

хронизацию момента запуска с тактовой частотой умножитель выполняет перемножение 32-разрядных чисел за 2,2 мкс, а 16-разрядных за 1,3 мкс.

Автор выражает глубокую признательность канд. техн. наук А. Н. Касперовичу и канд. техн. наук И. И. Коршеверу за ценные замечания, сделанные при чтении рукописи.

#### ЛИТЕРАТУРА

1. Карцев М. А. Арифметика цифровых машин. М., «Наука», 1969, с. 470—481.
2. Дроздов Е. А., Комарницкий В. А., Пятибратов А. П. Многопрограммные цифровые вычислительные машины. М., Воениздат, 1974, с. 250—259.
3. Уэйзер, Питерсон. Быстродействующий цифровой умножитель для обработки сигналов в реальном времени.— «Электроника» (США), 1977, т. 50, № 20, с. 40—49.
4. Каган Б. М., Каневский М. М. Цифровые вычислительные машины и системы. М., «Энергия», 1973, с. 339.
5. Мик, Спрингер. Однокристалльный умножитель в цифровых процессах.— «Электроника» (США), 1976, № 10, с. 47—54.
6. Мик. Цифровой умножитель для совместной работы с ЦП.— «Электроника» (США), 1977, т. 50, № 20, с. 75—76.
7. Habibi A., Wintz P. A. Fast multipliers.— "IEEE Trans. on Computers", 1970, vol. C-19, N 2.
8. The TTL Data Book for Design Engineers. Second Edition. USA, Texas Instruments Inc., 1976, p. 7. 420—7.423; p. 7.391.
9. Sapowski Goseph J. The matrix transform processor.— "IEEE Trans. on Computers", 1976, vol. C-25, N 7, p. 703—712.
10. Stenzel W. J., Kubitz W. J., Garcia G. H. A compact high— speed parallel multiplication scheme.— "IEEE Trans. on Computers", 1977, vol. 26, N 10, p. 948—957.
11. Паразурмен. Аппаратное обеспечение микропроцессоров.— «Экспресс-информация. Приборы и элементы автоматки и вычислительной техники», М., ВИНТИ, 1977, № 46, с. 31—42.

Поступила в редакцию 20 октября 1978 г.;  
окончательный вариант — 3 января 1979 г.

УДК 519.853.6+681.3 : 621.38

А. С. ЗАГОРУЙКО

(Новосибирск)

### РЕЗУЛЬТАТЫ СРАВНЕНИЯ РЕЛАКСАЦИОННЫХ МЕТОДОВ МНОГОМЕРНОЙ МИНИМИЗАЦИИ НА ЭВМ

**Постановка задачи.** В настоящее время существует большое разнообразие методов многомерной минимизации. Применение их к конкретной задаче ведет к большим или меньшим затратам машинного времени для достижения заданной точности решения. Некоторые методы могут оказаться совсем неработоспособными в применении к каким-то задачам, хотя по отношению к другим задачам они показывают хорошие результаты. Существенно сказывается также программная реализация метода для ЭВМ. Поэтому в тех случаях, когда необходимо решать задачи оптимизации с широким набором различных функций, как это имеет место в машинном проектировании электронных схем, целесообразно пользоваться не одним методом, а целой совокупностью их. При этом возникает задача сравнения различных программ минимизации.

В работе сравниваются разработанные программы, реализующие различные алгоритмы методов многомерной минимизации, с целью уста-

новления среди них иерархии по условному показателю качества программ. Это позволит обоснованно подходить к выбору одного наиболее подходящего алгоритма или переходить от алгоритма к алгоритму автоматически (а также по требованию пользователя) в определенном порядке.

**Краткая характеристика программ.** Все программы написаны на языке ФОРТРАН и эксплуатируются на ЭВМ АСВТ М4030. Использование принципа модульности их построения позволило довольно просто реализовать на основе разработки основных подпрограмм (модулей) ряд алгоритмов минимизации.

