А. А. КАЛЕНТЬЕВ

ВВЕДЕНИЕ В АВТОМАТИЗИРОВАННОЕ ПРОЕКТИРОВАНИЕ

1985

Министерство высшего и среднего специального образования
Р С Ф С Р

Куйбышевский ордена Трудового Красного Знамени авиационный
институт имени академика С.П.Королева

А.А.Калентьев

ВВЕДЕНИЕ В АВТОМАТИЗИРОВАННОЕ ПРОЕКТИРОВАНИЕ

Утверждено
редакционно-издательским
советом института
в качестве учебного пособия

Куйбышев 1985

УДК 681.31: 001

Калентьев А.А. ВВЕДЕНИЕ В АВТОМАТИЗИРОВАННОЕ ПРОЕКТИРОВАНИЕ.- Куйбышев: Куйбышевский авиационный институт, 1985.- 72 с.

В пособии излагаются основы автоматизации проектирования, приводятся сведения о процессе проектирования, его свойствах, основных типах инженерных задач, вводятся понятия расчетной задачи, проектной задачи, дается классификация задач, рассматриваются принципы функционирования отдельных подсистем. Общие положения иллюстрируются конкретными примерами из технических дисциплин.

Пособие предназначено для студентов, обучающихся по специальности "Прикладная математика" со специализацией "Математическое обеспечение САПР", также может быть использовано студентами технических специальностей, будущих пользователей САПР.
Ил. 19, табл.5, библиогр.-9 назв.

Рецензенты: кафедра вычислительной техники Куйбышевского ордена Трудового Красного Знамени политехнического института им.В.В.Куйбышева, И.В.Чезганов

ПРЕДИСЛОВИЕ

Бурное развитие производительных сил и производственных отношений тесно связано с потребностями народного хозяйства в создании новых все более сложных машин и механизмов. Решение вопроса обусловливается противоречивыми требованиями: сокращение сроков разработки проекта и повышение его качества. Путем наращивания численности проектировщиков выполнить эти требования не удается. Разрешение поставленных задач возможно путем всесторонней автоматизации проектно-конструкторских работ, создания новой методологии проектирования, широкого использования методов математического моделирования, роботизации процесса производства и изготовления готовой продукции. Вопросам автоматизации проектных задач и посвящено настоящее пособие.

Цель автоматизации проектирования - повышение качества; снижение материальных затрат; сокращение сроков проектирования и количества инженерно-технических работников, занятых проектированием.

Для эффективного достижения поставленных целей необходимо теоретические достижения математиков поставить на службу инженеру. Математики разрабатывают для системы математические модели, методы исследования моделей, методы анализа результата и принятия решения, реализуют средства взаимодействия пользователя с системой в удобной для них форме, что позволяет инженеру-проектировщику сосредоточить внимание на профессиональных вопросах - получении желаемого результата. В САПР конструктор должен конструировать, а программист - программировать.

Под САПР, согласно определению, данному в работе [1], понимается комплекс программного, информационного, технического, организационного и методического обеспечений, направленный на решение проектных задач. В пособии рассматриваются только первые две компоненты комплекса. В первых трех разделах излагаются вопросы, предшествующие программному и информационному обеспечению. В последнем разделе формируется структура такого обеспечения.

Под программным обеспечением САПР понимается совокупность программ, предназначенных для описания процесса проектирования и выполнения проектных задач в автоматическом либо автоматизированном режиме. Что это за программы, чем определяется их выбор?

Под информационным обеспечением понимаются наборы данных, обеспечивающие функционирование программного обеспечения. Что это за наборы данных, как они взаимодействуют с программным обеспечением? Ответы на поставленные вопросы можно получить, изучив внутренний механизм процесса проектирования. Такое изучение проводится в первом разделе пособия.

