Efficiency of the numerical algorithms and the decompesition principle for modern compute~~r~~s

The notions of the algorithm and its effectivity have changed essentially in recent years as a sequen due to (~~because of~~) the rapid development of the mathematical modeling and ~~complicated structur~~ growing complexity of the computers.

6. VII.83 We shall restrict ourselves by considering the ~~practic~~ algorithms ~~of the specific~~ specifically oriented towards ~~the~~ mechanics of continus, although many points of view presented at the paper are of ~~skal~~ ~~Re~~ general character. This orientation tanes into account the traditionally unified ~~system~~ ~~approach~~ treatment of mechanical problems, which is ~~base~~ based on: i) conservation laws, ii) closure relations, ~~terms~~ iii) interpretation of ~~the~~ a global process as a sum of of local ones, ~~re~~ representati

iv) representation of the interaction of adjacent domains through boundary conditions (autonomous or nonautonomous)

1

Real computational agrorithm

Real computational algorithm (r.c.a), executed on a computer, consists of two parts:

i) analytical algorithm (a.a)

ii) cybernetical algorithm (c.a)

A.a. is a sequence of arithmetical algebraical, symbolical operations, executed conditionally or unconditionally, orderedly in a given order - serially or in parallel - on the logical elements (l.e) of a computer. The information connected with the a.a. is transformed on l.e. Sometimes this transformation is followed by the loss of information, for example typical. The typical case is the loss of information in arithmetical operations, due to round-off errors

On the contrary, there is no loss of information in algebraic and symbolic manipulations.

C. a consists of the information transfer, ~~executed~~ realised ~~in~~ by ~~the~~ elements, ~~and~~ functional units (f.u.'s) and interconnection ~~communication~~ networks. ~~Logical~~ Manipulation of ~~generating~~ the instructions ~~codes~~, logical reconfiguration of the f. u.'s and their interconnections belong to the C. a too.

All these operations are information conserving

As long as the computers were slow, and their structure simple, the ~~quality~~ efficiency of the C. C. a. was measured essentially by ~~the quality~~ that of the a. a.

In particular the time for information transfer was not ~~[illegible]~~ taken into account ~~consideration~~

[Along with the increasing ~~growing~~ speed of the computer and complexity ~~and~~ the relative execution time of the information transfer ~~cybernetical algorithm~~ grows ~~more~~ increases too.] ?

~~good~~ ~~good~~ concordance between
all components of the computing
process.

In this way the ~~quality~~ efficiency of the 2.c.a. ~~shall be~~ is intimately connected ~~not only~~ with the structure of the a.a., ~~but~~ and that of the program but with the architecture of a computer and the implementing ~~implanting~~ the 2.v.a. on it ~~of realization of the 2.v.a. in the computer (in the one) (in that) (in it)~~

All the more ~~gets~~ insistent ~~(urgent) get~~ is the ~~necessity~~ the need for [harmony correspondence coordination] of all components of the computation process, that is the good correspondence of its parts.

2 The structure of the complex algorithm

For large ~~problems of the mechanics~~ of continua The modular structure of the algorithm is universally ~~accepted~~ acknowledged as most efficient way of ~~computing~~ numerical solving the large problems. ~~of mechanics of continua~~

An ~~The~~ algorithm, modelling the evolution or the state of a complex physical system, is ~~divided~~

into the parts (called modules, which correspond ~~that~~ to those of the system (decomposition of a system into subsystems and correspondingly of the algorithm into modules). The modules are joined together with the aid of the matching module. This kind of the modular design we call the physical decomposition

There is another approach for constructing the modular structure.

The algorithm, representing the behavior of the system, maps that by means of finite difference or finite element approximation on the system of linear algebraical equations, as a rule, with ~ big number of unknowns (global algebraic system, g.a.s.).

The g.a.s is divided into a set of simpler systems, which are solved separately (serially, or in ~ parallel way)

after exchange of data. ~~and then joined together~~

~~Their~~ These partial solutions are joined together into ~ general global solution by means of the matching module.

We ~~with~~ The second approach will be called algebraical decomposition. This approach is more general, but the algebraic decomposition is not so clear, as the physical one

Over more the most important aim of ~~the~~ an approximation is constructing of an adaptive mesh and analytical representation in the numerical cells, to get convenient structure of the global algebraical matrix

Here we come to very important problem of constructing the matrix by ~~means~~ the choice ~~of coordinate system~~ of both algebraical algorithm and informational medium

It is to be noted here, that to facilitate synthese ~ of a global algorithm, its modules should be homogenous and their basis ~~minimal~~ relatively small.

ce| In the papers [ ] were ~~given the definitions~~ introduced the notions of the simple (homogenous), ~~module~~ antinomous, nonantinomies modules and of modular (basis), ~~base~~, local, global algorithms.

Simple module is called a program, realizing analytical algorithm and having following properties:

i) stability (control over the) of the algorithm and ~~its~~, in particular, over the ~~conti~~ condition ? ~~control over the~~ numbers of the matrices, used in algorithm.

ii) ~~control~~ control ~~of the~~ over the approximation and accuracy of the algorithm ~~and their controllability~~

iii) ~~homogenous~~ uniformity of the difference scheme

iv) ~~homog neg~~ homogenity of the informational medium (regular curvilinear mesh, regular triangulation)

v) ~~concordance (consistence, compatibility)~~ correspondence of the analytical model to the physical one ~~of the physical and analytical model~~:

The Algorithm approximates a boundary Cauchy problem globally HA 02

for one integration step, i.e. approximates solution or step operator

VI) compactness (~~homogeneity of the memory~~ / ~~compaction of the m.~~ closedness of the memory) : the program and data (set), associated with (simple modules) are mapped in some subdomain Ω of the operational ~~homogeneous~~ memory, fixed for the execution time. ~~that is used during the execution time.~~

VII. Completness : the data (set), associated with the (simple) module ~~belong~~ ~~elements of a given~~ , ~~are~~ (consists of) arbitrary functions, belonging to some functional ~~&~~ class.

