought of as a product of one-dimentional meshes with point number $n-N^{1/\rho}$. Every point of a regular mesh can be exactly mapped onto points having integer coordinates and belonging to the ρ-hypercube with an edge n. The meshes introduced will be further denoted by Ωh.

The simplest explicit discretization of (2) on Ωh is a system of algebraic equations of the form:

$$U_i^{n+1} + \sum_{k=1}^{R} P_{ik}\, U_{i+k}^{n} = f_i\,, \qquad i = \overline{1, N}, \tag{4}$$

where P_{ik} values are calculated from coefficients of (2) and point parameters of Ωh. These coefficients have nonzero values only for points which are either closely located to a point considered, or are its neighbors of several first orders.

In such cases the (P_{ik}) matrix becomes considerabl, incomplete for large values of N. We assume that $U_{N+k}^{i} = U_k^{i}$ here.

For a regular mesh, the equation (4) takes a more commonly used form of a difference equation: for example, with $\rho = 3$ one obtains

$$U_{ikl}^{n+1} + \alpha^{\circ} U_{ikl}^{n} + \alpha' U_{i+1kl}^{n} + \alpha'' U_{i-1kl}^{n} + \beta' U_{ik+1l}^{n} + \beta'' U_{ik-1l}^{n} + e' U_{ikl+1}^{n} + e'' U_{ikl-1}^{n} = f_{ikl} \tag{5}$$

$$i, k, l = \overline{1, n}.$$

Equations (4) and (5) are complemented with correctly set initial data ($U^{0}_{i\kappa\ell}$) and boundary conditions having the same form as the equations but with less addends. Solving the problem consists in sequential calculation of U^{n}_{i} unknowns over time layers; first U^{1}_{i} is calculated from U^{0}_{i}, then U^{2}_{i} and we proceed up to U^{n}_{i} where $i=\overline{1, N}$.

Our considerations are restricted to an explicit discretization of (2); in the remainder of the some remarks on generalization feasibility will be made. It is seen from (4) and (5) that they allow the most extensive parallel calculations up to N branches at each timestep. This makes them well-suited to multiprocessors with shared main memory.

The simplest explicit discretization of the integral equation (3) is the equation of the form:

$$U^{n+1}_{i} + d_{i} = \sum_{\kappa=1}^{N} q_{j\kappa}\, U^{n}_{j+\kappa}, \qquad i = \overline{1, N}, \quad U^{n}_{j-N} = U^{n}_{j} \tag{6}$$

This differs mainly from (4) in that summing is performed over all the N points of Ω_h. In contrast, only relatival, close points were added in (4). This fact results particularly in disappearance of differences between regular and irregular meshes in the case of integral equation during discretization.

It should be noted that we do not specify as separate classes the problems for time-independent equations, that is those of (2) and (3) type, where the following condition is satisfied:

$$A = 0 \tag{7}$$

since the algorithms for solving them with selection method do not differ from that for solving problems under consideration. Yet, studying the features of algorithms intended to obtain solutions of time-independent problems is beyond the scope of the present paper.

Currently, problems with millions of points ($N \sim 10^6$) become common. The number of unknowns varies from tens to hundreds and even to thousands in some cases. Such problems often require 10^{14} -10^{16} arithmetic operations. A performance of billions or tens of billions of arithmetic operations per second is required to obtain a solution by running a computer during several hours. Now such performance is not available in general-purpose single processor machines. A desired performance is achieved on multiprocessors. However replacing one powerful processor by a small or large number of relatively low-level performance processors and splitting a single memory of high enough capacity into many local processor memories result in some difficulties involved in method and program developments. Not all algorithms are found to allow a high-level parallelization and for some classes of problems implementing parallel calculations is very difficult.

Recently, many papers on parallel calculations has appeared. the class of efficient parallel implementations is continuously expanding primarily, due to new algorithms which prove to be not the most effective on single processor machines. However the need for parallel implementations

results in specific computer system structure requirements. The aim of this paper is an attempt to provide simulated efficiency estimates of using a computer system including a host and multiprocessors with two different switching network architectures.

2. COMPUTER SYSTEM MODEL

Currently, there is a great number of various computer systems: a single processor, a multiprocessor with common or shared memory etc. [3]. As it was noted above, for mathematicians it is more convenient to have one powerful enough CPU and a single sufficiently large main memory, than many low-level performance processors with a memory splitted into small local pieces. However the inability to achieve a desired performance on a single processor and its extremely high cost encourage one to find the ways of using nonconventional system structures to solve various problems. Two classes are naturally distinguished among multiprocessors: systems comprising a relatively small number of highly powerful processors communicating with a shared memory and those including a very great number of relatively low performance processors. In the latter case splitting the main memory into local units for each processor proves to be reasonable, but those cannot have a very large

capacity. Such systems are sometimes called "multicomputers".

In multicomputers, processing element switching network is an essential component. Many various switching system architectures were proposed, but still no unique general-purpose architecture was found which would be efficient on a broad class of applications. Consider a computer system including a host and a multiprocessor with M processing elements which is connected to the host by a link (see Figure 1). The algorithm for solving problems of (3) or (5) types using this system is assumed to consist in the following. The portions of the algorithms which are not or poorly suited to parallelization are run on the host. The portions well suited to parallelization, that is which do not result in nonefficient hardware usage when parallelized are implemented on a multicomputer. The question that we are primarily interested in is: which of two switching archtectures (array or hypercube, see figuers 2 and 3) is more preferable for solving problems of type (3)-(5)? We introduce some notations. Let F and U be the CPU performance and the host storage capacity, respectively, while f and u denote the similar characteristics of each processing element in the multicomputer. We further assume that the switching network performance is such that a time τ is required for each processing element to send one full-digit number to all its nearest neighbors. Of course, we mean by nearest processing elements those PEs which have direct electrical

connection with a considered one. As no connections other than those between nearest PEs, are present in switching networks under consideration, data transfers between remote nodes may be performed only via transfer strings between the nearest adjacent nodes. Fore example, 2n-1 transfers need to be performed to send data from lower left corner point to the right upper one in a two-dimensional array.

The distance between these points is said to be l=2n-1.

It is easy to verify that the average distance between the corner node and any other point is

$$\ell_n^M(M) = \sqrt{M} - 1 \qquad (8)$$

in an array architecture.

For an $m=\sqrt{M}$ array, more complex considerations may be required to show that the distance between a node with coordinates k and e and any other point is

$$\ell_{ke}^M(M) = \frac{1}{m}\left[\frac{(m-\kappa)^2 + (m-e)^2 + \kappa^2 + e^2}{2} + m - \kappa - e\right] \qquad (9)$$

In a hypecube, each node is at the vertex; so we need only to obtain a formula similar to (8) here:

$$\ell_M^H(M) = \frac{1}{2}\log_2 M \qquad (10)$$

Some more conclusions are worth to be made. For any M, a two-dimensional array architecture requires that each PE should have only four connections with its neighbors. If $\mathcal{E}$

hardware is assumed to be needed to implement one connection, then

$$B^{M}(M) = 4\mathcal{E}M \qquad (11)$$

hardware will be required to implement the overall network. It follows from (9) and (10), that increasing the processor number, M, results in growth of the average distance between PEs which leads to higher loads of switching hardware when data are transferred between remote PEs. Equations (11) and (12) also show that the overall hardware grows with increasing M; note that the growth rate in a hypercube is higher than that in an array though in the former case the average distance grows less rapidly. If each PE is assumed to contain δ "arithmetic" hardware, then M δ "arithmetic" hardware will be required for the whole multicomputer. The relationship between the overall hardware and the "arithmetic" one which defines the cost of an operation in an array multicomputer is

$$\rho^{M}(M) = \frac{\mathcal{E}}{\delta} + 1 \qquad (13)$$

while we have

$$\rho^{H}(M) = \frac{\mathcal{E}}{\delta}\log_2 M + 1$$

for a hypercube architecture.

This shows the cost of an arithmetic operation to grow with

increasing M in a hypercube multicomputer, so the switching hardware will dominate for a large enough M. This must, of course, ~~be~~ inhibit unnecessary growth of PE number. Finally, we shall describe in some words the communication channel between the host and the multicomputer. We assume that only one such channel is available which is connected to a corner array point or to a hypercube vertex. In fact, many channels may exist which are connected to different multicomputer nodes to achieve their highest throughput. Suppose that τ_o seconds are required for the channel to transfer one number.

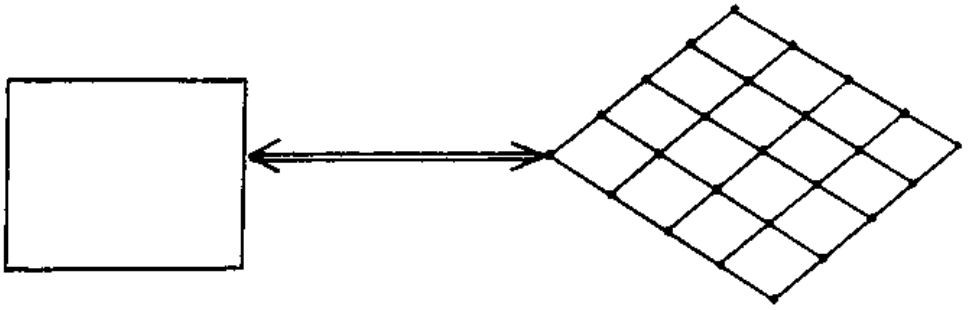


Figure 1

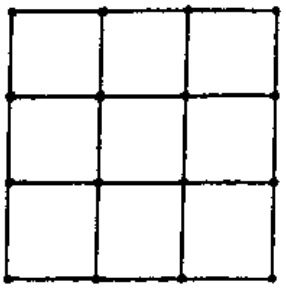


Figure 2. An array, M=16

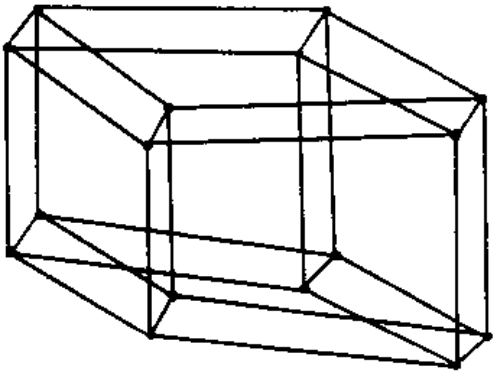


Figure 3. A hypercube, M=16.

3.SOLVING A DIFFERENTIAL EQAUTION

We start the simulation of solving a differential equation on a multicomputer by considering a partial case which is reported in [3]. We examine a multicomputer containing $M=12^4$ PEs designed for solving three-dimentional problems of time-dependent aerodynamic application type (ρ =3) where the switching network has the form of a two-dimensional array (λ =2). The differential equation is assumed to be approximated by the differential equation (4) on a regular mesh Ωh . ($N=M^3=12^{21}$). The multicomputer switching array architecture makes one to think it is convenient to establish an exact correspondance between the processor array and a layer of the mesh Ωh where the solution to be found is defined U^n_{ike} , i,k=1, $\bar{m}$, l=const.