Для автоматизации процесса проектирования его надо в определенном виде представить в ЭВМ, т.е. формализовать, представить совокупностью множеств и функций со всеми их структурными связями, для чего процесс проектирования задается своими свойствами, каждое из которых есть отношение над множеством переменных математической модели [2-5]. Они формализуются и вводятся в ЭВМ.

Во втором и третьем разделах пособия рассмотрен механизм решения инженерных задач, являющийся основой соответствующих алгоритмов. В качестве элементарного вводится понятие расчетной задачи. Процесс проектирования представляется как совокупность действий над расчетной задачей. Проводится классификация расчетных задач по типу ограничений, по размерности; рассматриваются алгоритмы решения для различных типов задач.

В последнем разделе рассматривается общая структура и логика функционирования базовой САПР, ориентированной на решение расчетных и проектных задач.

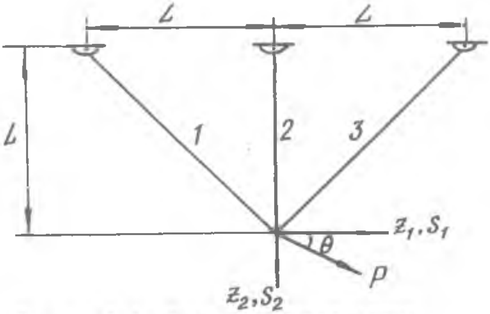
В основу пособия положен курс лекций, читаемых автором студентам специальности 0647 "Прикладная математика" со специализацией математическое обеспечение САПР. Автор выражает благодарность коллегам по кафедре технической кибернетики за критические замечания и полезные обсуждения.

I. АНАЛИЗ ПРОЦЕССА ПРОЕКТИРОВАНИЯ ТЕХНИЧЕСКОГО ОБЪЕКТА

I.I. Примеры инженерных задач

I.I.I. Проектирование трехстержневой фермы

Обозначим площади поперечных сечений стержней I,2,3 соответственно b_1, b_2, b_3; ρq - удельный вес материала, из которого изготовлена ферма; z_1, z_2 - горизонтальное и вертикальное смещения общего узла (рис. I.I).



Р и с. I.I. Трехстержневая ферма

Математическая модель фермы:

вес конструкции

f_1: $Q=\rho q(L\sqrt{2}\,b_1+Lb_2+L\sqrt{2}\,b_3)$;

горизонтальное и вертикальное смещения получаются из системы

f_2: $K(b)z=S$,

где $K(b)$ - положительно определенная матрица жесткости,

$$K(b)=\frac{\sqrt{2}\,E}{4 0}\begin{vmatrix} b_1+b_3 & b_1-b_3 \\ b_1-b_3 & b_1+b_3+2\sqrt{2}\,b_2 \end{vmatrix},$$

$$S=\begin{pmatrix} P\cos\theta \\ P\sin\theta \end{pmatrix},\quad z=\begin{pmatrix} z_1 \\ z_2 \end{pmatrix};$$

2-2453

напряжения в стержнях

$f_3: \quad \sigma_1 = \frac{E(z_1+z_2)}{2L}$,

$f_4: \quad \sigma_2 = \frac{Ez_2}{L}$,

$f_5: \quad \sigma_3 = \frac{E(z_2-z_1)}{2L}$;

осевые усилия

$f_6: F_1 = b_1\sigma_1$; $f_7: F_2 = b_2\sigma_2$; $f_8: F_3 = b_3\sigma_3$.

На заданной модели могут быть поставлены задачи.

З а д а ч а   а н а л и з а. При заданных линейных размерах L фермы, изготовленной из материала ρq с заданными сечениями (b_1 , b_2 , b_3) для фиксированной нагрузки (P , θ) рассчитать смещение z_1, z_2 общего узла. При этих же исходных данных рассчитать усилия в стержнях F_1 , F_2 , F_3 и вес конструкции Q .