**Общие модули.** Рассматриваются задачи, в которых необходимо найти с требуемой точностью  $E$  минимум скалярной функции  $\Phi(x)$ , заданной в евклидовом пространстве  $E_n$  на  $n$ -мерном гиперпараллелепипеде  $Q$ :

$$x \in Q \subset E_n, \quad Q = \{x: \underline{p}_i \leq x \leq \bar{p}_i, \quad -\infty \leq \underline{p}_i \leq \bar{p}_i \leq \infty, \quad i = 1 \div n\}.$$

Функция  $\Phi(x)$  задается в модуле KSEDI\* в явном виде или определяется численно как норма вектора невязок выходных характеристик электронных и других систем, работа которых описывается нелинейными дифференциальными и алгебротрансцендентными уравнениями. В последнем случае применялся упрощенный вариант пакета [1] с решением нелинейных алгебротрансцендентных уравнений обобщенным методом хорд (модуль KNEAL) [2] и упомянутая подпрограмма KSEDI содержала в зависимости от вида электронной схемы до двадцати модулей. Соответственно весь комплекс программ при этом был рассчитан на 180 кбайт (в обычном случае — от 55 до 66 кбайт).

Численная информация (количество  $n$  оптимизируемых параметров — максимально до 20, пределы  $\underline{p}$ ,  $\bar{p}$  допустимого изменения их, точность  $E$  определения минимума  $\Phi(x)$ , начальная точка  $x$  и некоторые другие величины) вводится в модуле KBXODA. Здесь же в случае неявного задания  $\Phi(x)$  модулем KBVOD вводятся численные параметры [1], необходимые для анализа оптимизируемой системы, в результате которого определяется  $\Phi(x)$ .

Для таких систем градиент, а тем более гессиан в явном виде определить очень трудно или вообще невозможно. Поэтому были рассмотрены только алгоритмы с численным вычислением производных или без их вычисления.

В общем случае релаксационные методы спуска сводятся к построению итерационной последовательности вида

$$x^{k+1} = x^k + h_k S^k, \quad (1)$$

где  $x^k$ ,  $x^{k+1}$  —  $n$ -мерные векторы (точки)  $k$ -го и  $(k+1)$ -го приближений, причем

$$\Phi(x^{k+1}) < \Phi(x^k); \quad (2)$$

$S^k$  — вектор направления спуска;  $h_k$  — скаляр, называемый шаговым множителем.

Шаг  $h_k$  для всех рассмотренных программ минимизации определяется с помощью модуля КООРТ ( $S^k$ ,  $\underline{p}$ ,  $x^{k+1}$ ,  $x^k$ ,  $\bar{p}$ ,  $p_{\min}$ ,  $p_{\max}$ ,  $n$ ,  $z$ ) одномерной минимизации функции  $z = \bar{\Phi}(h_k) = \Phi(x^k + h_k S^k)$ , реализующего алгоритм золотого сечения [3]. Скалярные величины  $p_{\min}$ ,  $p_{\max}$ , характеризующие границы одномерного поиска шага  $h_k$ , вычисляются в модуле КММА ( $S^k$ ,  $\underline{p}$ ,  $x^k$ ,  $\bar{p}$ ,  $p_{\min}$ ,  $p_{\max}$ ,  $n$ ) из условий отстояния ком-

\* Передача необходимых для работы подпрограммы переменных везде осуществляется или через ее параметры (указываются при первом ее упоминании), или через фортранский оператор спецификации COMMON.

понент вектора  $x^k$  от границ  $\underline{p}$  и  $\overline{p}$  и величины и знака соответствующих компонент  $S^k$ . В результате работы программы величины  $p_{\min}$  и  $p_{\max}$  определяются так, чтобы в процессе поиска перекрывались все диапазоны изменения компонент  $x^k$ , причем перекрытие хотя бы одного из них было бы точным без остатка. При этом всегда с заданной точностью  $E$  должно выполняться условие (2). В противном случае пределы  $h_k$  определенным образом варьируются. Это в ряде случаев может привести к увеличению затрат машинного времени, но зато повышает надежность вычислений. В программе предусмотрена возможность окончания поиска и по достижению заданного количества  $K_M$  обращений к процедуре KSEDI.

Этой же цели, т. е. повышению надежности получения результата, служит модуль KSCN ( $x_i^{k+1}, \overline{p}_i, \underline{p}_i$ ), реализующий один из алгоритмов внешних штрафных функций [3] и работающий в случае необходимости как резерв перед обращением к модулю KSEDI.

Отметим, что некоторые константы, необходимые для надежной работы указанных модулей, были получены эмпирически в результате предварительных машинных экспериментов и затем встроены в эти программы.

*Модули, реализующие алгоритмы минимизации.* Эти модули отличаются различным выбором направления спуска  $S^k$ .