Each processor has to solve a "one-dimensional" problem, that is to calculate U^{n+1}_{ike}, l=1, $\bar{m}$, i k=const. Data on U^n_{ike} , l=1, $\bar{n}$ is loaded into the local memory of each (i, k) processor. According to (4), the solution values at neighbouring points U^n_{i+1ke} , U^n_{i-1ke} , U^n_{ik+1e} , and U^n_{ik-1e} should be contained in local memory of an (i, k) PE for U^{n+1}_{ike} l=1, $\bar{n}$ to be computed. Other values of U^n_{ike} , U^n_{ine+1} , and U^n_{ine-1} at the (i, k) column are already put into its local memory. If boundary effects are

neglected, then $4N$ numbers should be transferred from the local memories of M PEs to those of neighbouring PEs to compute U_{ike}^{n+1}. In a general case where $M=n^{\lambda}$ processors are used, ρ-dimension problem is considered instead of a three-dimensional one, and each processor solves a "$(\rho - \lambda)$- dimension" problem rather than a "one-dimensional" problem

$$\Pi_{\rho}^{M} = 2\rho N = 2\rho M^{\frac{\rho}{\lambda}} \qquad (15)$$

transfers should be required at each timestep n+1 which takes

$$T_{\rho}^{M} = 2\tau\lambda\rho M^{\frac{\rho}{\lambda}-1} \quad sec \qquad (16)$$

if all switching links achieve their highest performance. Arithmetic computation amount to be done at each n+1 timestep will be

$$A = C N \text{ arithmetic operations} \qquad (17)$$

When a full load for all processors is achieved, the time required will be

$$T_{A} = \frac{c}{f} M^{\frac{\rho}{\lambda}-1} \quad sec \qquad (18)$$

It is found from (16) and (13) that time relationship

$$\alpha_{\rho}^{M}(M) = \frac{T_{A}}{T_{\rho}^{M}} = \frac{c}{2f\tau\lambda\rho} \qquad (19)$$

does not depend on the processor number.

Therefore we can use:

Assertion 1. For an unlimited growth of the PE number M in

multicomputer with array switching architecture, a linear increase in performance is possible when a differential equation is explicitely solved on a regular mesh.

If the same problem is run on a multicomruter with a hypercube network, it is more convenient to split the point set Ωh using another method, that is by dividing Ωh into M equal arrays which represent simple ρ -hypercubes each containing $\frac{N}{M}$ points.
As opposed to the above mentioned case, all the points of a simple hypercube can be divided into two categories: inner and boundary ones. We mean by inner points those nodes which do not require data from neighbouring PEs to be computed. Boundary points are all remaining points, that is those which have not all their nearest neighbors located in the local memory of the processor under consideration. Probably, adjacent PEs should be accessed to compute these ones. The edge of a simple hypercube introduced has the length of $n^{1-\lambda/\rho}$ points. For a hypercube, a sum of face areas is

$$S = 2\rho\, n^{(1-\lambda/\rho)(\rho-1)}$$

Therefore the whole hypercube Ωh will have an average number of $\tilde{N} = 2\rho\, n^{\rho-1+\lambda/\rho}$ boundary points, that is

$$\Pi_\rho^M = 2\rho\rho M^{\frac{\rho}{\lambda}-\frac{1}{\lambda}+\frac{1}{\rho}} \qquad (20)$$

transfers should be executed between neighbouring PEs at each timestep. Assuming that all the transfers can be uniformly distributed among the switching nodes and their highest

performance can be explcited, the time required will be

$$T_p^H = 2p\rho\tau \frac{M^{\frac{\rho-1}{\lambda}-\frac{1}{\rho}+1}}{\log_2 M}$$

Relations (21) and (10) show that the growth of $\frac{T_A}{T_p^H}$ which is

$$\alpha_p^M(M) = \frac{c}{2p\rho f\tau} M^{\frac{1}{\lambda}-\frac{1}{\rho}+1} \log_2 M \underset{M\to\infty}{\longrightarrow} \infty$$

becomes unlimited with increasing M, that is the minimum time required to perform transfers is an infinitesimal value for $n \to \infty$ compared to the time needed for the arithmetic computation to be done by all processors.

Assertion 2. For an unlimited growth of the processor number M in a multicomputer with a hypercube switching architecture, a linear increase in performance is possible when a differential equation is explicitely solved on a regular mesh, but the switching network performance may be low for $M \to \infty$. Comparing the time needed for performing transfers in an arra, multicomputer with that of a hypercube network shows that transfers are considerably less time-consuming for larg M's in the latter case:

$$\frac{T^M}{T^H} = \frac{T_p^M}{T_p^H} = M^{\frac{1}{\lambda}-\frac{1}{\rho}} \log_2 M \underset{M\to\infty}{\longrightarrow} \infty$$

that is a hypercube has an advantage with respect to this parameter. Particularly, for assumptions made in [3], this advantage gives a multifold decrease in time required for

performing transfers that is the concepts of the switching architecture reported in [3] cannot be considered as optimal ones.

Now turn to an irregular mesh. Let $N = n^{\rho}$ points are randomly located in a hypercube Ω. Distribute them among M PEs close, if possible, to those which were considered above for the case of a regular mesh. In the former example each processor solved a($\rho - \lambda$)-dimension problem consisting of points with constant first $\rho - \lambda$ coordinates. Construct similar regions on an irregular mesh. Proceeding in ascending order of the first coordinate, divide the set Ω into n portions each containing a $n^{\rho-1}$ point. The next stage will further divide each of the resulting subsets into n parts with an equal number of points, proceeding again in the ascending order of the second coordinate. Each (i, k) subset is assigned to an associated PE. Our implementation will give the same result which was already obtained.

Hence, we do not expect the number of data transfers to be lower than that of (15). In an irregular case, V will be considered as a mean number of i point neighbors participating in (3). For a mean distance between an i point and the remaining points in the array architecture given by (9), the computation of U_i^{n+1} will require an average number of $V \cdot M^{1/2}$ transfers performed by neighbouring PEs.

The neighbouring PEs would perform an average number of

$$\Pi_H^M(M) = V \cdot \rho \cdot M^{\frac{\rho}{\lambda} + \frac{1}{2}} \qquad (24)$$

transfers to compute all $U_i^{n+1}{}_F$ values, where i = 1, $\bar{N}$. For the highest switching network performace, these will require the time

$$T_H^M(M) = \frac{\tau \cdot \nu\rho\, M^{\frac{\rho}{\lambda}-\frac{1}{2}}}{4M} \simeq \tau\frac{\rho\nu}{4} M^{\frac{\rho}{\lambda}-\frac{1}{2}} \qquad (25)$$

Consider another way of splitting Ωh . Select $n^{\rho/\lambda}$ points with the lowest first coordinate values from N. Repeating this procedure λ times, we perform a partial ordering of points over the first coordinate. Once this is done, a similar partial ordering is made over the second coordinate etc. in every resulting subset. Finally, after ρ iterations the set Ωh is divided into n^{λ} subsets of close points which would play the role of simple hypercubes in the case of a regular mesh. Resulting subsets will contain inner and boundary points. As on a regular mesh, the inner points will not require to access the neighbouring PEs for additional data. This should be done only when boundary points are computed. Calculate the overall number of boundary points, Ωh , when divided into n^{λ} hyperparallelopipeds each containing $n^{\rho-\lambda}$ points, has its boundary area equivalent tc the area $\rho\, n^{\lambda/\rho}$ of Ωh sections parallel to one of the coordinate axes. An average number of $\nu \cdot n^{\rho-1}$ boundary points is located near each section. Hence, the total number of these point will be $\nabla = \rho\nu M^{\frac{\rho}{\lambda}-\frac{\rho-\lambda}{\rho\cdot\lambda}}$.

As for the number of transfers between the neighbouring elements, this will not exceed that of boundary points

multiplied by ρ and the mean distance between random PEs. Thus

$$\Pi_H^M(M) = \frac{\rho\rho\nu}{2} M^{\frac{\rho}{\lambda}-\frac{1}{\lambda}+\frac{1}{\rho}} \log_2 M \qquad (26)$$

For the highest switching network performance the transfers will require

$$T_H^M(M) = \frac{\rho\rho\nu\tau}{2} M^{(\frac{1}{\lambda}-\frac{1}{\rho})(\rho-1)} \quad sec.$$

Comparing this with a similar result in the case of an array (25) yields

$$\frac{T_H^M(M)}{T_H^M(M)} = \frac{1}{2\rho} M^{\frac{1}{\lambda}+\frac{1}{2}-\frac{1}{\rho}} \xrightarrow[M\to\infty]{} \infty \qquad (28)$$

On a regular mesh, this relation was growing with M but less rapidly(23).

Let us consider two assertions.

Assertion 3. A linear increase of performance with increasing the number of PEs is impossible for a multicomputer having an array switching network when a differential equation is solved on an irregular mesh using explicit scheme.

To prove this compare the times required for arithmetic work (18) and performing transfers (25). They are related by

$$\frac{T_A}{T_\Pi^M} = \frac{4c}{f\rho\tau\nu} M^{-1/2} \xrightarrow[M\to\infty]{} 0 \qquad (29)$$

Thus the main portion of the multicomputer time will be consumed by data transfers rather than by arithmetic work for large M's, so the arithmetic performance will not be fully exploited in this case.

Assertion 4. According to assumptions made in Assertion 3, a linear growth of the multicomputer performance is possible, if it has a hypercube switching network. The proof is given by comparing arithmetic work time T_A (18) and transfer time T_n^M (27).

$$\frac{T_A}{T_n^M} = \frac{2c}{f\rho\tau\nu} M^{(\rho+1)(\frac{1}{\lambda}-\frac{1}{\beta})} \xrightarrow[M\to\infty]{} \infty \qquad (30)$$

In this case all the transfers are hidden by the arithmetic work time, thus the multicomputer performance will be proportional to the PE number for large M's.

4. INTEGRAL EQUATION

Discretized integral equation (6) differs from discretized differential equation (4) primarily in that ~~that~~ its right-hand side contains the terms integrating the effect of not only close points but also that of all the remote ones (see Section 1). Consequuently each point has ρ^2 associated coefficients, q_{ik}, that is we deal with larg-

amounts of data. If local memories were sufficiently large to include all $\rho^2 N^2$ coefficients q_{in}, no difficulty would occur.

Currently, it is unreasonable to rely upon such large local memories. A more reasonable approach is to consider the total multicomputer main memory as having a desired capacity. In this case each local memory would contain a part of needed data, $\frac{\rho^2 N^2}{M}$ allowing to compute not the total sum in the right - hand side of (6), but only its portion including M addends. Once possible computations have been performed, data circulation must be provided to permit transmitting data from one PE to another. After possible computation have been completed, processors resume data circulation. This procedure is repeated M times. It is clear that every q_{ik} from $\rho^2 N^2$ would appear only once in each memory of M processing elements. For an array architecture, this would require $\rho^2 N^2 M^{3/2}$ data transfers between neighbouring PEs which takes

$$T^M(M) = \frac{\tau \cdot o_i^2 M^{\frac{2\rho}{\lambda} + \frac{1}{2}}}{\tau} \quad sec \tag{31}$$

In a hypercube architecture, the global data circulation would require an average number of $\rho^2 N^2 M \log_2 M$ transfers to be performed between neighbouring PEs which results in

$$T^n(M) = \tau \rho^2 M^{\frac{2\rho}{\lambda}} \quad sec \tag{32}$$

if the switching network performance is fully exploited. Clearly, the arithmetic work time does not depend on the

switching architecture and is

$$T_A = \frac{C}{f} \cdot \rho^2 \cdot N^2 \qquad (33)$$

where C is a constant.