The property VII garantees flexibility and adaptivity of the modular system, (which are especially) important for (matching the modules) ~~joining up~~ in a large program and for (the transportability of) ~~transfer of the~~ of the modules ~~from one program's preset package into another one~~.

VIII) the possibility of utilizing the module in both (modifications) ~~versions~~ : (autonomous and non autonomous) ~~one~~.

In the case of the autonomous module M_i $[n\tilde{\varepsilon} \to (n+1)\tilde{\varepsilon}]$, all input data $In(M_i)$ are computed from output data $On(M_i)$ of the same module and the data of another modules $M_j\ (j \neq i)$ are not used. For the nonautonomous modules the transformation and/or transfer of data from another modules, belonging to $\{M_j\}$ is necessary, the transfer and/or transformation of data is realized with the aid of matching module.

For autonomous module M_i $[n\tilde{\varepsilon} \to (n+1)\tilde{\varepsilon}]$ input data $In(M_i)$ are computed from output data $On(M_i)$ of the same module and the data of another modules $M_j\ (j \neq i)$ are not used.

It is to be noted here, that the notion of the simple module

mainly
~~essentially~~ is connected with the properties of homogenity and it means that the simple module can be decomposed further into smaller simple modules of smaller size.

This process of decomposition can be continued to the modules, which cannot be decomposed further*. These may be called basic ones and constitute the basis of the modular system. The main aim of good modular design is minimizing the ~~number of~~ modular basis and factorizing at the same time ~~the condition of~~ a simple and clear macro language

(4) The classification of algorithms for simple modules

Before considering the effectivness of the qualified algorithms, pertaining to a simple module, we shall introduce some relative to ~~connected with~~ notions, and symbols, the structure of the module and of the modular system.

Ref 16a

First of all we shall give the definition of module tuning

8.VII.85 Definition. In the case of simple analytical module We call tuning a (nonantonomous) module $M_i \in \{M_\alpha\}$ the computing coefficient functional, (associated with) ~~acting to~~ M_i, ~~from~~ (through) output ~~output data~~ input data, ~~taken~~ (transferred) from the system $\{M_\alpha\}$ as output data of (its) modules $M_j \in \{M_\alpha\}$, ~~the case $i=j$~~ excluded. ~~M_i included~~. For the nonantonomous module M_i the data are taken (only) from M_i itself. For the cybernetical module tuning (is the) transformation of logical (inter) connections and change of codes. If an analytical module contains logical branching, tuning the module is transition from one logical branch to another one, realization (of the module) get more difficult in this case.

Let us introduce some notions and symbols (associated with the module M_i ~~obtaining to~~

Nonantonomous tuning

from the page 10.

m_i – continuous module 12 used in the algorithm of M_i

d_i – discrete module " "

n_i – the mesh pertaining to the algorithm of M_i

Ω_i – (domain of initial value data of M_i

ω_i – data set adjoining to Ω_i and used for matching non autonomous module M_i with another ones from the set $\{M_k\}$

$\bar{\Omega}_i = \Omega_i + \omega_i$

Ψ_i – the data set of coefficients functions pertaining to M_i

Z_i – the data set of boundary conditions pertaining to M_i

Y_i – the data set $\{\alpha\}$ of undefined boundary conditions, corresponding to a nonautonomous module M_i

y_i – the data set, adjoining to Y_i and defined by the condition:

$\bar{Y}_i = Y_i + y_i$

where $\bar{Y}_i$ is minimal data set, necessary for obtaining

the ~~set~~ numerical values of $\{\alpha\}$

W_i – data set, associated with the module M_i, necessary for its computing

Following relations hold:

$W_i = \Lambda_i + \Psi_i + Z_i$ (autonomous module)

$W_i = \Lambda_i + \Psi_i + Y_i$ (nonautonomous module)

$W_i = \bar{\Lambda}_i + \Psi_i$ (nonautonomous explicit module)

~~It is to noted, that there~~

There exists two classes of nonautonomous modules: those with ~~algebraical~~ numerical ~~fixed~~ boundary conditions and those with partly algebraical, partly ~~fixed~~ numerical b.c.'s

The modules of the first kind will be ~~denoted by~~ called numerical ~~algebraical~~ ones

~~In the whole~~ M_i can ~~is to~~ be considered as ~~a~~ function of the data sets Λ_i, Ψ_i, Y_i:

$$M_i = M_i(\Lambda_i, \Psi_i, Y_i)$$

~~For fixed~~ data Ψ_i, ~~fixed~~ the module M_i get autonomous, realizes ~~a~~ a step operator and can ~~is to~~ be considered

the modules of the second kind – the algebraical ones.

as the function of Ω_i:

$$M_i = M_i(\Omega_i)$$

Let us introduce the notion of

frame (general) module

~~By definition,~~ A module $M(\mathcal{X}, Y)$, is called frame module if the following conditions are fulfilled:

i) $\mathcal{X}, Y$ are standard data sets

ii) M is standard representative of a modular system $\{M_i\}$

iii) matrices of $M_i\{\mathcal{X}_i, Y_i\}$ are equivalent to that of M.

The representation ii) can be put into the form:

$$M_i\{\mathcal{X}_i, Y_i\} = T(\mathcal{X}_i \to \mathcal{X}, Y_i \to Y) \cdot M(\mathcal{X}, Y)$$

where $T(\mathcal{X}_i \to \mathcal{X}, Y_i \to Y)$ is transfer module.