**ДФ.** Этот модуль реализует алгоритм [4] метода Давидона—Флетчера—Пауэлла и включает модуль KGRAD, определяющий численно антиградиент ( $-g^k$ ) минимизируемой функции  $\Phi(x)$  в точке  $x^k$  и модуль КМИМ ( $A, B, C, M, N, L$ ) перемножения массива  $A$  на массив  $B$  ( $A, B, C$  — матрицы;  $M, N, L$  — их размеры)\*. С помощью этих процедур происходит формирование вектора  $S^k$ .

Так как используемый метод приводит к накоплению ошибок, в программе предусмотрено через задаваемое количество  $J_{\max}$  итераций «обновление» накопленной информации (текущая матрица, определяющая вектор  $S^k$ , заменяется единичной матрицей). Если  $J_{\max} = 1$ , то данный модуль будет реализовывать алгоритм наискорейшего спуска НАСП. И если к тому же  $K_M = 1$  и задан параметр  $Y$ , сужающий диапазон одновременного поиска, то НАСП вырождается в градиентный метод.

Требуемая память в случае явного задания функции на весь комплекс программ с этим модулем — 66 кбайт.

**СОПР1, СОПР2, СОПР3.** При помощи модулей KGRAD и КМИМ формируется новый вектор  $S^k$ :

$$S^k = \begin{cases} -g^k & \text{при } k = 0; \\ -g^k + \alpha_k S^{k-1} & \text{при } k > 0, \end{cases}$$

где скаляр  $\alpha_k$  в различных модификациях программ, реализующих метод сопряженных градиентов, определяется соответственно, как в [3—5]. Здесь также через  $J_{\max}$  итераций происходит «обновление» расчетов. Требуемая память — 55 кбайт.

**РОЗ.** В основу программы взят метод поворота системы координат Розенброка [5]. Вначале формируется единичная  $n \times n$ -матрица направлений. Далее производится последовательно с помощью модулей КМИМ и КООРТ  $n$  спусков по этим направлениям, что соответствует методу покоординатного спуска. Из  $n$  приращений  $\Delta x^k$  переменных образуется система  $n$  векторов:

$$A_i = \sum_{k=i}^n \Delta x^k, \quad i = \overline{1, n}.$$

\* С целью сокращения времени обращения к массивам везде в программах матрицы представлены в виде одномерных массивов.

Условия экспериментов

Номер эксперимента	$\Phi$	$E$	$p$	$x^0$	$\bar{p}$
1	$\Phi_1$	$10^{-4}$	(-1, -1)	(3, 4)	(4, 4)
2	$\Phi_1$	$10^{-12}$	(-1, -1)	(3, 4)	(4, 4)
3	$\Phi_2$	$10^{-4}$	(-5, -5, -5, -5)	(2, 3, -4, 5, )	(5, 5, 5, 5)
4	$\Phi_2$	$10^{-12}$	(-5, -5, -5, -5)	(2, 3, -4, 5)	(5, 5, 5, 5)
5	$\Phi_3$	$10^{-4}$	(-1, -1)	(3, 4)	(4, 4)
6	$\Phi_3$	$10^{-12}$	(-1, -1)	(3, 4)	(4, 4)
7	$\Phi_4$	$10^{-4}$	(-4, -4, -4, -4)	(-1, -2, -3, -4)	(4, 4, 4, 4)
8	$\Phi_4$	$10^{-12}$	(-4, -4, -4, -4)	(-1, -2, -3, -4)	(4, 4, 4, 4)
9	$\Phi_5$	$10^{-4}$	(-3, -3)	(-1, 0)	(3, 3)
10	$\Phi_5$	$10^{-12}$	(-3, -3)	(-1, 0)	(3, 3)
11	$\Phi_6$	$10^{-4}$	(-100, -100, -100)	(-5, 4, 2)	(100, 100, 100)
12	$\Phi_6$	$10^{-12}$	(-100, -100, -100)	(-5, 4, 2)	(100, 100, 100)

С помощью модуля KSKS — скалярного умножения двух векторов, взятых из соответствующей матрицы, — реализуется процедура ортогонализации Грама — Шмидта [6]. После нормировки получается  $n$  новых ортогональных направлений, по которым осуществляется спуск и т. д.

Если «обновление» расчетов производить через  $J_{\max} = 1$  итераций, то данный модуль будет реализовывать циклический покоординатный спуск ПКСП. Требуемая память — 61 кбайт.