Comparing (32) and (31) gives

$$\frac{T_A}{T^M(M)} \sim M^{-1/2} \underset{M \to \infty}{\longrightarrow} 0 \qquad (34)$$

This is the reason for the following assertion.

Assertion 5. For a multicomputer with an array switching network, the linear law of performance growth with M does not hold when solving the integral equation (3) using an explicit scheme. Comparison between (32) and (33) allows the following assertion.

Assertion 6. For a multicomputer with a hypercube switching network, the perfomance may continue to grow linearly with M when solving the integral equation (3) using an explicit discretization (6), since

$$\frac{T_A}{T^H(M)} = \frac{1}{\tau f} \qquad (35)$$

does not depend on M.

5. SOME REMARKS

1. There are some papers which consider multicomputer projects where a strong mutual correspondance between PEs and difference mesh points is established. This results in a more narrow class of problems where the highest performance of the multicomputer can be achieved.

Those regions only become valid which are composed of some squares isomorphic to a PE array. The figure 5 shows an example of such region consisting of four squares.

In [3], for example, the highest performance can be obtained for bodies composed of several prisms with a square base containing 128x128 points. This restriction on the regions allowed should be considered as extremely severe. First, computed regions normally have a more complex shape in real applications, particularly in the case of time-dependent problems. Second, the point number selected for each computation region depends not on the region shape, but on the medium parameters. Finally, the number of points varies rapidly when the computation is performed. According to the above said, the authors of this paper believe that assuming a "polyprismatic" nature of Ω^{λ},Ω^{λ} regions cannot be considered as acceptable.

We feel that another approach is more suitable where the correspondance between PEs and region points is program-specified. Of course, this approach require

additional work of PEs and the operating system becomes more complex. However this limits considerably the restrictions on the computation region. In addition, such approach allows to switch out some PEs (for example, those that were failed) without deactivating the multicomputer and in the absence of large hardware resource. Naturally, a question about overheads arises for such implementation. In the worst case, a random correspondance may occur between PEs and the region points, that is a correspondance similar to that in the case of an irregular mesh. The estimates obtained above show that this approach can considerably degrade the multicomputer parameters for an array switching network.

However this approach is quite acceptable for a multiprocessor with a hypercube switching network, since its parameters will change slightly. The figure 6 shows a triangle-shaped region with the lower cathetus containing $\frac{n}{2}$ points and the left one containing 2n points. For a multicomputer with a tightly-coupled array architecture, this problem can be solved on a regular mesh using four runs. If one applies the irregular scheme, the problem is solved in one run using the same regular mesh. It should be noted that rigid coupling between adjacent mesh points and neighbouring PEs will be violated on horizontal boundaries of the square denoted by dashed lines to represent a PE array.

2. For problems described in Section 1, one would wish to have a sufficiently large local memory. However the large capacity results in higher hardware cost without increasing

the multicomputer performance which is undesired. As one can see, it would be more reasonable to implement a switching network by using a slightly more complex approach rather than constructing a pure hypercube, that is to connect PE clusters (instead of separate PEs) across the hypercube, each cluster, for example, containing 4, 8 ore 16 PEs. It is suitable to connect PEs over the whole graph inside the cluster thus permitting each cluster processor to access the memory of another PE. This approach increases the local cluster memory capacity with a factor of 4, 8 or 16 which now may achieve tens of megabytes thus making easy the algorithm developments for mathematicians and expanding the class of problems to be solved.

Finally, it is worth to note that clusters allow to switch out some processors or even processing elements without changing the relationship between the clusters; in this case redundant hardware is no longer needed. Notice that implementing a four-element cluster require an insignificantly greater amount of hardware and two new lines are only added per cluster.

3. Only explicit discretizations of (1) and (2) were considered. The results obtained can be applied to many iterative schemes without any modification. Discretizations solved with different ADI schemes require further investigations. Multicomputer efficiency issue for such schemes is extremely interesting and will be addressed later.

4. The host usage was completely ignored so far in this

paper. A sufficiently powerful host would perform a considerable portion of the work unsuitable for parallel exection, for example, matrix conversion using AD1 schemes. The latter circumstance may require that above mentioned assertions should be revised, particularly, Assertions 3 and 5.

5. We have shown above that a more complex architecture (hypercube) allows the multicomputer to increase its performance with growing PE number on an expanded class of problems and methods. The limiting case of a switching network is represented by a complete graph (see figure 7) or a "wheel" structure [6] where each PE is the nearest neighbor for any other PE. In this case any transfers are performed between the nearest neighbors. In addition to extenting the class of problems where the performance linearly increases, this facilitates the programming since transfers are no longer needed. Furthermore making redundant hardware available is considerably simplified. Theese favorable features, of course, are not free of charge. It is eas, to see that the amount of switching hardware is equal to the squared processor numwer, M, for a multicomputer with a switching architecture shown in figure 7, that is

$$\kappa(M) = \mathcal{E} \cdot M(M-1) \qquad (36)$$

This results in the cost of an arithmetic operation given by

$$\zeta^{\kappa}(M) = \frac{2}{\delta}(M-1) - 1, \qquad (37)$$

which is substantially higher for hypercube architecture (13) and array network (14).

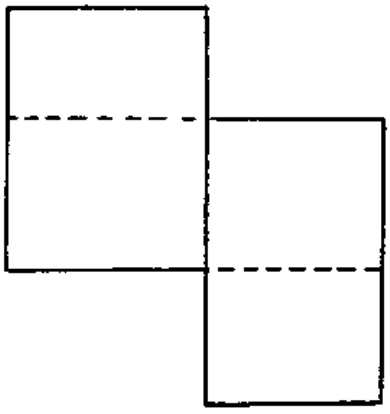


Figure 5

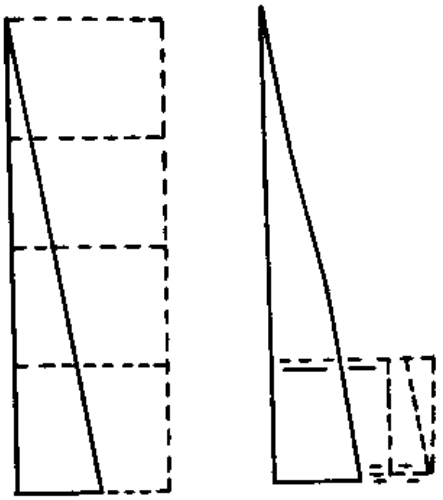


Figure 6

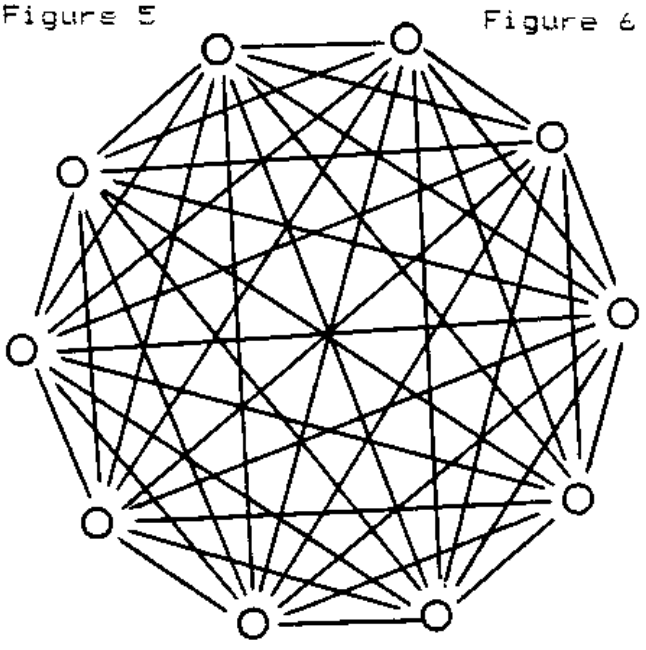


Figure 7

REFERENCES

1. Sofronov I.D.,Parameter estimation of a computer for solving problems in continuum mechanics, Chislennye metod. resheniya zadach mekhaniki sploshnoy sred,, vol.6, no.1, Novosibirsk, 1975.

2. Sofronov. I.D., Rasskazova V.V., Nesterenko L.V., The use of nonregular nets for solving two - dimensional nonstationary problems in gas dynamics Numerical Methods in Fluid Dynamics, Edited by N.N. Yanenko, Yu.I. Shokin, MIR Publishers, Moscow, 1984, pp. 82-121.

3. Andrianov A.N. et al., Solver structure for flo.. calculations. Computational processes and systems, Edited by G.I. Marchuk, Nauka Publishers, Moscow, 1984.

4. Bahvalov N.S. et al., Parallel algorithms for solving stationary problems in computational physics. Nauka Publishers, Moscow, 1991.

5. Zabrodin A.V. et al., BMS-14 array system structure and some single-base computer developments. Institute of Applied Mathematics, Academy of

science of the USSR,no. 134, 1990, Moscow.

6. Sofronov I. D., A model of Scientist Problems of the Science of Science. Special issue of the Polish quarterl, Zagadnienia Naukoznawstwa, 1974.

SESSION F

Distributed Computing

Focus of Study of the Mathematical Department
Valentin Kuropatenko (Chelyabinsk)

F-5

V. F. Kuropatenko

Focus of Study of the Mathematics Department

All-Union Scientific Research Institute of Theoretical Physics
(VNIITF)
City of Chelyabinsk-70

1992

1. Mathematical modelling

At the present time computer calculations are the most important part of the process of developing and building the manufactured products, tools, and machines which are produced at the Institute. Mathematical modelling is used at all stages of developing a new technological process, from the search for optimal scientific and technical problem solving methodology to tests on experimental specimens.

The components of mathematical modelling are the following:

- physical and mathematical models,
- applications programs,
- system programs and programming aids
- computer.

The department generally uses a calculation technology which is based upon fragmentation of design into nodes and physical processes and upon providing each type of problems with separate models and programs. There is a great number of physical and mathematical models; corresponding mathematical applications programs have been developed, and rich experience in their use has been accumulated.

2. Physical models

The following groups of models can be distinguished according to characteristic criteria:

- adiabatic mechanics of continuous media,
- mechanics of continuous media with strong heat flows,
- interaction of radiation with matter

- energy emission, transfer of neutrons and contaminated particles,

- aerodynamics

- properties of matter.

We will briefly reveal the content of each group of models, limited only to the main models.

2.1. Adiabatic mechanics of continuous media

The department has developed and is using the following original models of:

- ideal detonation,

- ideal (without tensor properties) and non-ideal (accounting for tensor properties) media (solid - extensibility, ductility, friability), liquid, gas and plasma (viscosity),

- isotropic and non-isotropic equiponderant and non-equiponderant disintegration of a solid,

- ideal and non-ideal porous medium,

- equiponderant polymorphic phase transitions and sublimations,

- unstable flows, destruction of contact boundaries and intermixing of substances.