We can consider, more generally, the module M_i as a function of the model and mesh:

$$M_i = M_i\,(m_i, d_i, n_i, \ell_i, \varphi_i, \gamma_i)$$

We will consider the frame module in broader sense of word and put:

$$M_i = \Pi\,(m \to m_i, d \to d_i, n \to n_i, \varphi_s \to \varphi_i, \gamma \to \gamma_i)\cdot M(m, d, n, \varphi, \gamma)$$

where Π is an operator of the transformation:

$$m \to m_i,\ d \to d_i,\ n \to n_i,\ \varphi \to \varphi_i,\ \gamma \to \gamma_i,$$

which contains, in particular, the operators of analytical transformation and ~~of~~ of transfer

The notion of frame module originates from the theory of the difference schemes and is intimately connected with (that) ~~the notion~~ of local difference operator (see p. 8.) ~~the frame module~~

The most important application of the frame module is connected with the transformation of coordinates and correspondingly with that of the difference scheme.

9.VII.83
10.VII.83

Let be

$Lu - f = 0$

(frame scheme) with constant coefficients,

which has constant coefficients in some initial coordinates (i.e. eulerian)

According to the transformation of the coordinates it becomes the scheme with variable coefficients and may be considered as the image of the frame scheme under the transformation

Thus, the frame scheme is a prototype of a class of equivalent schemes,

The frame prototype completely defines the type of boundary value problems

Note By appropriate choice of the coordinates a frame scheme with small parameters can be reduced to a regular one. In this case the small parameter is introduced into the transformation function.

22.VII.83. 11 ⟷ 16а.

11. изымается из [illegible] между 10 и 12 и переносится на 16а.

— 16а —

We will call tuning (or activating) a module the preliminary fixing all its input data [illegible] to get prepared the module to the further functioning as an autonomous model. These data can be obtained in many ways: through the direct transfer ~~of output data of another~~ of output data within the ~~system~~ modular system $\{M_\alpha\}$, through their intermediate transforming, by means of the matching module, by solving the set of equations for algebraical (non fixed) parameters.

Now let us formulate the conditions ~~requirements~~ on the algorithms and accordingly give their classification.

The stability ~~of the algorithm together with~~ an and consistence of the algorithm ~~enhance~~ ~~accordingly~~ as is well known guarantees its convergence

The verification ~~control~~ and realization of these conditions for an ~~the~~ autonomous simple algorithm is based on the fundamental theorems of convergence (see ... [ ])

The analysis of the algorithms, having the structure of a semigroup or of its step operator generates ~~gives rise to~~ ~~the~~ a dichotomy of the difference schemes.

These can be subdivided into the classes

i) absolutely or conditionally stable schemes

ii) absolutely or conditionally consistent ones

iii) explicit or implicit ones

The relations between these notions are formulated in the theorems as follows [ ]:

Theorem 1. Let

$$\frac{\partial u}{\partial t} = \mathcal{L}(\partial)\, u = \sum_{\alpha=0}^{\ell} A_\alpha \partial^\alpha u \qquad *$$

be a correct system with constant matrices A_α, ~~that is correct~~

then the explicit scheme

$$\frac{u^{n+1} - u^n}{\tau} = \mathcal{L}\left(\frac{\Delta}{h}\right) u^n, \qquad \Delta = \frac{\tilde{T}_1 - \tilde{T}_{-1}}{2}$$

is absolutely consistent with * conditionally stable and ~~correct~~ if the condition

$$\tau / h^{2p} \le \text{const}$$

holds. Here p is the order of $\mathcal{L}(\partial)$ and const a positive constant

Theorem 2

The implicit scheme

$$\frac{u^{n+1} - u^n}{\tau} = \mathcal{L}\left(\frac{\Delta}{h}\right) u^{n+1}$$

is absolutely consistent with * and absolutely stable

10.VII.83.

If: i) the differential operator $\mathcal{L}(\partial)$ in x has unbounded spectrum

ii) the difference operator

$$(T_0 - E)/\tau - \Lambda_0(T_1)$$

is absolutely consistent with the differential operator $\frac{\partial}{\partial t} - \mathcal{L}(\partial)$

Then: the explicit scheme

~~the explicit~~ $\frac{u^{n+1} - u^n}{\tau} - \Lambda_0(T_1)u^n$

cannot be absolutely stable

Theorem 3 states, that no explicit scheme can be simultaneously absstable and absconsistent

A scheme having both these properties ~~belongs to the class of~~ must be implicit.

~~Of~~ Theorems 1,2 use the correctness and the stability ~~understood~~ in the sense of Petrovsky.

It is to be noted, that for large classes of Cauchy problem and difference schemes (in the theorems 1-3) solution operators constitute the semigroups in some Banach spaces

For the effectiveness of the simple autonomous module essential is the number of operations in step operator per point.

We introduce into consideration a positive function $\varphi(N)$, that signifies the number of operations corresponding to the step operator of the module, containing N points.

Definition: If $\varphi(N)$ has

$$\varphi(N) \leq \text{const}\ N$$

where const is not dependent on N, then the scheme is locally economical (economical on the step operator).

If

$$n \cdot \varphi(N) \leq \text{const} \cdot \frac{N \cdot t}{\tau_a} = \text{const} \cdot \frac{N \cdot n\tau}{\tau_a}$$

see 20a.

where const is not dependent on N

τ_a – maximal time step, admitted by accuracy requirement

n – the number of time steps

then the scheme can be called globally economical or efficient (economical on the solution operator)

NB Перенести определение глобальной экономичности

The explicit scheme is always locally economical, with $c = m_e$, where ~~The implicit one dimensi~~ m_e is the number of operations for a point.

The one dimensional implicit scheme is locally economical too with $c = m_i$ where m_i is the number of operations, corresponding to one point.