З а д а ч а   с и н т е з а. При заданных линейных размерах L фермы, изготовленной из материала ρq, выбрать ее сечения b_1, b_2, b_3 таким образом, чтобы при фиксированной нагрузке (P, θ) осевые усилия не превышали допустимых $F_1^{доп}$, $F_2^{доп}$, $F_3^{доп}$. Для фермы, изготовленной из материала ρq с заданным сечением (b_1, b_2, b_3) выбрать ее линейные размеры таким образом, чтобы минимизировать вес фермы Q при фиксированной нагрузке (P, θ) и допустимых осевых усилиях стержней $F_1^{доп}$, $F_2^{доп}$, $F_3^{доп}$.

I.I.2. Колебания материальной точки массой m и с коэффициентом упругости С под действием периодической негармонической силы

М а т е м а т и ч е с к а я   м о д е л ь   к о л е б а н и й: уравнение колебаний

$\ddot{x}(t) + p^2 x(t) = \frac{1}{m} Q(t)$,

$Q(t)$ - периодическая негармоническая сила, действующая на материальную точку,

$Q(t) = \frac{H_0}{2} + \sum_{s=1}^{\infty} H_s \cos(s\omega t + \theta_s)$, $s = 1, 2, \ldots$

p - угловая частота собственных колебаний, $f_1: p^2 = \frac{C}{m}$.

Свободные колебания точки

$f_2: \quad x^{св}(t) = A\cos(pt - \theta)$,

где A - амплитуда колебаний, $f_3: A = \sqrt{x_0^2 + \left(\frac{\dot{x}_0}{p}\right)^2}$,

θ - угол сдвига фаз $f_4: \theta = arc\,tg \frac{\dot{x}_0}{p x_0}$,

$x_0, \dot{x}_0$ - заданные начальные условия.

Вынужденные колебания точки

$$f_5:\ x^{вын}(t)=\sum_{s=1}^{\infty} B_s \cos(s\omega t-\theta_s),$$

где $$f_6:\ B_s=\frac{H_s}{m(p^2-s^2\omega^2)},\ s=1,2,\ldots$$

На заданной модели могут быть поставлены задачи.

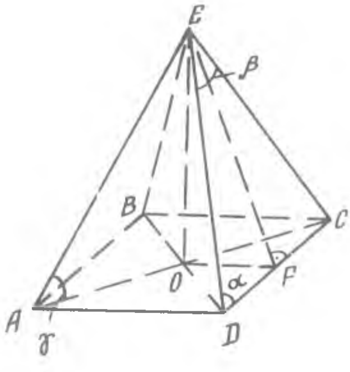
З а д а ч а а н а л и з а. Для точки заданной массы m и коэффициента упругости c при фиксированных начальных условиях определить максимальную амплитуду колебаний A_{max}, определить собственную частоту колебаний p.

З а д а ч а с и н т е з а. Подобрать массу точки m и коэффициент упругости c так, чтобы при фиксированных начальных условиях амплитуда колебаний A не превышала допустимого значения $A^{доп}$.

I.I.3. Расчет геометрических характеристик правильной пирамиды

Рассматривается правильная пирамида, в основании которой квадрат со стороной a, высота пирамиды h, угол между плоскостью основания и боковыми ребром γ (рис.I.2).

Обозначим: V - объем пирамиды; h_1 - высота боковой грани, $S_{бок}$ - боковая поверхность, S - полная площадь поверхности, z - функция ограничений, $\angle EDC=\alpha$, $\angle DEC=\beta$, ℓ - длина ребра, d - диагональ основания, $\ell=|AE|$.