2.2. The mechanics of continuous media with strong energy flows

Original models of the following are used:

- equiponderant and non-equiponderant plasma,

- transmission of radiation in equiponderant plasma with averaged coefficient of heat conductivity,

- transmission of photons in an approximation of radiant heat transfer and in spectral formulation or aspect,

- transfer of electromagnetic radiation and the interaction of plasma with an electromagnetic field and a gravitational field.

2.3. Energy emission, energy transfer

Original models of the following are used:

- the kinetics of nuclei and energy emission,

- the generation, multiplication, absorption, and transfer of neutrons with isotropic or anisotropic diffusion in spectral or

group formulation,

- transfer of γ quanta and contaminated particles.

2.4. Aerodynamics

Original methods of the following are used:

- stationary and non-stationary flow around aircraft by heterogeneous atmosphere,
- interaction of shock wave with moving aircraft.

2.5. Interaction of radiation with matter

Original models of the following are used:

- transmission, absorption and diffusion of X-ray emission,
- generation of electromagnetic radiation
- generation of laser emission.

2.6. Properties of matter

The following theoretical models have been created and are being used:

- Tomas-Fermi
- Tomas-Fermi with quantum and exchange corrections,
- Harter-Fok-Slayter,
- Sakh for plasma
- equiponderant and non-equiponderant liquefaction and evaporation,
- equiponderant and non-equiponderant dissociation and ionization,
- dislocation and disclination models of a solid for crystals and poly-crystal media.

Applied equations of state of:

- metals,
- soil and rocks,
- dense explosives (BB),
- component materials.

Storage and research on equations of state have been created:

- solid state physics database AREDUS,
- program complexes KRAB and SIRIUS
- nuclear physics data system KOBRA.

Figure 1 and 2 depicts the equations of state of KIM

Figure 1. Rhyolite

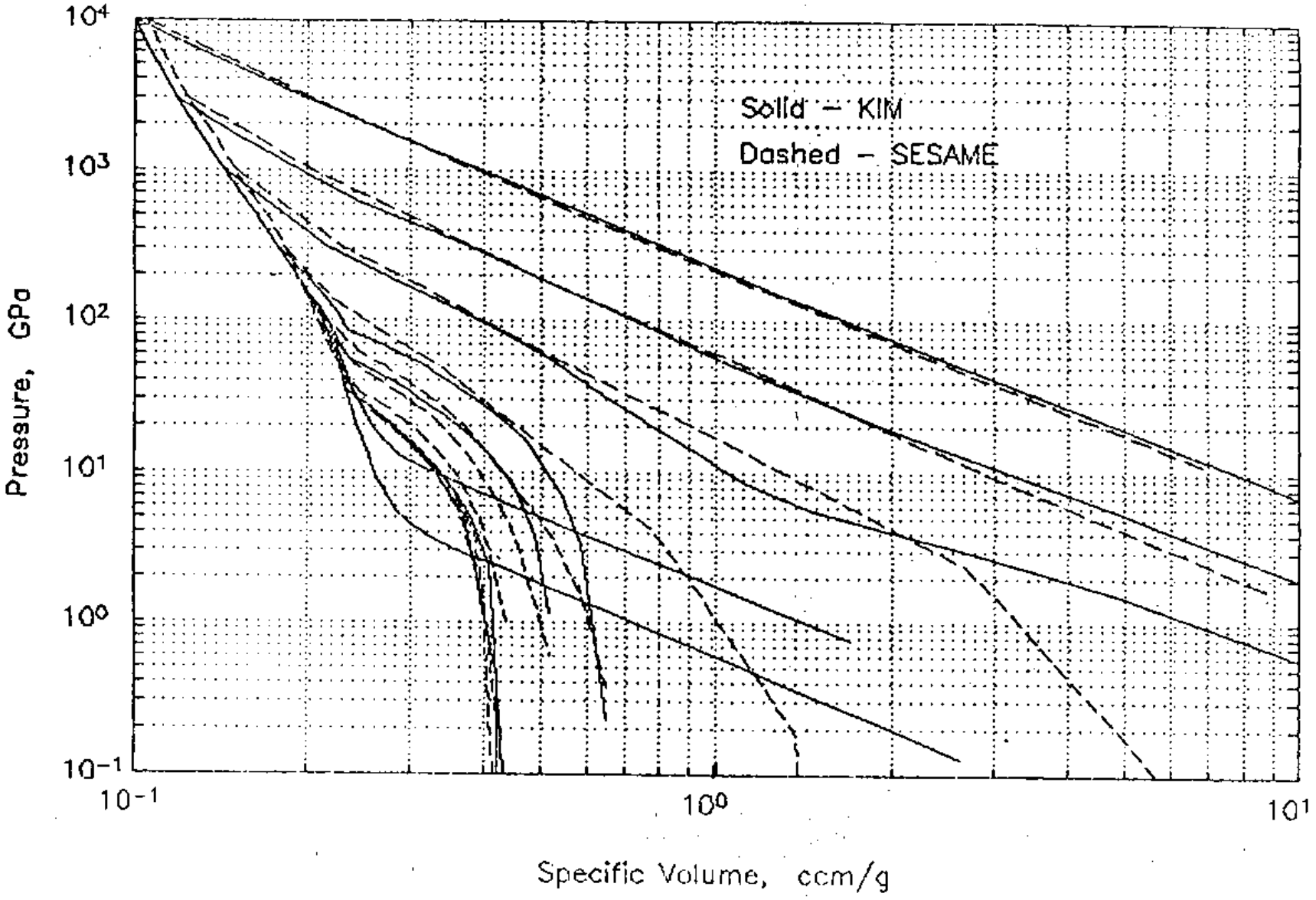
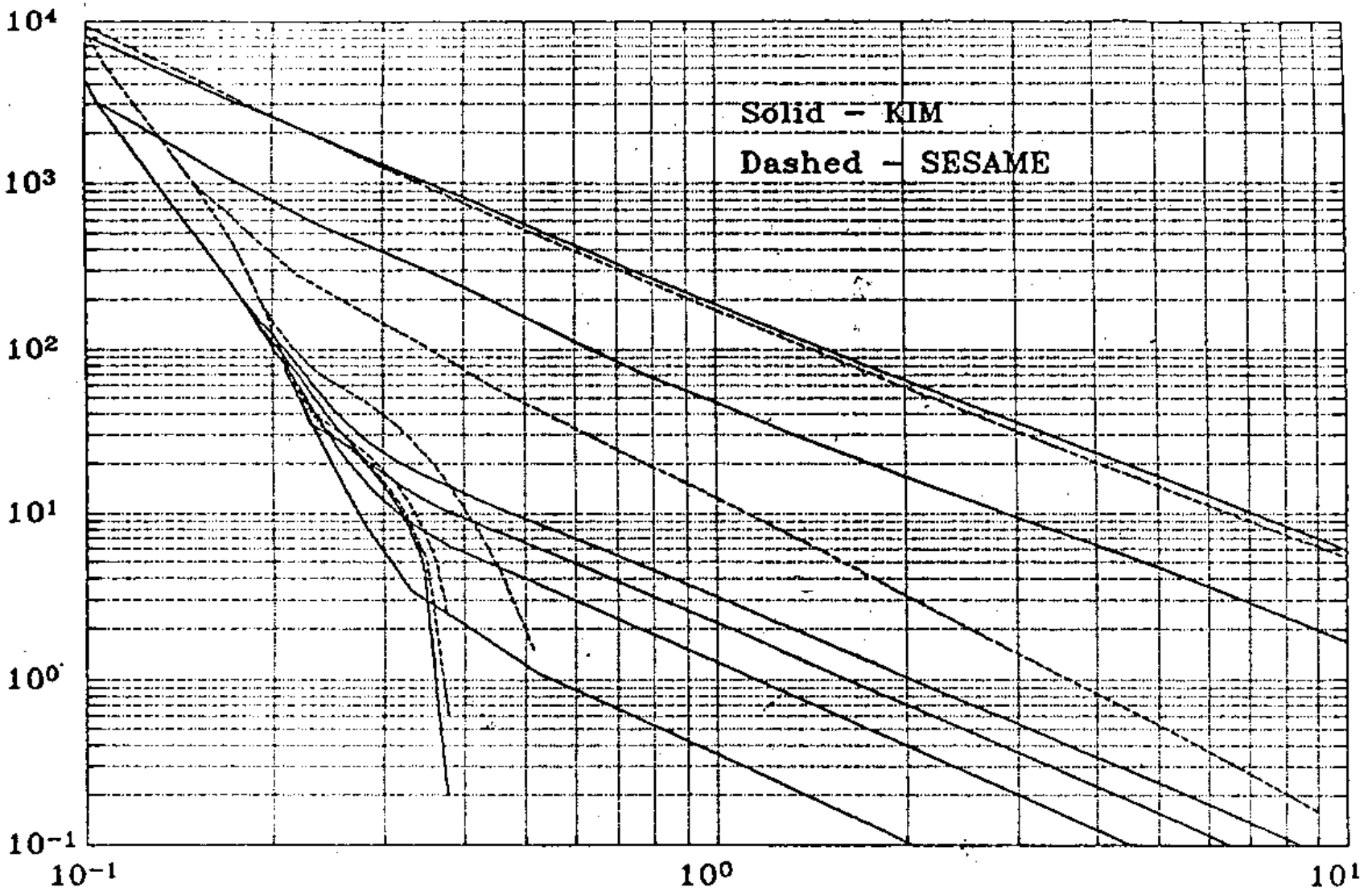


Figure 2. Granosyenite



(percussive adiabat and isentrope) for two kinds of rock in which a joint Soviet-American experiment was conducted - nuclear explosions in Nevada (rhyolite) and at the Semipalatinsk test site (granosyenite). A comparison was made with URS SEZAM (USA).

2.7. Unstable flows

To describe the behavior of matter in the area of instability of Kelvin-Helmholtz, Railey-Taylor or Rikhtmayer-Meshkov, models of the following were created:

- diffusion turbulent intermixing,
- multi-velocity interacting continua,
- separation model.

2.8. Kinetic models

Kinetic models of the following have been created and are being used:

- destruction,
- chemical reactions in explosives,
- limits of extensibility and ductility, depending upon the velocity of deformation.

3. Mathematical models and methods

Understanding of the problems and the requirement to guarantee a high degree of precision in mathematical experimentation requires perfection of the methods already available and the creation of new ones. Using one and the same design of several different methods for calculations allows us to evaluate the stability of results in relation to model and method. Let us focus on several general questions, problems and areas of development of these models and methods.

3.1. Explicit and non-explicit methods

The choice of correlation of steps across time and space is dictated by conditions of precision, stability, and solvability of problems in a reasonable time on a particular computer. Explicit methods are widely used to solve problems of adiabatic mechanics of a continuous medium. The original explicit absolutely stable

difference method was created for flows with low multiplicities. Non-explicit methods are used in problems with heat conductivity.

3.2. Homogeneous and heterogeneous methods

Homogeneous methods are used to solve the overwhelming majority of problems: "spreading" explosions in several units of a grid.