Let τ_e^s denote the maximal timestep admitted by stability conditions τ^a – the maximal step, admitted by the accuracy requirement

then the ratio

$$K_e^i = \frac{m_e \cdot \tau_e^s \cdot \tau_e^a}{\tau_e^s \cdot \tau_e^a} : \frac{m_i}{\tau_i^s \cdot \tau_i^a} = \frac{m_e}{m_i} \times \frac{\tau_i^s \tau_i^a}{\tau_e^s \cdot \tau_e^a} +$$

approximately represents the relative efficiency of the implicit scheme as compared with explicit one

In the expression $\frac{\tau_i^s \tau_i^a}{\tau_e^s \tau_e^a}$ it is necessary to choose τ_i^a max for all admissible τ_i^s, and τ_e^a max for all admissible τ_e^s,

Then the expression

$$K_e^i = \frac{m_e}{m_i} \times \frac{\tau_i^s \cdot \tau_i^a}{\tau_e^s \cdot \tau_e^a}$$

approximately represents the relative efficiency of the implicit scheme as compared with explicit one

In the ratio

$$\frac{\tau_i^s \cdot \tau_i^a}{\tau_e^s \tau_e^a}$$

τ_i^a is to be taken as maximum value for all admissible τ_i^s,

τ_e^a - as the max value, for all admissible τ_e^s.

sol

20 VII 83

~~It is to be noted~~ Up to now we have ~~We~~ considered the effectiveness in fixed mesh. Applying the mesh transformation, we can increase the effectiveness of the algorithm due to ~~because of~~ the homogeneity ~~accuracy~~ of local ~~point~~ approximation accuracy and the increasing the maximum time step τ_s admitted by stability (see example and fig.)

~~If m denotes the number of operations per a point, τ_1 - time step for an explicit scheme, m_2, n_2~~

Let m_1, τ_1, m_2, n_2 denote the number of the operations and time step ~~correspondingly~~ for an explicit scheme ~~m_2, τ_2~~ correspondingly for absolutely stable implicit one. Then the implicit scheme is preferable to explicit one, ~~assume~~ if

$$\frac{\tau_2}{\tau_1} > \frac{m_2}{m_1} ?$$

assuming, that τ_2 satisfies to the accuracy requirement.

NB classitive dynamics $\varphi(N)$

Can use

(It is) possible in multidimensional case to construct simple ?
autonomous efficient algorithms

Generally speaking, the following statement
is true:

There is a decomposition of the multidimensional
implicit absolutely stable and consistent scheme
into the product of one dimensional schemes
that are implicit, abs. stable on each fractional step
and consistent on the whole step. ⇒ This decomposition
can be realized as the splitting up
scheme or as that by means of approximate
factorization

For the large classes of the differential
equations and the difference schemes
this statement is exactly exact
and can be considered
as a theorem (the theorem 4)
As the one dimensional implicit abs. stable
and absolutely consistent schemes are

efficent , the therem IV asserts, that there are multidimensional. efficent simple modules

There arises the question: Is the decomposition of (an) onedimensional implicit module into the set of onedimensional impicit modules of smaller size possible?

The following statement holds (see

Theorem V. The system of threediagonal equations with $N \times N$ matrix ($N = N_1 \cdot N_2$) admits the decomposition into $\leq N_1$ threediagonal subsystems, each with $N_2 \times N_2$ matrix. The matching module is a $N_1 \times N_1$ threediagonal system (see [ ]).

(сл. Кол. Эл. Бурдов сл. док.)

The result holds for recurrent-scalar and vector-relations, scalar and vector „sweeps" (progonkas)

The given analysis is related to the abstract (analytical) computational algorithms and abstract (analytical) modules.

The computer architecture ~~of the com~~ and the algorithm structure have not been taken into consideration. Notwithstanding already on this level the possibility and efficiency of the algorithm decomposition into onedimensional ones is evident

Note 1. Not every splitting along the coordinate directions generates the decomposition into purely onedimensional problems. F.e., when splitting a second order system with the mixed derivatives, the later are being taken from a lower level and fractional step modules are not onedimensional as regards the data sets. The full decomposition into onedimensional modules with onedimensien data sets can be realized in the following manner

i) by splitting along coordinate and diagonal directions (see..

ii) by ~~&~~ interpolation ~~the~~ the mixed derivatives from the values on ~~the~~ two adjacent coordinate lines to the ~~intermediate~~ middle line

~~iii)~~ It is to be noted that the interpolation operator can be considered as ~~supplementary~~ additional fractional step module or as a ~~matching~~ matching module.

Note 2

According to basic properties of the scheme (stability, consistence, explicitness, efficiency) a dichotomic classification can be introduced

Let us introduce the notations:

the symbol $S(1,1,1,1)$ ~~the symbol~~ denotes absolutely ~~stable~~ consistent, absolutely stable, implicit, locally economical scheme;

the symbol $S(1,1,1,0)$ signifies ~~the~~ abs. ~~stable~~ consistent, absolutely stable, implicit, local noneconomical scheme and so on.

→ 25a

theorems I–IV can be formulated in these terms in following manner:

Th. 1 $\exists \; S(1, 0\,0, 1)$

Th 2 $\exists \; S(1, 1, 1, x)$

Th 3 $\min \exists \; S(1, 1, 0, x)$ } $x = 0, 1$

Th 4 $\exists \; S(1, 1, 1, 1)$

this dichotomic classification can be enlarged by adding another properties (group properties, conservativness a.s.o)

the Note 3 For the quality and effectivness of the algorithm are very important stability and approximation control. For this aim functions and parameters of the algorithm have to be adaptive. For example, weight coefficients in a two level implicit scheme has to garantee the maximum of dissipation in the region of shock wave (shock layer) and maximum of accuracy

in the (region of) smooth solution

The scheme predictor-corrector is especially adaptive and garantees the reasonable ~~compromiss~~ agreement between stability and accuracy, predictor ensuring the stability, corrector doing that for accuracy. (see Yanenko)

Splitting up according to physical processes enhances for the stability and accuracy control.