Р и с. I.2. Правильная пирамида

Рассматривая математическую модель пирамиды, ограничимся некоторыми геометрическими соотношениями:

$$f_1:V=\frac{1}{3}a^2h;$$

$$f_2:S=S_{бок}+a^2;$$

$$f_3:\ell=\sqrt{h_1^2+\frac{a^2}{4}};$$

$$f_4:\alpha=\arccos\frac{2\ell}{a};$$

$f_5: \beta = 180° - 2\alpha;$ $\qquad f_{10}: \ell = \frac{h}{\sin\gamma};$

$f_6: S_{бок} = 2ah_1;$ $\qquad f_{11}: d = 2\ell\cos\gamma;$

$f_7: h = \frac{a}{\sqrt{2}}\, tg\,\gamma;$ $\qquad f_{12}: \alpha = arc\,tg\frac{2h_1}{a};$

$f_8: h_1 = \sqrt{h^2 + \frac{a^2}{4}};$ $\qquad f_{13}: \gamma = arc\,tg\frac{2h}{d};$

$f_9: d = a\sqrt{2};$ $\qquad f_{14}: \gamma = arc\sin\frac{h}{\ell};$ $\qquad f_{15}: z = 1{,}5h_1 + h.$

На модели пирамиды могут быть поставлены следующие задачи.

З а д а ч а а н а л и з а. Вычислить объем V и площадь боковой поверхности $S_{бок}$ пирамиды по заданным значениям стороны основания a и длине ребра ℓ.

З а д а ч а с и н т е з а. Пусть $a \in [2{,}7; 3{,}5]$; $\gamma \in [45°; 60°]$; требуется подобрать такую пару (a, γ), при которой $15 \leqslant S_{бок} \leqslant 20$ и $z = 1{,}5h_1 + h = 7{,}5$. Требуется найти все множество пар (a, γ), при которых выполняются заданные ограничения.

Анализируя и обобщая приведенные примеры, классифицируем инженерные задачи.

I.2. Классификация инженерных задач

Обозначим через $x = (x_1, \ldots, x_N)$ вектор проектных переменных $x \in X$. Под проектными переменными будем понимать конструктивные параметры и основные характеристики объекта. Такое деление проектных переменных предлагается в литературе по проектированию [2,3]. По существу оно идет от инженерной практики и связано с двумя задачами - анализа и синтеза объекта. Под к о н с т р у к т и в н ы м и п а р а м е т р а м и, как правило, понимаются геометрические характеристики (размеры) изготавливаемых деталей, положение центра отверстия, диаметр отверстия и т.д., т.е. параметры, по которым производится изготовление спроектированного объекта.

Х а р а к т е р и с т и к и о б ъ е к т а определяют его поведение в процессе функционирования и связь с внешней средой: полезная нагрузка, скорость полета, максимальная дальность, фазовые характеристики и т.д. Как правило, в уравнениях, выражающих закон функционирования объекта, конструктивные параметры в связанном либо преобразованном виде представлены коэффициентами, а характеристики объекта - переменными этих уравнений [5].

Введем отображение

$$P: S \to Y, \tag{I.I}$$

определенное в параллелепипеде;
$S \subset R^{n_1}$, $S = (S_1, \ldots, S_{n_1})$ - вектор конструктивных параметров объекта; $Y = (y_1, \ldots, y_{n_2})$ - вектор характеристик объекта;
P - отображение, заданное вектор-функцией и являющееся математической моделью изучаемого объекта.

В примере I.I.I отображение (I.I) задано совокупностью функций $f_1, \ldots, f_8$. Вектор конструктивных параметров $S=(L, b_1, b_2, b_3)$.

В примере I.I.2 отображению (I.I) соответствует уравнение колебаний. В частном случае оно может быть задано совокупностью функций $f_1, \ldots f_6$. Вектор конструктивных параметров $S=(m, c)$. В примере I.I.3 отображение задано совокупностью функций $f_1, \ldots f_{15}$. Конструктивными параметрами могут быть $S=(a, h, r)$.

З а д а ч а а н а л и з а. На математической модели объекта (I.I) по заданным значениям конструктивных параметров S_0 рассчитать его характеристики [6].