The heterogeneous method (VOLNA) was created to solve unidimensional problems of a broad class (gas dynamics, extensibility, ductility, heat conductivity, magnetic fields, phase transitions, and others). Separation of strong and weak explosions sharply increases the precision of calculations but makes the method extremely complex.

Partial separation of explosions is done in calculating two problems of gas dynamics in the program complex MAKh.

3.3. Optimization

Reliable means of choice have been established on a computer, of designs with the prescribed characteristics. Optimization methods shorten the time to plot the design, sharply reduce the volume of visually processed information, and permit a reduction in the number of personnel.

3.4. Metrology

Standard analytical solutions have been constructed for each class of problems. Each program and method must be certified as to precision and economy. The most important analytical solutions describe the following:

- converging and diverging shock and detonation wave,
- attenuation of the shock wave in a solid (short and long detonation),
- shock and detonation waves, in a two-dimensional formulation, produced by a curvilinear piston,
- heat shock,
- dynamic destruction,

- dissemination of heat waves
- splitting of shock waves in phase transition,

and others.

3.5. Grids

Precision of calculations depends heavily upon choice of grid.

The Lagrangian approach is used to calculate implosion. Grids are constructed, accounting for the properties of the future solution. Methods of construction are developed for regular and irregular grids.

In case of extensive deformations the grids are reconstructed with the aid of interpolations.

Euler's approach is more appropriate for calculations of explosion and aerodynamics. Contact boundaries are uncompacted by special markers.

Mixed coordinates are widely used:

- in a two-dimensional case LE (TIGR),
- in a three-dimensional case LEE (TIGR), LLE (GRAD),

Mobile grids are used, which are adapted to the solution.

Figure 3 shows the first irregular grid (Dirichlet's units) and the compression and divergence of the ellipsoid (Schultz's problem, RAPID complex).

4. Applications programs

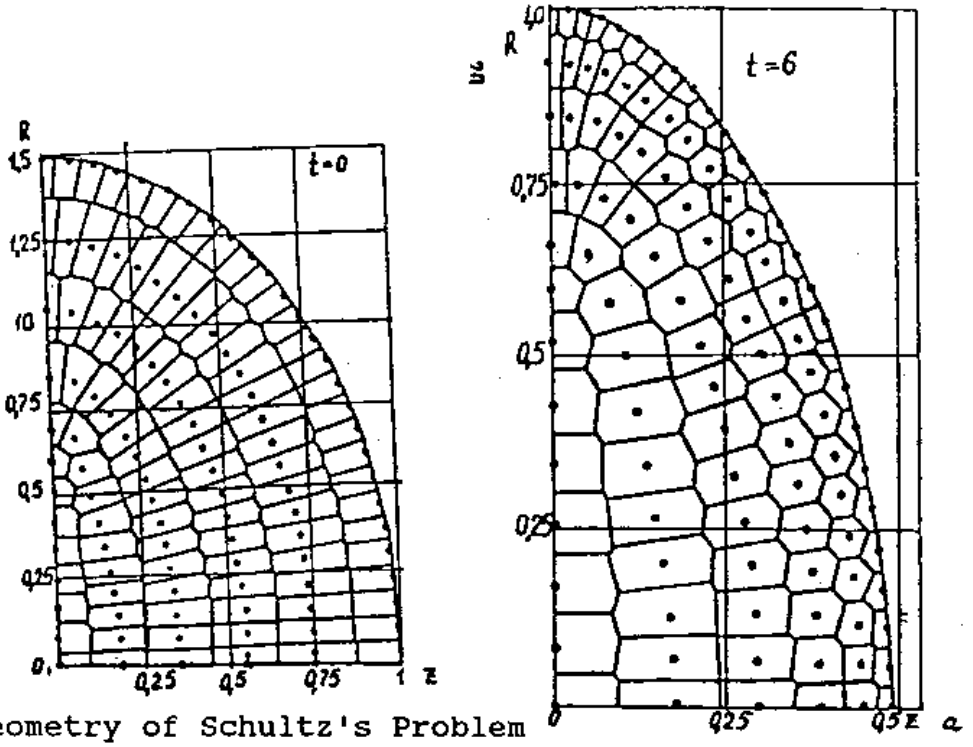
There are enough applications programs to optimize separate units of articles and an article as a whole.

The best:

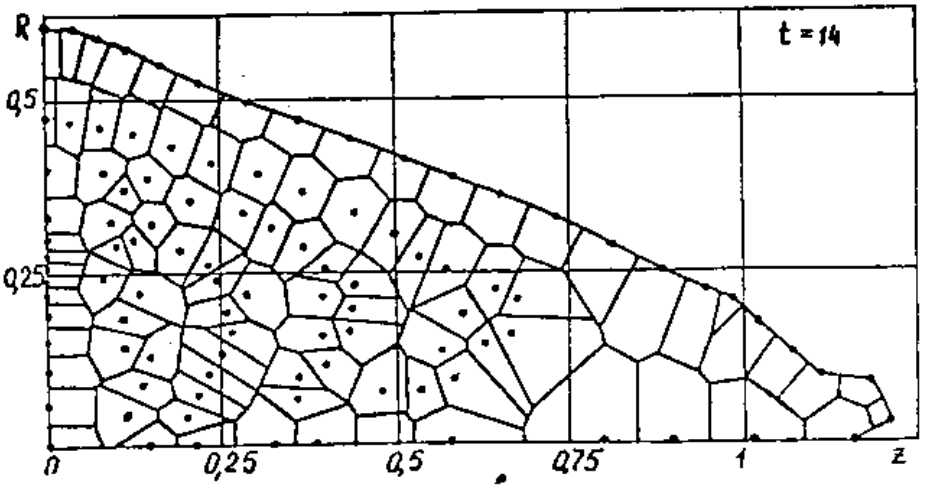
4.1. The VOLNA program complex. Figure 4 depicts a shock-wave pattern, obtained in calculating the flat automodel, layered system used to accumulate energy.

4.2. The MAKh complex for calculating the gas dynamics in a two-dimensional formulation. Figure 4 gives the results of calculating the disturbances of the contact boundary during transmission of a stationary shock wave from a light gas to a heavy one and from a heavy one to a light one. Amplitude a(t) is compared with V. Rupert's (Livermore) solution.

Figure 3. RAPID Complex
Compression and Divergence of Ellipsoid



Initial Geometry of Schultz's Problem



Grid of Dirichlet's units at moment: а – $t = 6$; б – $t = 9{,}15$; в – $t = 14$

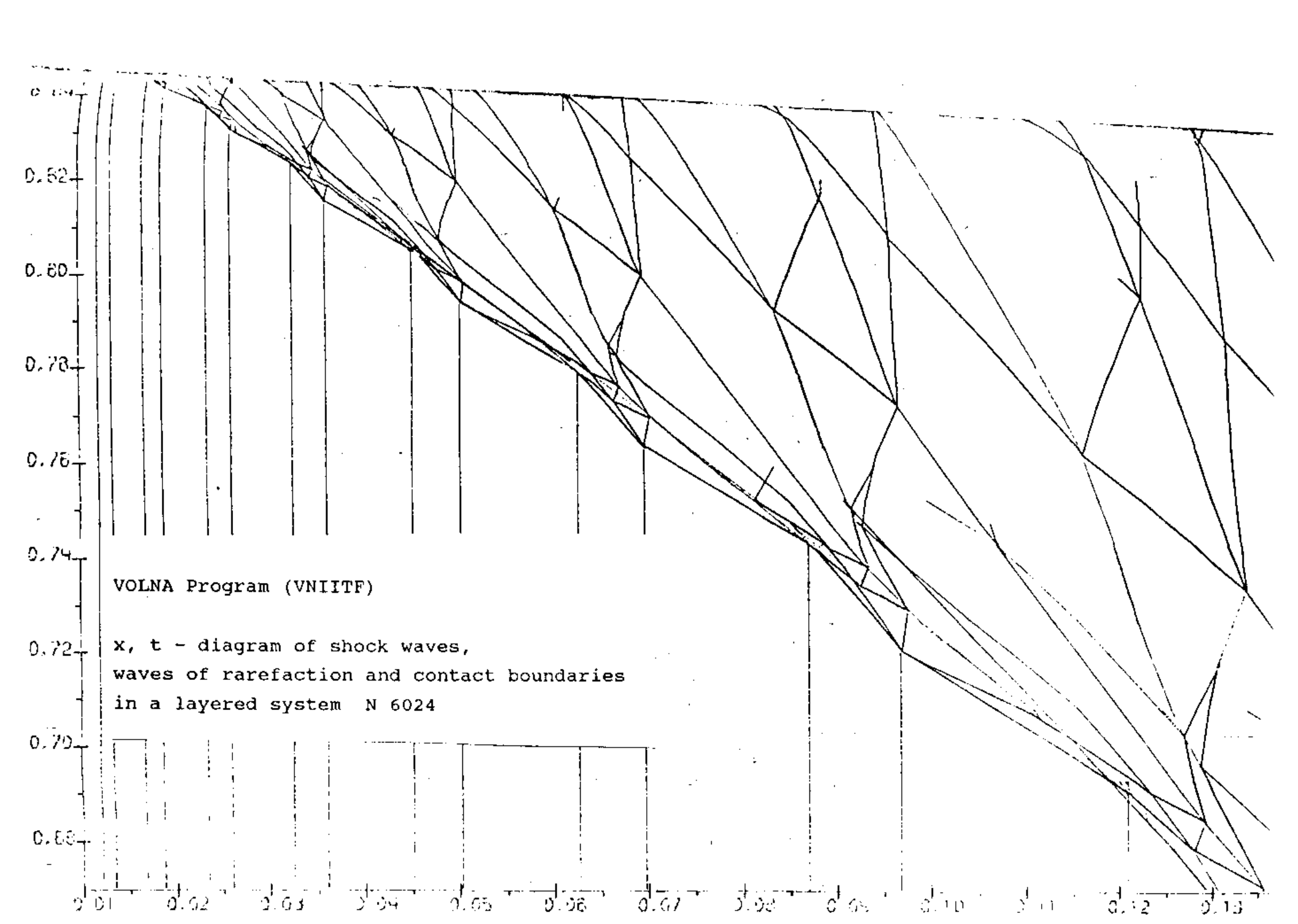
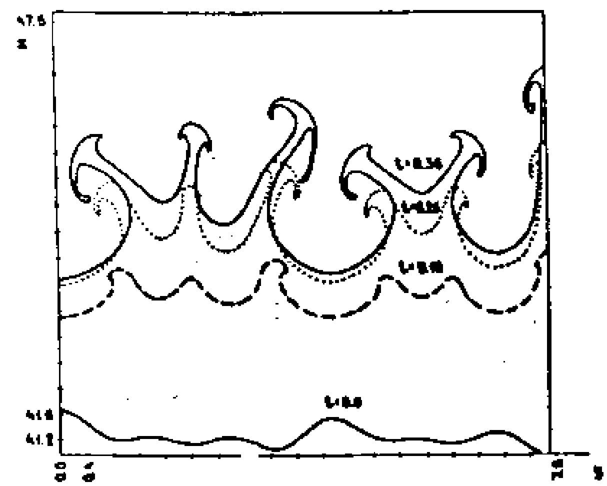
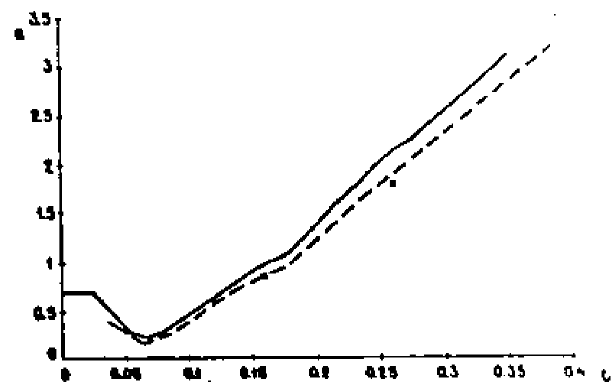