F.e., splitting up diffusion process in a reagent medium, described by the equations

$$\frac{\partial u}{\partial t} = \Delta u + f(u)$$

into two alternatively operating subsystems

$$\frac{1}{2}\frac{\partial u}{\partial t} = \Delta u \quad \text{(diffusion process)}$$

$$\frac{1}{2}\frac{\partial u}{\partial t} = f(u) \quad \text{(reaction)}$$

enables one to integrate diffusion equation by means of noniterative implicit scheme and correspondingly the reaction equations by stiff equations method.

11.VII.83
12VII.83

The time steps and fractional schems are quite different.

The application of a homogeneous analytical representation, in particular, the uniform order of local approximation as well as constructing the adaptive mesh permits accuracy control of the algorithm.

The presence of the interior iterations in a step operator hinders the accuracy evaluation. Equally this is valid for interior iterations in overlapping matching.

5. The global algorithm

Now we will consider the real computational process and modules

In the case of the global algorithm, which represents a composition of the simple nonautonomous modules, efficiency of the algorithm depends strongly on the architecture of the computer, on the structure of the modular system and that of matching module in particular

Let us compare the efficiency of the explicit and implicit schemes for the global algorithm.

In the case of the explicit scheme the matching module consists simply in joining to the domain R_i the strip a_{ij} of neighbouring modules M_j (see fig...)

$a_i \in \sum_j a_{ij}$ where a_{ij} are adjacent strips, $a_{ij} \in R_{ij}$, R_{ij} are adjacent domains (see fig...) The data sets are taken for the same time level.

This matching can be realized by direct data exchange and obtaining the solution in R_i on the upper time level with $\bar{R}_i$ taken on the lower time level (see fig)

In the case of the explicit schemes the matching module completely conserves the analytical algorithm and leaves it unchanged

Thus the explicit matching causes no

~~[illegible]~~ information loss and can be considered as ~~[illegible]~~ the cybernetic one.

It is to be noted that in the case of explicit matching the global algorithm is flexible as regards the algorithm decomposition in general and the choice of the domains Ω_i (see fig...)

The negative property of the explicit ~~schemes~~ modules is the ~~rigidity~~ strictness of the stability criteria, especially for large problems and complex modular systems. For example, the ~~solution~~ profiles of the viscous flows with high Reynolds numbers ~~past the bodies~~, have large gradients ~~in the case~~ near the body. The condition of uniform local accuracy requests refining of the space mesh in the ~~region~~ boundary layer domain. ~~And as a consequence~~ According to the ~~of~~ Courant stability criteria for explicit schemes, time step $\tau_i = \tau_i(h_{min})$ becomes very small and (as the consequence) the difference scheme ~~is uneconomical~~ is not efficient. (see fig...)

The stability conditions (imposed by an implicit scheme) restrict less rigidly, if not at all, the step-size [of the implicit scheme], but the matching algorithm is more complex, than that for explicit one. Moreover, the global analytical algorithm changes because of (approximate factorization), iterations, interpolations (truncation error) solving the system of equations, pertaining to the matching module (round off error)

The analytical algorithm can lose the semigroup structure, because of additional interior nonlinearity iterations

It is to be noted, that for the module (synthesis with overlapping of neighbor domains) ~~mass~~

In the case of module sinthesis with overlapping of neighbors domains, the algorithm loses semigroupstructure too, and this is true both for implicit and explicit schemes

~~In many cases~~ ~~In the case~~ If the application of complete directional splitting ap is possible, the matching module is reduced to the ~~standard operation~~ transposing the matrix and/or to the agle braical sweeps (see fig...)

Efficiency of the ~~the~~ implicit algorithms heighgens and (attains) ~~attains~~ the maximum for the homogenous modular ~~struct~~ system

6 Homogenous modular System

The programm, ~~XXXXXXX~~ that has good modular structure, can be effectively realized on the parallel computers. It is especially done in the case of the homogenous modular ~~structure~~ system

Definition.

the set of homogenous simple modules $M_i(N_i)$, $i=1..p$ (N_i - the number of points in modules' mesh ~~informational medium~~) constitute the homogenous modular system, if the following conditions are fulfilled

i) $N_1 = N_2 \ldots = N_p$

ii) The continuous models M_i and discrete schemes ~~d_i~~ pertaining to the modules $\{M_i\}(i=1..p)$ are equivalent, i.e. become identical for the equal coefficient functions. The same is valid for discrete schemes

iii) the meshes are equivalent under differentiable one-to-one transformation with controllable condition number

If the exchange $\{\Gamma_i\}$, and tuning $\{\Pi\}$ and boundary values modules constitute homogeneous systems

we call the homogeneous modular system the homogeneous modular structure (see fig...)

NB: Замечание от 83 на 85, 85 и 88

In scalar computers the global computational step is reduced to the s cycle

$$(M_1, E_1)\ (M_2 E_2) \ldots (M_p E_p) =$$

$$= (M T_1, E_1) \cdot (M T_2, E_2) \ldots (M T_p, E_p) =$$

$$= M(T_1, E_1) \cdot M(T_2, E_2) \ldots M(T_p, E_p)$$

For parallel computers this cycle can be realized by parallel functioning of the modules $M_1 \ldots M_p$ in SIMD type

i) $N_1 = N_2 \ldots = N_p$

ii) the continuous models M_i, pertaining to the modules $\{M_i\}$ $(i = 1 \ldots p)$ are equivalent, i.e. become identical for the equal coefficient functions. The same is valid for discrete schemes di

iii) The meshes are equivalent, i.e. become identical under piecewise smooth one to one transformation.