**ММП.** В основе программы лежит алгоритм модифицированного метода Пауэлла [4]. Как и в программе РОЗ, вначале производится покоординатный спуск последовательно по всем координатным осям. Но в отличие от РОЗ в каждой итерации обновляется не вся матрица направлений, а только один ее столбец (одно направление).

Из особенностей данной программы отметим процедуру формирования вспомогательного целочисленного массива  $I_p$ , указывающего в соответствующей матрице последовательность направлений, по которым должно быть произведено  $n$  спусков в каждой итерации алгоритма. Благодаря этому элементы матрицы в каждой итерации не сдвигаются, а обращение к направлениям (столбцам) происходит через массив  $I_p$ . Требуемая память — 59 кбайт.

**Условия численных экспериментов.** Эти условия приведены в табл. 1. Рассматривались функции трех типов [4, 7].

1. Квадратичные функции с хорошими линиями уровней:

$$\Phi_1 = (1 - x_1)^2 + (x_1 - x_2)^2;$$

$$\Phi_2 = (1 - x_1)^2 + (x_1 - x_2)^2 + (x_2 - x_3)^2 + (x_3 - x_4)^2.$$

2. Квадратичные функции с вытянутыми линиями уровней (с «оврагом»):

$$\Phi_3 = (x_1 - 1)^2 + 100(x_1 - x_2)^2;$$

$$\Phi_4 = (x_1 - 1)^2 + 10(x_2 - 1)^2 + 100(x_3 - 1)^2 + 1000(x_4 - 1)^2.$$

3. Неквадратичные функции.

Функция Розенброка [1]:  $\Phi_5 = (1 - x_1)^2 + 100(x_1^2 - x_2)^2;$

$$\Phi_6 = (x_1 x_2 x_3 - 1)^2 + 5[x_3(x_1 + x_2) - 2]^2 + 2(x_1 + x_2 + x_3 - 3)^2.$$

В табл. 2 представлены в качестве примера некоторые результаты экспериментов для неквадратичных функций. При этом для каждой программы минимизации приведены три показателя:  $E_g$  — достигнутая точность определения минимума функции; КФ — количество вычислений

Таблица 2

## Результаты экспериментов для неквадратичных функций

Программа (алгоритм)		Номер эксперимента			
		9	10	11	12
НАСП	$E_d$ КФ $t, c$	$2,1 \cdot 10^{-1}$ 442	$7,8 \cdot 10^{-4}$ 738	$5,8 \cdot 10^{-1}$ 1280	$9,8 \cdot 10^{-4}$ 2017
ПКСП	»	$5,5 \cdot 10^{-1}$ 70	$9,0 \cdot 10^{-2}$ 4745	$2,5 \cdot 10^{-1}$ 98	$1,6 \cdot 10^{-1}$ 3084
СОПР1	»	$6,1 \cdot 10^{-1}$ 59	$5,2 \cdot 10^{-5}$ 1572	$1,7 \cdot 10^{-2}$ 3291	$9,0 \cdot 10^{-4}$ 6696
СОПР2	»	$5,2 \cdot 10^{-3}$ 311	$4,1 \cdot 10^{-6}$ 793	$3,2 \cdot 10^{-4}$ 571	$3,9 \cdot 10^{-5}$ 1429
СОПР3	»	$7,9 \cdot 10^{-5}$ 435 27,0	$9,7 \cdot 10^{-8}$ 775	$7,3 \cdot 10^{-5}$ 1273 63,1	$6,4 \cdot 10^{-4}$ 1143
ДФП	»	$8,9 \cdot 10^{-2}$ 210	$2,1 \cdot 10^{-5}$ 506	$4,3 \cdot 10^{-5}$ 537 21,3	$1,0 \cdot 10^{-6}$ 2551
РОЗ	»	$6,8 \cdot 10^{-5}$ 1389 33,6	$7,2 \cdot 10^{-13}$ 1166 19,3	$4,1 \cdot 10^{-4}$ 818	$9,1 \cdot 10^{-8}$ 2804
ММП	»	$9,6 \cdot 10^{-5}$ 541 18,2	$7,2 \cdot 10^{-13}$ 817 15,5	$1,9 \cdot 10^{-1}$ 165	$8,1 \cdot 10^{-13}$ 3005 62,3

функции (обращений к процедуре KSEDI);  $t$  — машинное время решения задачи в секундах. Время  $t$  указано только для тех задач, которые решены с заданной точностью  $E$ .