0.82
0.80
0.78
0.76
0.74
0.72
0.70
0.68
VOLNA Program (VNIITF)
x, t - diagram of shock waves,
waves of rarefaction and contact boundaries
in a layered system N 6024
coordinate

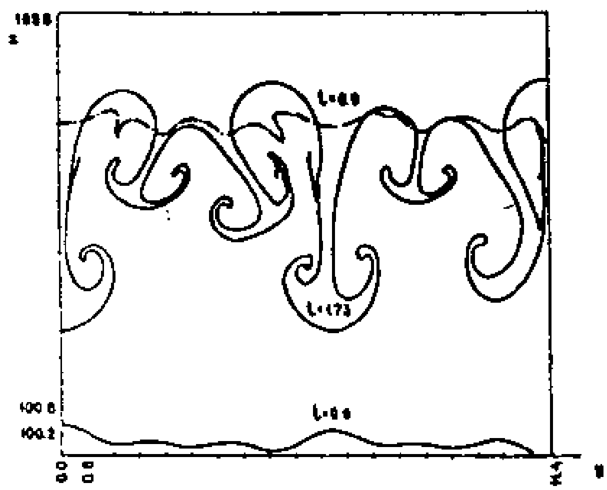
Fig. 4



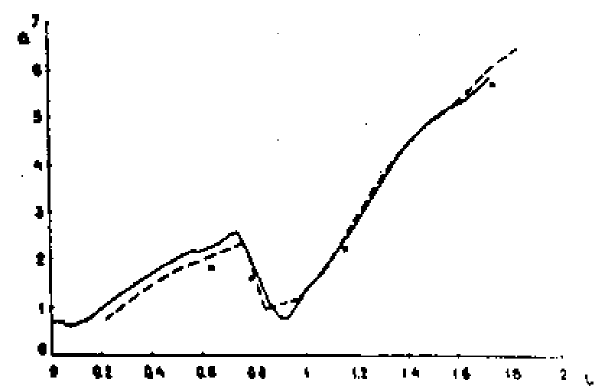
Воздух-НЕ. Суперпозиция мод. Форма КГ на моменты времени t = 0,0, 0.16, 0.26, 0.36



Воздух-НЕ. Суперпозиция мод. Зависимость амплитуды $a(t)$ в расчетах "МАХ" и В.Руппорт (x – значения амплитуды, полученные по обычной экспериментальной [illegible])



Воздух-SF_6. Суперпозиция мод. Форма КГ на моменты времени t = 0,0, 0.9, 1.73



Воздух-SF_6. Суперпозиция мод. Зависимость амплитуды $a(t)$ в расчетах "МАХ" и В.Руппорт (x – значения амплитуды, полученные по обычной [illegible])

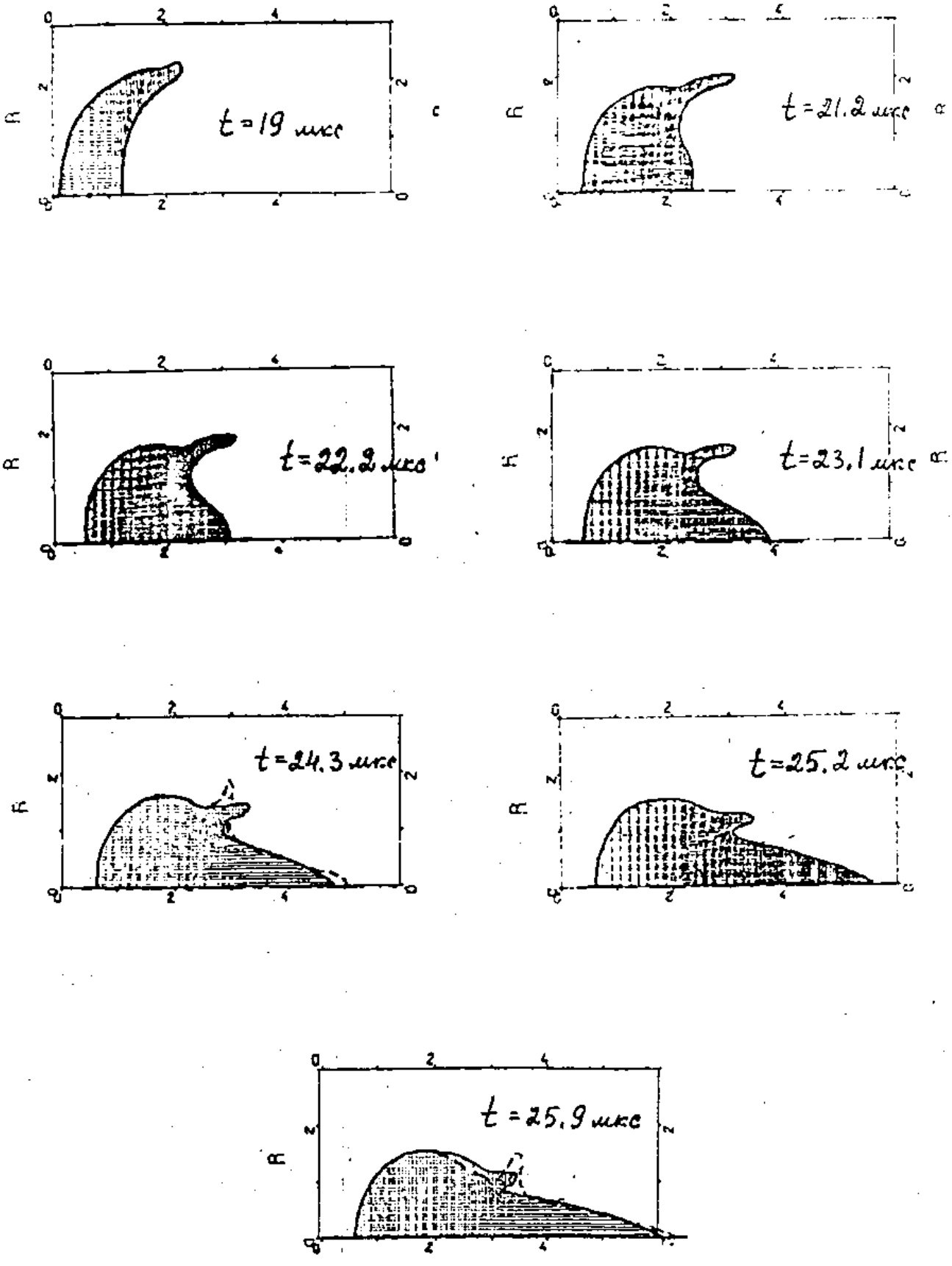


Figure 5

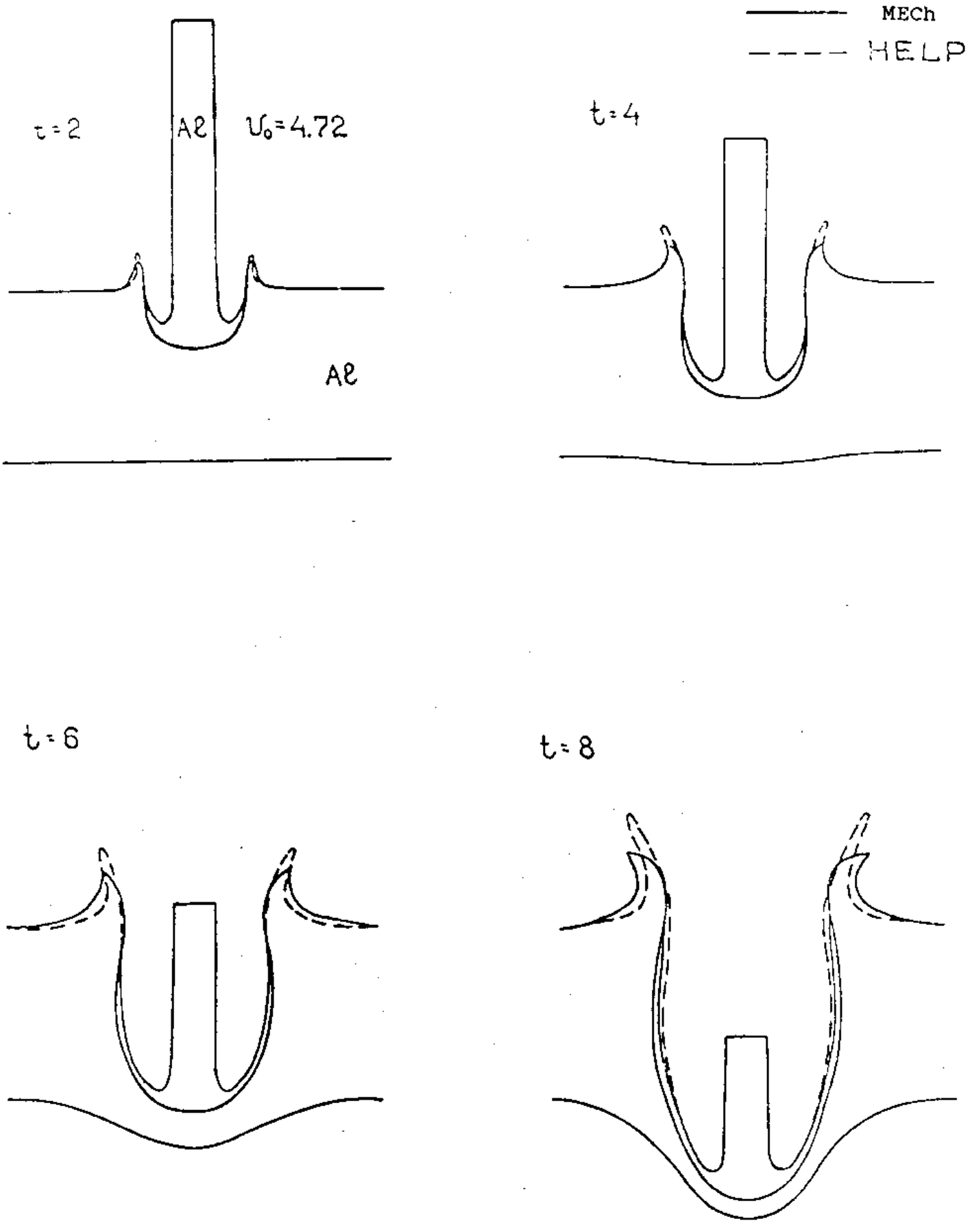


Figure 6

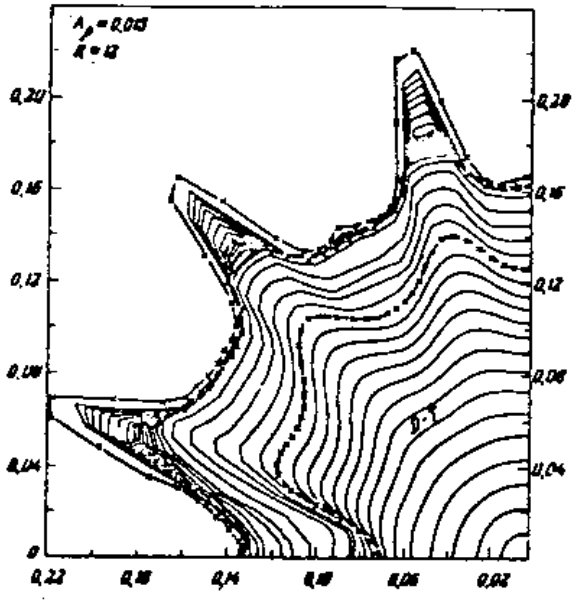
SHELL TARGET IMPLOSION ASYMMETRY.