The condition number of the transformation must be controlled in order the machine implementation be possible.

If the sets of the exchange, tuning and boundary value modules constitute separatedly (each for itself) homogenous systems, we call the homogenous modular system the homogenous modular structure

(see fig. ...)

For scalar computers the global computation process can be reduced to the cycle

$$(M_1, E_1)\,(M_2, E_2)\,\ldots\,(M_p, E_p) =$$
$$= (MT_1, E_1)(MT_2, E_2)\,\ldots\,(MT_p, E_p)$$
$$= M(T_1, E_1)\cdot M_2(T_2, E_2)\,\ldots\,M(T_p, E_p)$$

For parallel computers this cycle can be realized by concurrent implementation of the modules $M_1 \ldots M_p$ in SIMD type computing.

13.VII.83

If N_i are not equal, the macrocodes of M_i ~~is invariant~~ are essentially the same, but the input parameters of M_i are different.

In this case the homogenization of the modular structure is necessary.

§ For some cases of problems it can be realized, for example, by the method of ~~fictions~~ fictitious domains (MFD) (see Konovalov)

ce)

7 The effectivness of the global algorithm

The presence of the transfer exchange operators complicates the global algorithm. Therefore the minimization of transfer operations operators is desirable. The optimal minimization is possible in directional splitting up. In this case the modules become one-dimensional and autonomous, transfer operator is reduced to matrix transposing. But such an optimum can be realized in most simple particular, although typical cases of splitting up f.e. when differential system contains only derivatives along the coordinate axes and no mixed ones. (see fig.)

In general case, besides the matrix transposing, it is necessary to introduce exchange operators between neighbour modules with the to in order to express mixed derivatives through the pure ones by means of interpolation. → see 359

Another approach is to compute the mixed derivatives on the lower time level t_n / on the middle line and consider them as the right side to obtain the values of the pure derivatives on the upper time level t_{n+1}.

Interpolation operator can be considered as a kind of explicit scheme and in this interpolation it can be reduced to the step operator, which can be flexibly decomposed The idea to realize fractional step method splitting up on parallel processors computers was told in the paper (see [March. Yan.]) 70-80 years seventies In Western literature publications the utilizing of the ADI methods for supercomputers became generally accepted. in seventies. In much more general More generally one can span about realization of fractional step method on parallel computers To this aim the through classification and evaluation of the efficiency of the fractional step schemes on parallel computers becomes the necessity the econometrics of the global algorithm is not yet developed. Nonetheless already reliable now there are qualitative evaluations

13.VII.83

Of the different MFS-schemes

xxx

When realizing the algorithms on parallel computers the MFS schemes which are analytically equivalent, become nonequivalent

F.e., the heat equation

$$\frac{\partial}{\partial t} = \Delta u, \qquad \Delta = \sum_{i=1}^{m} \frac{\partial^2}{\partial x_i^2} \qquad m = 2,$$

has two analytically equivalent schemes

the alternating direction schemes [P.R]

(bidirectional splitting up along coordinate) scheme with weights [illegible]

but their realizations on parallel computer are different. In the first case matching module reduces to matrix transposition, in the second case the additional exchange operators between every three adjacent modules are necessary (see fig)

cc) In either both algorithms have the modular

? structure (homogeneous) and in this algorithm

the step fractional modules

ce) are identical (see the fig...)

13.VII.83.

It is to be noted that for $n=3$ (and A2 I) splitting up with weights schemes are even algebraically nonequivalent (the first scheme is absolutely stable and consistent, the second one is conditionally consistent and absolutely stable)

The A2 I scheme (m2s) was utilized in a parallel computer A2 I NA (see Nogi)

It is clear, that corresponding splitting up scheme is more effective. Both algorithms can be utilized in SIMD structure

The stabilizing correction scheme has nonequivalent fractional step modules and therefore cannot be realized in SIMD structure. (see [2])

Another examples and general considerations permit to conclude, that the splitting up schemes are most effective in real computational algorithm. They obtain the minimal data exchange, maximal homogenity and simplest matching module

Some very important examples show, that

efficiency ratio $\frac{T_{exp}}{T_{imp}}$ evaluated on the schemes of complete directional splitting up is of the same order for scalar and parallel computers

cc) In the paper [see Nogi] it was shown on the probe computer ASIVA with N parallel processor and for NxN global matrix that the coefficient of parallelism is approximately equal to 1.

By its structure ASIVA is array computer with buffer memory and universal switching network

Another approach can be proposed.

NII Mpro puter

Every one from N processors is a bundle of elementary vector processors, functioning in pipe line and sharing mode, and constitutes a functional unit, that realizes vector recurrent relation

13. VII.83 — 340 —

As the ~~realization of~~ one dimensional implicit scheme for the differential equations of mechanics is ~~realized~~ computed by ~~means~~ of vector sweeps (vector recurrent relations), every sweep can be essentially speeded up by ~~combining~~ the pipe line and chaining

14. VII.83 5.30

We shall try to give asymptotical evaluation ~~of the~~ for ~~the coefficient~~ rate ~~of par~~ the parallelism coefficient of MFS schemes

? Let: p_1 be the ~~number of partition number~~ number of subdivisions corresponding to the flow of scalar arithmetical operations ~~corresponding~~ in pipeline

p_2 – the additional gain of speed in the bundle of vector pipe line processes, due to the chaining

Then the overall gain in speed of the ~~chained~~ bundle of chained vector processors is equal approximately to

$$p = p_1 \cdot p_2$$

If the number of the bundles is N_2

where N_2 is the number of rows, the gain in speed on the first fractional step is equal to $p \cdot N_2$. The loss of the speed Tc The ratio For the first fr. st. the ratio

$$\frac{T_{imp}}{T_{exp}}$$

has the order of M is equal to where M is the number of operations for inversion the matrix of vector components in nonlinear relation.