Решением задачи анализа назовем вектор значений переменных Y, вычисленный на модели $Y=P(S)$;

Алгоритм решения задачи анализа:
построение модели $Y=P(S)$,
вычисление значений переменных Y.

При проектировании технического объекта предварительно выбирается его конструктивное решение из некоторого множества типовых конструкций. Будем говорить, что задана структура объекта, если зафиксирована его конструкция.

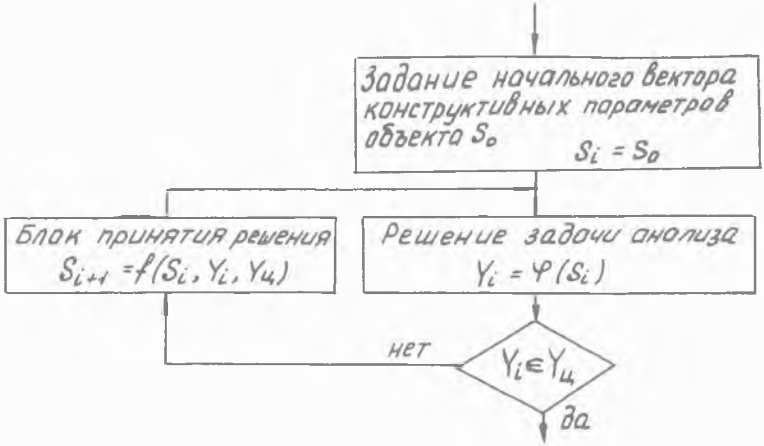
З а д а ч а п а р а м е т р и ч е с к о г о с и н т е з а. На математической модели (I.I) для объекта с заданной структурой требуется выбрать (спроектировать) такие значения его конструктивных параметров, при которых значения вычисленных характеристик удовлетворяют наперед заданным требованиям $Y_ц$, в частности, максимизируют заданный критерий.

Решением задачи параметрического синтеза назовем вектор значений конструктивных параметров S_n такой, что

$$Y=P(S_n) \in Y_ц . \qquad (I.2)$$

З а д а ч а с т р у к т у р н о г о с и н т е з а. Требуется выбрать такое конструктивное решение из заданного множества типовых конструкций и подобрать (спроектировать) такие значения конструктивных параметров объекта, при которых значения его вычисленных характеристик удовлетворяют наперед заданным требованиям, в частности, максимизируют заданный критерий.

Задача параметрического синтеза решается по схеме, приведенной на рис.I.3.



Р и с. I.3. Схема решения задачи параметрического синтеза

Выбор конструктивных параметров осуществляется методом итерации. Критерием такого выбора является выполнение условия (I.2) для вычисленных характеристик объекта. На одном шаге метода для заданного вектора конструктивных параметров решается задача анализа. При невыполнении условия включения $Y_i \in Y_ц$ осуществляют возврат к новой итерации путем корректировки значений конструктивных параметров блоком принятия решений. Корректировка может проводиться либо человеком, либо автоматически, если определена функция корректировки $S_{i+1} = f(S_i, Y_i, Y_ц)$. Эта функция выбирается такой, чтобы процесс поиска значений характеристик объекта Y_i сходился с заданной точностью за конечное число шагов.

Отличительной особенностью задачи структурного синтеза является наличие блока выбора типовой конструкции объекта из некоторого множества конструкций. Для каждой выбранной типовой конструкции будет своя математическая модель и, следовательно, будет меняться вид функции D, а возможно и метод ее решения. Соответственно изменится функция корректировки значений вектора конструктивных параметров объекта.

I.3. Свойства процесса проектирования

Анализ процесса проектирования различных технических объектов позволяет выделить некоторые инвариантные свойства этого процесса:

взаимосвязь проектных переменных;

многоэтапность проектирования;

итеративность процесса проектирования.

Свойство в з а и м о с в я з и означает, что переменная y связана с переменной x, если существует функция f, для которой x является исходной переменной, а y - вычисляемой.