**Анализ результатов, практическое применение программ.** По результатам экспериментов построена табл. 3. Она позволяет сравнить рассмотренные программы минимизации по величинам условных интегральных коэффициентов  $K_{\Sigma}$  их качества [8]:

$$K_{\Sigma} = b_t K_{\Sigma t} + b_{\Pi} K_{\Pi},$$

где  $K_{\Sigma t}$  — коэффициент суммарной эффективности по времени;  $K_{\Pi}$  —

Таблица 3

## Сравнение программ минимизации по условному показателю их качества

Программа (алгоритм)	Процент решенных задач и коэффициент эффективности по времени ( $\% \div K_{rt}$ )			$K_{\Sigma t}$	$K_{\Pi}$	$K_{\Sigma}$	M
	$r=1$	$r=2$	$r=3$				
НАСП	50 ÷ 2,4	0 ÷ 0	0 ÷ 0	1,4	8,3	35	7
ПКСП	50 ÷ 2,2	0 ÷ 0	0 ÷ 0	1,3	9,3	37	6
СОПР1	50 ÷ 2,0	0 ÷ 0	0 ÷ 0	1,1	10	38	5
СОПР2	50 ÷ 2,6	25 ÷ 0,8	0 ÷ 0	1,9	10	43	4
СОПР3	0 ÷ 0	0 ÷ 0	50 ÷ 0,5	0,3	10	32	8
ДФП	50 ÷ 2,6	50 ÷ 1,5	25 ÷ 0,5	2,7	8,3	44	3
РОЗ	50 ÷ 5,6	50 ÷ 6,7	50 ÷ 1,5	7,9	9,0	82	2
ММП	100 ÷ 11,9	50 ÷ 2,9	75 ÷ 2,6	10	9,3	98	1

коэффициент, характеризующий эффективность по затратам памяти;  $b_i, b_n$  — веса указанных показателей качества.

Показатель  $K_{\Sigma}$  можно определить следующим образом:

$$K_{\Sigma} = K_{1t} + K_{2t} + K_{3t};$$

$$K_{rt} = \sum_{i=1}^{n_r} \frac{n_i p_i}{t_i},$$

где  $K_{rt} (r = 1, 2, 3)$  — коэффициент условной эффективности по времени для задач  $r$ -го типа;  $n_r$  — количество решенных задач  $r$ -го типа;  $n_i$  — порядок (количество варьируемых параметров) решенной задачи;  $p_i$  — модуль показателя степени гарантированной достигнутой величины точности  $E$  решения задачи (в данном случае числа 4 и 12);  $t_i$  — время, затраченное на решение  $i$ -й задачи  $r$ -го типа.

Произведение  $n_i p_i$  характеризует в определенной степени сложность  $i$ -й задачи и точность ее решения. Таким образом, показатель условной эффективности по времени представляет собой сумму аддитивных добавок, каждая из которых прямо пропорциональна точности и сложности соответствующей решенной задачи и обратно пропорциональна затратам машинного времени на ее решение.

Коэффициент  $K_n$  обратно пропорционален памяти, занимаемой программой минимизации, и нормирован относительно наименьших затрат памяти (55 кбайт). Представленные в таблице величины  $K_{\Sigma}$  и  $K_n$  переведены в десятибалльную систему [8], причем таким образом, чтобы наибольшая оценка была равна 10.

Значения интегрального показателя  $K_{\Sigma}$  качества программ подсчитаны при весовых коэффициентах  $b_i = 7$  и  $b_n = 3$ , т. е. рассмотрен случай, когда более важным является быстрое действие, а не экономия памяти.

Исходя из этого критерия, можно считать, что в целом наиболее эффективен алгоритм ММП (см. последний столбец табл. 3, где М — место алгоритма в порядке уменьшения условного показателя качества).

При решении вопроса об использовании той или иной программы, кроме величины  $K_{\Sigma}$ , по которой определяется показатель М, следует учитывать первые три столбца таблицы, где приведены проценты решенных задач и коэффициенты эффективности по времени для различных типов функций. Поэтому, хотя для ММП  $M = 1$  в случае квадратичных функций с «врагом», с точки зрения большего быстрого действия лучше применять алгоритм Розенброка. По этой же причине, несмотря на то, что алгоритм СОПРЗ худший по показателю  $K_{\Sigma}$ , для квадратичных функций применение этого алгоритма предпочтительнее, чем первых четырех.