Taking into account that M is independent of N_1, N_2, we get to the conclusion, that the global module on the first fract. steps is effective

The transition from the first fractional step (x_1-direction) to the second one (x_2-direction) is realized by transposition of the global $N_1 \times N_2$ matrix.

(analogous situation holds for 3D case)

The utilizing ~~the switching commutation~~ network and

cc| buffer memory was proposed in the paper (see [Ref.])

For scalar sweeps the switching time is ~~not~~ ~~large~~ (negligible) and the parallelism coefficient is near to 1.

The system of bundle vector processors

by can ~~not~~ be transformed to ~~that of array processors~~ array computer with additional speed gain.

For this aim every pipeline processor of the bundle ~~must~~ should be segmented into ~~micro~~ micro processors, the processor bundle is segmented correspondingly into the ~~system~~ bundle of the ~~micro~~ micro processors. This ~~is~~ structure ~~can allow~~ permits the application of

cc algebraical sweeps (see ...)

If q - the number of points in ~~each~~ each segment, $\frac{N}{p}$ - ~~the number~~ that of segments, the gain of speed due to the segmentation is proportional (to) $\frac{N}{p}$.

The overall gain of sped is proportional to the number of microprocessors and is represented by the ratio

$$K \frac{N^2 \cdot P}{9 \cdot P}$$

where K is a constant, independent of N, q, P and connected with the algebraical sweeps and switching time

As a final conclusion we can state that the efficiency relation between explicit and implicit schemes holds for scalar, vector and array processors to within an universal constant

8 heterogeneous modular systems

Now we will consider the examples and types of the heterogeneous modular structures, which can be realized on SIMD computers with relatively small coefficient of parallelism or on MIMD ones.

~~The negative property of the algorithm, that~~

The ~~nonhomogeneity~~ heterogenity of the global algorithm is ~~connected with~~ a negative property that has as a consequence the ~~small~~ diminishing of the ~~coefficient~~ parallelism coefficient

~~the diminishing~~ the ~~nonhomogeneity~~ heterogenity of the modular structure and the ~~diminishing~~ lowering of the ~~parallelism~~ coefficient.

~~We show to~~ We will discuss some ~~kinds~~ types of the ~~nonhomogeneity~~ heterogenity

i) the ~~nonhomogeneity~~ heterogenity of the continuous and/or discrete model

When splitting up ~~on~~ according to physical processes ~~the splitting~~ by the ~~fractional step~~ approximated models ~~of~~ on the ~~fractional steps~~ are essentially different and as a consequence ~~Consequence~~ the corresponding analytical algorithms

This ~~nonhomogeneity~~ heterogenity requires quite different macrocodes, and can be realized only on a MIMD computer

The typical example of this situation is splitting up the systems of equations governing the behaviour of reagent gas

$$\frac{\partial w^i}{\partial t} = \frac{\partial \Sigma^{i\alpha}\left(w, \frac{\partial w}{\partial x}\right)}{\partial x^\alpha} + F^i(w).$$

On the first fractional step we have the system of gazdynamical equations

$$\frac{1}{2}\frac{\partial w^i}{\partial t} = \frac{\partial \Sigma^{i\alpha}\left(w, \frac{\partial w}{\partial x}\right)}{\partial x^\alpha}$$

on the second fractional step we have to integrate the system of ordinary differential equations

$$\frac{1}{2}\frac{\partial w^i}{\partial t} = F^i(w) \qquad *$$

is to be integrated.

The last one is typically stiff one problem

The system * is in many cases stiff.

Another example gives us the decomposition of the steady flow past body problem. For the transonic flow the ideal gas the domain of integration is subdivided into two domains Ω_1, Ω_2) Ω_1, Ω_2.

In Ω_1 the domain where the flow is subsonic algorithm on iteration stabilizing

Stationarization is applied

in the domain II of ~~steady~~ subsonic flow the marching method is applied (see fig)

Traditionally, when considering the viscous flow past the body, two domains are I, II introduced In the domain I gaz is considered as frictionless and the Euler's equations are to be integrated, in the domain II the equations of boundary layer hold. The dividing surface (line) is obtained by iterations (see fig)

ii) Parametric heterogeneity

This kind of heterogeneity is typical for complicated domains. In this case even if the model is uniform the directional splitting up generates two one dimensional segments with different numbers of points or different analytical units (not uniform see fig.)

iii) logical ~~inhomogenity~~ heterogenity
If the algorithm has nonlinear branching the realizing program has logical loops, the ~~that~~ chaining should ~~must~~ be of variable structure. For the MFS algorithms ~~the~~ computer is ~~no more~~ ~~and~~ ~~of~~ MIMD computer is necessary

iv) The most typical example of a hetero ~~inhomo~~-geneous modular structure is the decomposition of a MFE algorithm into physical modules (~~method of~~ substructure or superelement method). In this case the integration domains $\Omega_i \in M_i$ may have topologically nonequivalent meshes, or the solutions ~~may~~ may have quite different analytical representations
As a consequence, the matching module becomes complex, method of algebraically sweeps should be replaced by more general, but not very efferent

method of elimination

A NFE algorithm with a physical decomposition can be realized on a SIMD computer with a small parallelism coefficient or on MIMD computers.

There is possibility for another algorithmic realization which is connected with the decomposition of global linear algebraical system.

In this case the method of substructures can be applied two, but the last ones are not algebraical substructures

In this case the method of algebraical substructures can be applied. (see...)

V/ For a system of ordinary differential equations the decomposition of integration algorithm can be reduced to the splitting up of the right sides (see...)

The resulting subsystems are essentially different and the computer realization is only possible with a rather small coefficient of parallelism.