Letters of the J. Exper. Theoret. Phys. 26, No 9, p.630, 1977).

TARGET: GLASS SHELL DIAMETER 300 μm, WALL THICKNESS 3 μm, DT-GAS PRESSURE 5 atm.

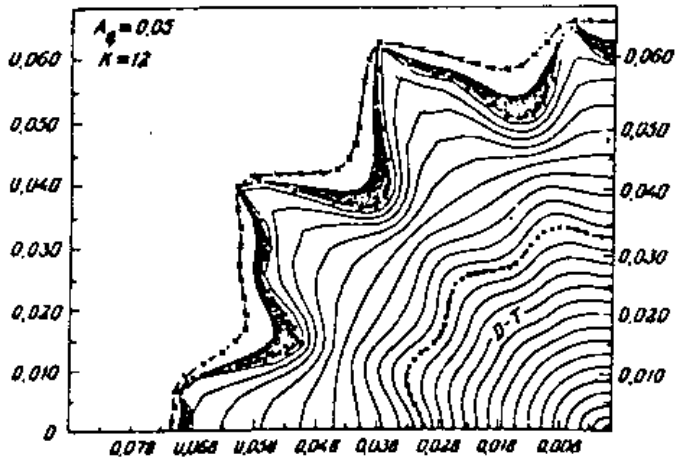
PULSE: GAUSSIAN SHAPE, E = 1 kJ, τ = 0,8 ns.

100% ABSORPTION OF LASER ENERGY IN OUTER LAYER WITH 0,1% MASS OF SHELL. CALCULATIONS WERE PERFORMED WITHOUT RADIATION TRANSFER.



SHELL DENSITY PERTURBATION:

$$\rho(\theta) = \rho_c (1 + A_\rho \cos(k\theta))$$



IRRADIATION NONUNIFORMITY:

$$P(t,\theta) = P(t)(1 + A_q \cos(k\theta))$$

Figure 7

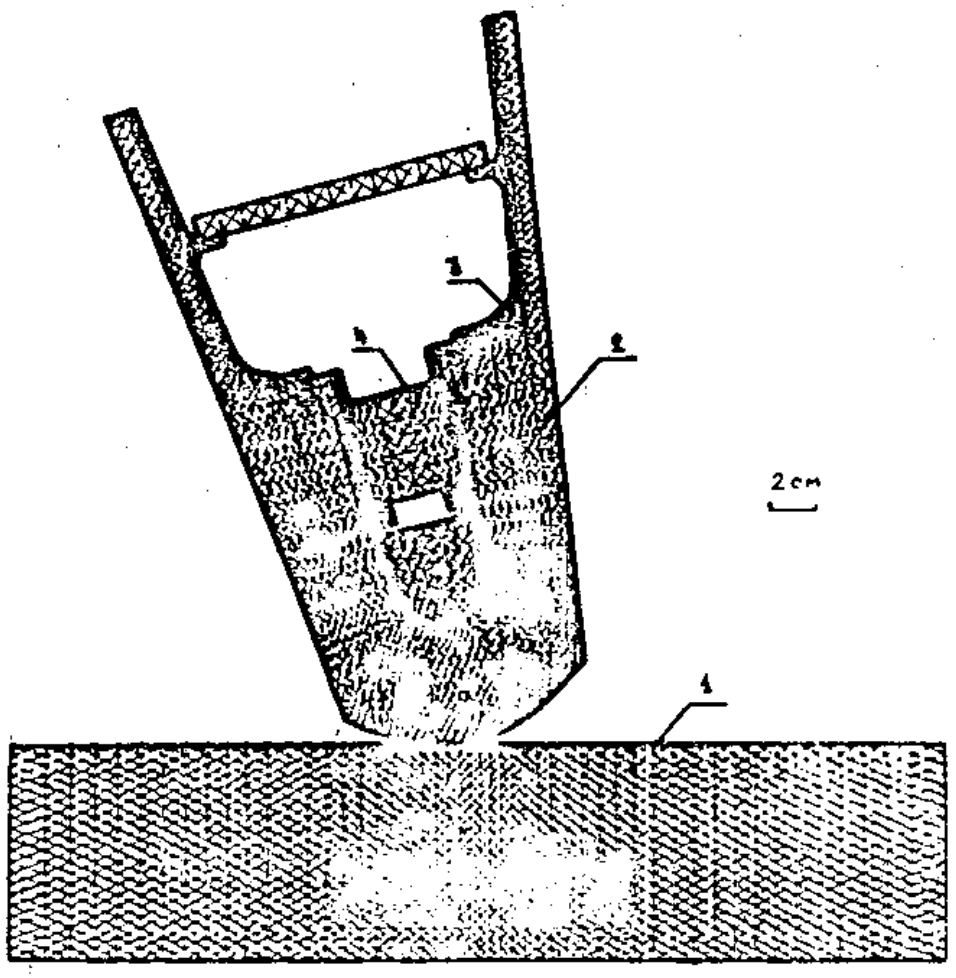


Fig. 6

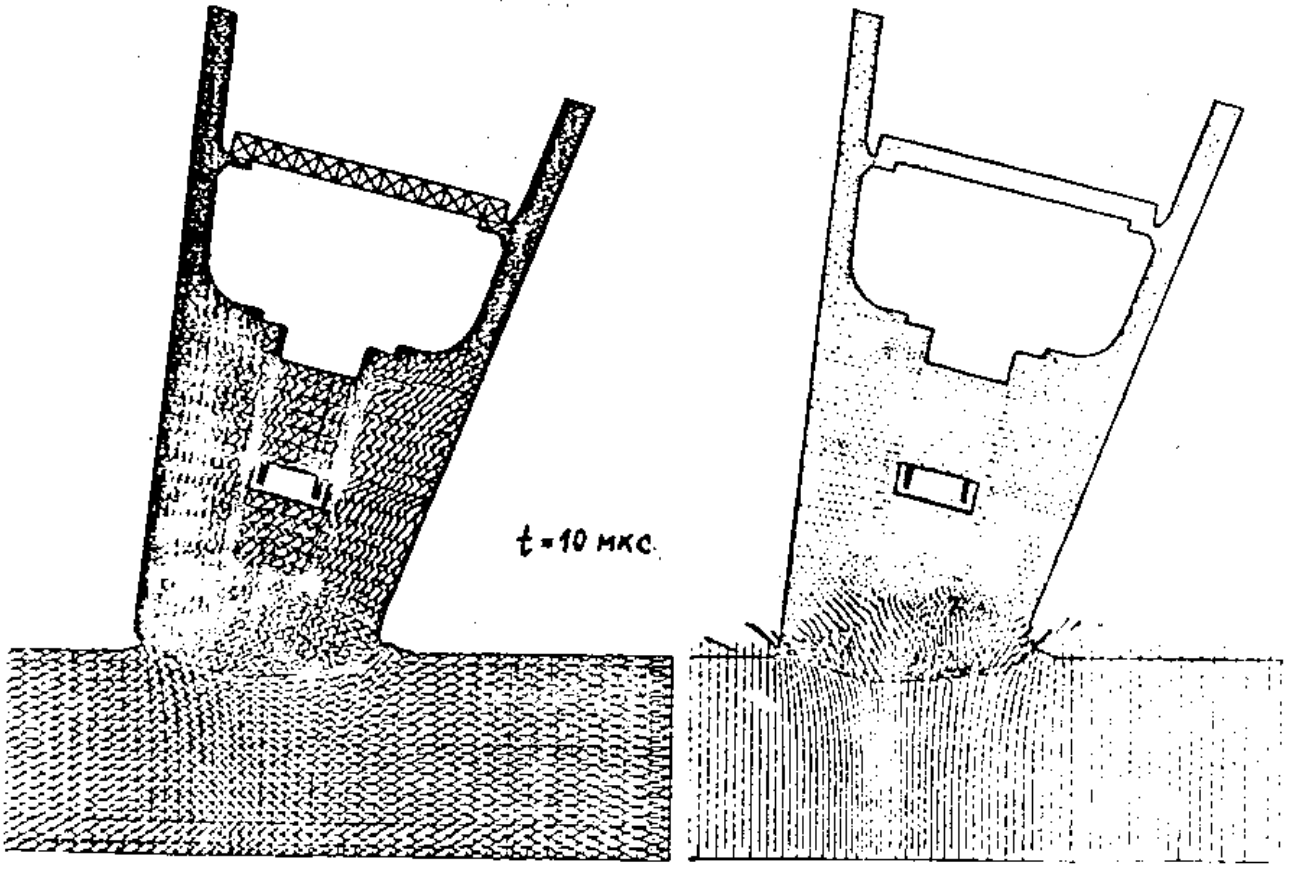


Fig. 9

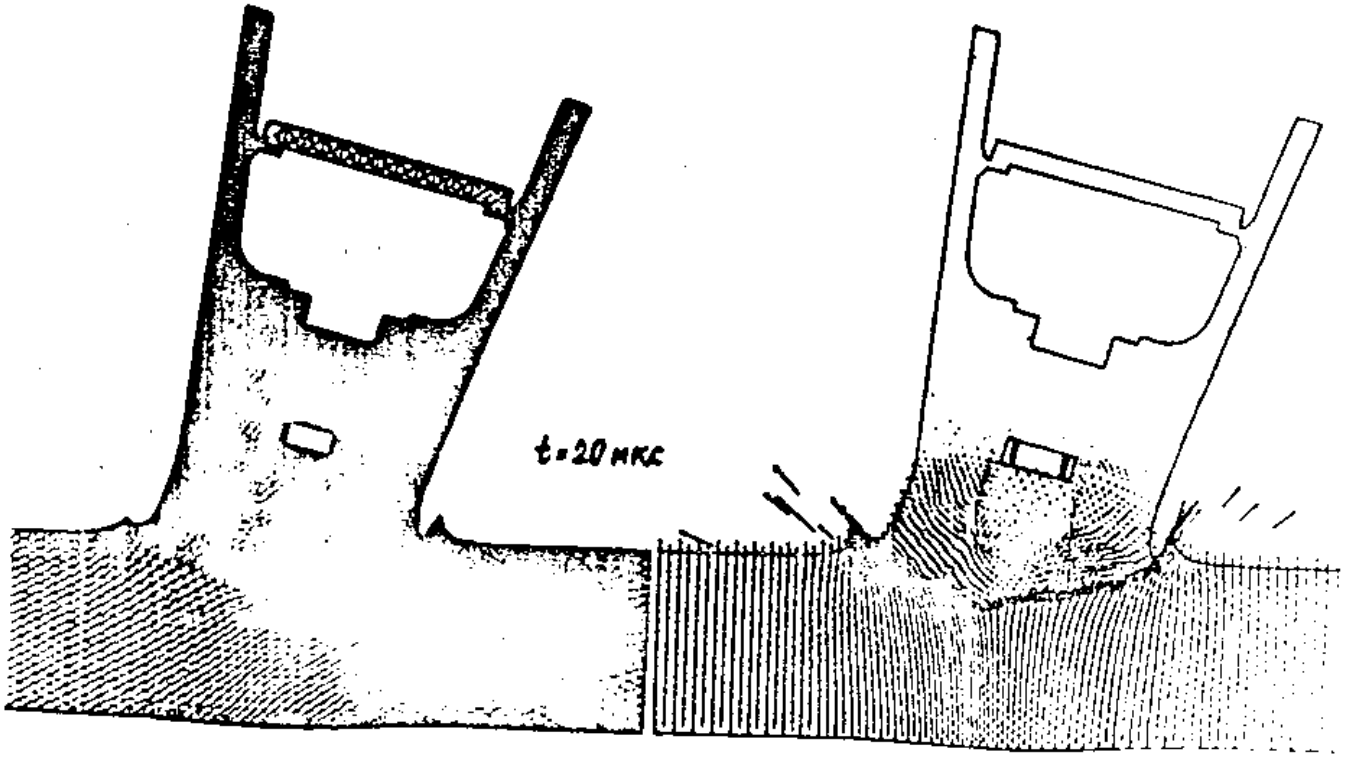


Fig. 10

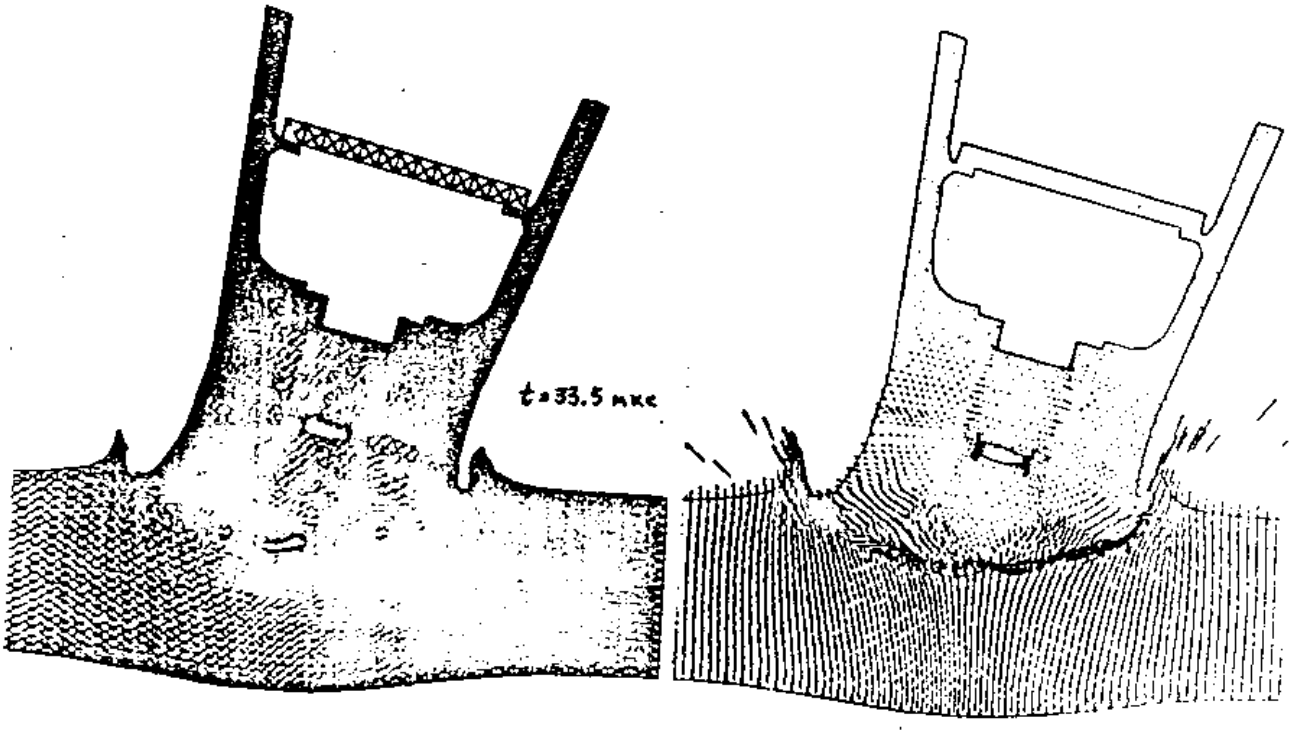


Fig. 11

4.3. The MECh complex (particles method). Figure 5 shows the formation of a shell from cumulative casing. A comparison with experiment (X-ray radioscopy) for the last moment of time.

Figure 6 shows fragments of a problem concerning piercing of an aluminum plate by an aluminum rod. A comparison is given using calculation according to the HELP (USA) program is given.

4.4. TIGR Complex

Figure 7 shows the form which is taken by the initial perturbations of density and surface of a glass shell in the process of its compression after impact of laser emission.

4.5. MKE Complex (Method of terminal elements)

Figures 8, 9, 10, and 11 show the stages of impact of a conical body with a concrete barrier (geometry and velocity field).

4.6. MK Complex (Monte-Carlo Method)

Figure 12 shows the results of calculating the field of particles around a ballistic rocket. Monitoring facilities, based on the calculations, permit the certain determination of the quantity and location of nuclear charges in the area of the shell.

5. Programming

System software is the combination of associated program systems:

- operational,
- programming,
- data management,
- remote access,
- provision of computer network,
- information protection.

The most widespread programming language is FORTRAN. Also used are ALGOL, PL-1, PASCAL, SI, ADDA, EL-76 and others.

Instrument technological support systems have been built for modular structure programming, graphics and information retrieval systems, and database management.

6. Computation Complex

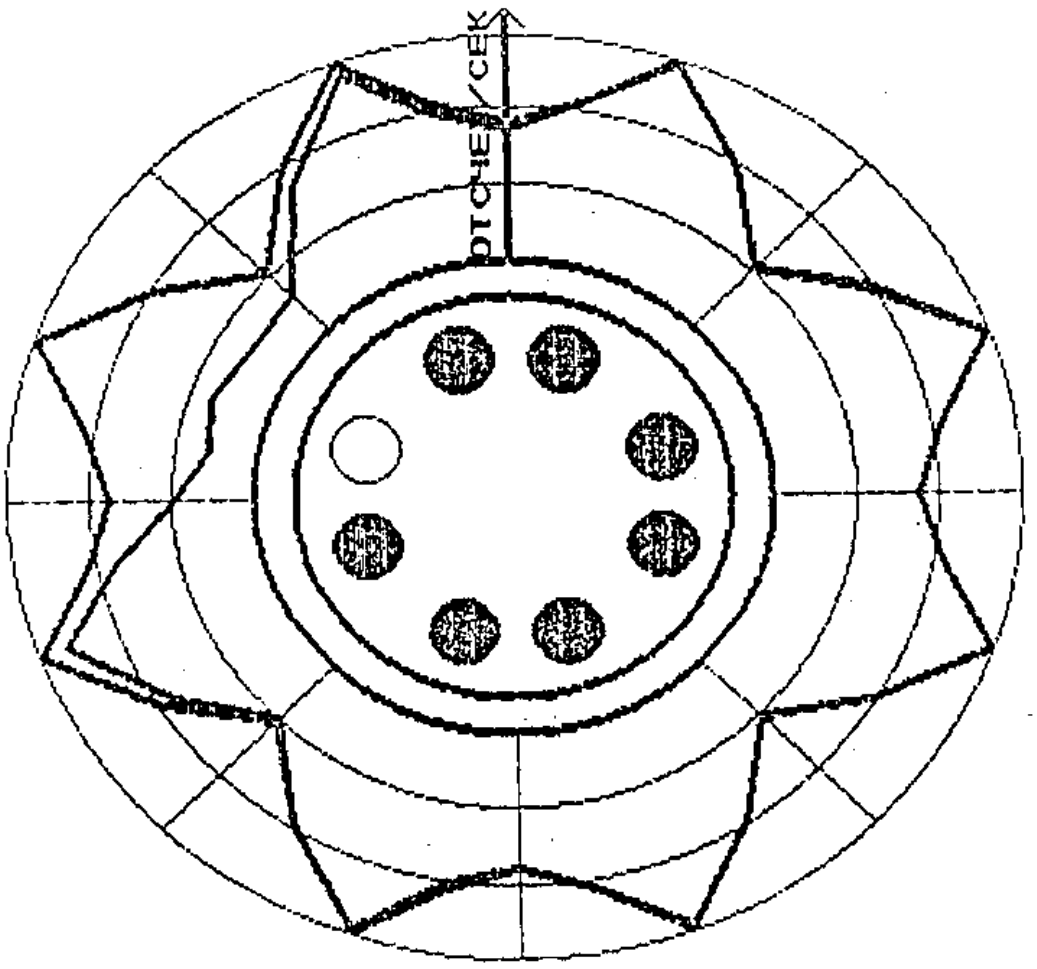


Figure 12. Dependence of velocity of recording of stilbene detector upon angle (photon) for 7 and 8 radiation units (placement in mine)

All computers are operated as part of a central computational complex (CCC). The local CCC network provides the following:

- transfer of files among subscribers,
- virtual terminal system,
- remote input of assignments,
- exchange of information among any CCC sites.

200 Million Ops/sec

The CCC foundation:

- common information base,
- common terminal network,
- common information input-output subsystems.

The local network (LAN) foundation:

- bus topology,
- simple connection protocols,
- modular structure.

The bus commutator assures speed of information transmission at 20 Mbytes/sec. The number of subscribers is 256, with the possiblity of expanding through intermediary stations.

The data transmission system (DTS) includes:

- data bus,
- processor bus
- terminal equipment bus (300 terminals, 80 personal computers, 12 mini-computers).

All equipment is domestic. Most of the apparatus has been developed here.

7. Conversion

The mathematics department is conducting work on the following topics in the area of conversion:

only producers of fiber in Russia.

- fiber-optics communication systems (modelling of fiber elongation, light transmission, signal attenuation on heterogeneities, development of central commutation stations),

Safronov is head.

- participation in development of the super computer project, monitoring of nuclear weapons tests,
- safety of atomic energy installations (neutron, heat, reliability and hydrodynamic problems),

- ecology (modelling of dissemination of pollutants, the creation of a data base, and provision of ecological monitoring),

- development of equipment for connecting different types of computers,

tomography (modelling of X-ray emission transmission and image reconstruction).

. They are the lead because there was only imported tomography machine.
. minimizing exposure with good resolution.
. financing from the central government for tomography.