При решении реальных задач (задач параметрической оптимизации электронной схемы, отражающего многолучевого интерферометра и взрывомагнитного генератора) наилучшие результаты из рассмотренных программ показала программа ММП, затем РОЗ, ДФП, что полностью соответствует табл. 3. Полученные результаты не противоречат данным, приведенным в ряде работ: ММП лучше РОЗ, ДФП лучше СОПР [6], ММП лучше ДФП, ДФП лучше СОПР [4].

Можно предположить, что полученные результаты позволят уменьшить в какой-то степени ту меру неопределенности, которая свойственна существующим представлениям о ценности различных методов оптимизации [6].

## ЛИТЕРАТУРА

1. Безносков Г. П., Ефименко В. В., Загоруйко А. С., Стукалин Ю. А. Пакет программ численного анализа электронных схем и систем из конечных элементов.— В кн.: Автоматизация научных исследований на основе применения ЭВМ. Новосибирск, изд. ИАиЭ СО АН СССР, 1977.
2. Безносков Г. П., Ефименко В. В., Загоруйко А. С., Стукалин Ю. А. Обобщенный метод хорд в задачах моделирования на ЭВМ статики и динамики нелинейных схем.— В кн.: Автоматизация научных исследований на основе применения ЭВМ. Новосибирск, изд. ИАиЭ СО АН СССР, 1974.
3. Карманов В. П. Математическое программирование. М., «Наука», 1975.
4. Трохименко Я. К., Каширский И. С., Ловкий В. К. Проектирование радиотехнических схем на инженерных ЭЦВМ. Киев, «Техника», 1976.
5. Вычислительные методы в электродинамике. Под ред. Р. Митры. (Пер. с англ.). М., «Мир», 1977.
6. Аоки М. Введение в методы оптимизации. (Пер. с англ.) М., «Наука», 1977.
7. Пакет минимизации. Вып. 1. М., изд. МГУ, 1975.
8. Генделев Г. М. О подходах к оценке качества программ.— «Упр. сист. и маш.», 1977, № 3, с. 51—56.

Поступила в редакцию 12 июля 1978 г.;  
окончательный вариант — 13 октября 1978 г.

УДК 621.317.335

С. Я. КОРСАКОВ, В. В. КРЫЛОВ, Д. М. ПОНОМАРЕВ  
(Горький)

## ИССЛЕДОВАНИЕ ЭЛЕКТРОФИЗИЧЕСКИХ ПАРАМЕТРОВ ДИЭЛЕКТРИКОВ МЕТОДОМ ИМПУЛЬСНОГО ЗОНДИРОВАНИЯ

В последнее время использование ЭВМ в измерительных системах позволило открыть новую область применения метода импульсного зондирования для исследования электрофизических свойств материалов в широком диапазоне частот. Суть метода заключается в том, что образец помещается в специальную волновую направляющую систему (рис. 1) и облучается пачкой коротких электромагнитных импульсов длительностью  $10^{-10}$  с [1]. Отраженные и прошедшие через образец сигналы поступают на считывающее устройство стробоскопического осциллографа, кодируются и записываются в память ЭВМ. Далее производится специальная обработка записанной информации с целью получения данных об электрофизических свойствах материалов. Применительно к исследованию диэлектрических свойств различных веществ метод был опробован в работах [1, 2]. В [3] представлен полный обзор достигнутых результатов в этой области и показаны возможности дальнейшего использования метода. В работах [4—7] развита методика импульсного зондирования для большего числа различных приложений. Однако до настоящего времени отсутствовали результаты применения этого метода при исследовании веществ, изменяющих свои свойства под воздействием физических факторов. Это объясняется как сложностью алгоритмов определения электрофизических характеристик веществ, требующих использования мощных ЭВМ, так и трудностью применения прецизионных СВЧ-трактов формирования и распространения электромагнитных импульсов.

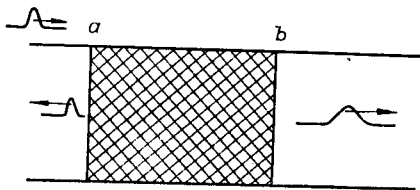


Рис. 1. Образец материала в волновой направляющей системе.