VI) Systems of linear network equations (see [B]) can be reduced by means of algebraical sweeps to a parametrically heterogeneous modular structure, ~~consisting of~~ which consists of linear finite dimensional modules and its matching module constitutes a system of Kirchhoff's relations (see fig...)

14.VII.83 -51-

Homogenization of heterogeneous modular structures

According to our classification of heterogeneity, we will consider the means of the homogenization.

i) In the case of a model heterogeneity the homogenization can be realized by means of a generating model, i.e. the model which generates as private cases, the models m_i, pertaining to M_i. For example, the system of Navier Stokes equations, generates as particular cases, the equations of an ideal gasflow those of boundary layer detachment, reattachement flows, jets, wakes, coherent structures a.s.o. The homogenization can be reached if heterogeneous modular system $m_i \in M_i$ of simple models is approximated everywhere, for each $i = 1..p$ by the generating model, and a homogenous system of modules is constructed on its base.

This homogenization may enlarge the number of arithmetic operations in the global algorithm, but the number of basic modules diminishes, and the programm becomes more clear and simple.

It is to be supposed that enhancing the speed of parallel computers will stimulate the utilizing the generating models, because of simplifying the macrolanguage and programming technology.

ii) In the case of parametrical heterogeneity there are several approaches to the homogenization problem.

MFD allows to complement an arbitrary domain Ω to the rectangular one $\Pi \supset \Omega$

MFD permits to in

In a MFD method an arbitrary integration domain Ω is included in some simple domain $\Pi \supset \Omega$, admitting the regular mesh, and

the boundary conditions are transferred to $\sqrt{\Pi}$

Overmore the coefficients in Π are accordingly ε-approximated (see [Kun])

cc/ func! All these transformations allow to approximate a ~~inhomogenous~~ heterogenous structure $M_i[\Omega_i(N_i), A_i]$ by homogenous one $M_i[\Omega_i(N), A_i^\varepsilon]$, where ~~$A_i(N)$~~ A every domain $\Omega_i(N)$ has ~~the same~~ an equal number of points N and algorithms A_i^ε can be represented in the form

$$A_i = A B_i$$

Here B_i is a tuning operator and A is a pattern module, ~~the same~~ one for all simple modules M_i

Coefficient functions in MFD are singular

~~coordinate linear maps~~

and their computation is connected with a linear branching

~~A~~ Another approach to ~~a~~ treat ~~ing~~ imp the boundary value problems with arbitrary

domain is an appropiate choice of coordinates permitting an unifred regular mesh (see fog....)

iii) ~~Domain~~ Nonlinear branching, lowering the parallelism coefficient can be eliminated through approximating the algorithm with the branching by that without the one.

For example the scheme of Lettez, which is good for discerning between the waves of rarefaction and those of compression, can be xplaced by uniform scheme of Neumann-Richtmeyr

(see fog...)

Linear branching (conditions), remaining in linearized algorithm, can be included in homogenous analytical algorithm as additional - logical - functions (example)

In this way ~~one can § construct~~ a mixed analytical-logical homogenous algorithm can be constructed. (see example ../ of the algorithm

The cited approaches ~~to~~ the homogenization ~~problem~~ do not cover all possibilities. and the preceding problem and the homogenization ~~remains~~ plays essential role ~~constitutes the essential part of the~~ for the theory of effective algorithms ~~for~~ on parallel ~~process~~ computers. of homogenization The effective homogenization ~~allows~~ reduces the remains open number of base modules, ~~allows~~ heightens the parallelism coefficient, diminishes the ~~the~~ ~~change~~ number of logical reconfigurations. ~~Strictly~~ In strict meaning the word one can speak say about talk homogenous modular structure ~~in the frame~~ only for a set of the modules of the same dimension.

M correctne → correction

15.VII.83. -56-

F.S.M allows in combination with M.F.D and other another approaches to homogenization to reduce ~~the~~ the set of multidimensional modules to ~~one~~ that of onedimensional and to minimize in such way the module basis. A special ~~place~~ takes place the belongs to modules of boundary conditions

For all the diversity ~~variety~~ of boundary conditions it is possible to represent the step operator as a product of two operators:

2) ~~B~~ step operator of the first boundary problem

b) ~~operator~~ of correction operator of boundary conditions (see ... Con.

? This allows representation to ~~standardize~~ bring to a standard form an arbitrary boundary value problem, expressing its peculiarity through the correction operator of boundary conditions

10. Conclusion

From the above presented ~~arguing~~ the preceding presentation the following conclusions can be drawn:

i) the modern computers stressed the growing importance of the cybernetic algorithm. For the complete evaluation of the global algorithm efficiency the real time needed for executing the cybernetical algorithm should be taken into account.

ii) For the parallel computers the relative efficiency between implicit and explicit schemes remains principally unchanged as compared to scalar computers.

For many important problems, the implicit schemes conserve their principal advantages. It is true especially for dynamic meteorology, stationary flows pest & bodies etc.

iii) The modular structure of the algorithm became the necessary condition of efficient parallel computations.

2. The ~~non~~ inhomogeneity of If the modular structure is heterogeneous ~~inhomogeneous~~, the value of parallelism coefficient diminishes.

The maximum of parallelism coefficient is achieved on splitting up schemes when the modular structure has a model and parametrical homogeneity

iv) Homogenization - full or partly - of the modular structure can be obtained by several approaches:

α) the method of generating model

β) the method of fictitious domains

γ) the application of adaptive meshes

v) Approximation control is facilitated under following conditions

i) homogeneous analytical & representation of the solution

ii) ~~mesh distribution~~ adaptive mesh ✓ securing the ~~local~~ accuracy uniform distribution of the local truncation error.

iii) The absence of inner iterations

6 (VII) the effectively evaluation of a real global algorithm (econometrics) becomes the important branch of computational mathematics, whose development is at the very beginning