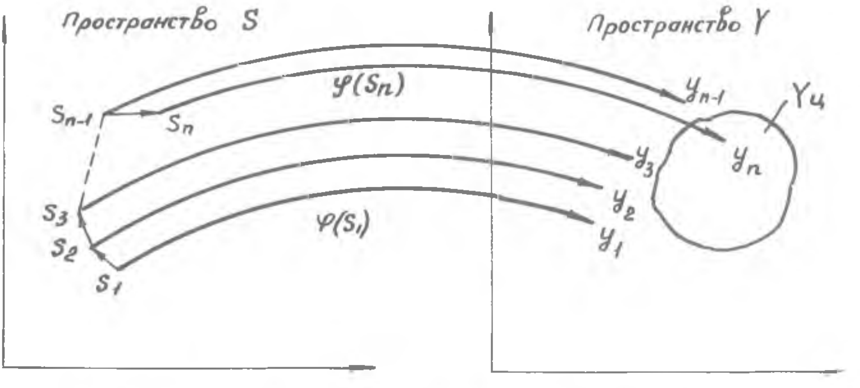
Цепочкой функциональных зависимостей назовем упорядоченную последовательность функций такую, когда выход одной функции является входом для другой. Переменная y может быть связана с переменной x цепочкой функциональных зависимостей, если такая цепочка существует. Из свойства взаимосвязи вытекает, что любая переменная может быть вычисляемой через связные с ней переменные, либо исходной для вычисления других переменных.

Переменная может быть только исходной, если она не связана ни с какими переменными математической модели. Переменная может быть только вычисляемой, если с ней не связана ни одна переменная математической модели (I.I). Переменные x, y называются независимыми на модели, если не существует цепочки функциональных зависимостей, связывающей эти переменные.

Свойство м н о г о э т а п н о с т и означает, что получаемые проектные решения, во-первых, носят приближенный характер, адекватный той математической модели, по которой производились расчеты. Во-вторых, получаемые проектные решения могут быть улучшены, если использовать другую, более точную математическую модель. Следует подчеркнуть, что проектировщик не может сразу взять за основу более точную математическую модель. В силу того, что любая проектная задача есть задача со множеством неопределенных факторов, относительно которых проектировщик должен вводить смысловые допущения и ограничения, и в силу того, что для более сложной (более точной) математической модели число неопределенных факторов существенно больше, невозможно использовать точную математическую модель на предварительном этапе. Пользуясь грубой математической моделью, проектировщик получает приближенные проектные решения, которые позволяют ему устранить часть неопределенных факторов. На новом этапе, при переходе к другой математической модели, устраняются другие неопределенные факторы и т.д. В силу наших ограниченных человеческих возможностей только таким последовательным переходом от одной модели к другой можно устранить большую часть неопределенных факторов. Вот как об этом сказано в работе [2] : "Процесс проектирования самолета - процесс последовательных приближений. Все решения здесь взаимосвязаны, и так как нельзя охватить сразу до деталей эти связи в целом, то каждое решение последовательно уточняется много раз".

Реализуется свойство многоэтапности в два действия: вычисление на математической модели, соответствующей одному этапу, и формирование модели нового этапа.

На каждом этапе процесса проектирования получение проектных решений носит итеративный характер (и т е р а т и в н о с т ь процесса проектирования). Получаемые проектные решения должны удовлетворять заданным требованиям. Требования к проектным решениям могут быть выражены либо через критерий оптимальности, накладываемый на вычисляемую переменную, либо условием включения получаемого решения в некоторую заданную область. Свойство итеративности проявляется при решении задачи синтеза (рис.I.4).



Р и с. I.4. Итеративный характер задачи синтеза

Решение задачи синтеза заключается в выборе такого вектора S_n значений исходных данных, при котором вычисленные значения проектных переменных удовлетворяют условию включения (I.2). Невыполнение условия включения (I.2) на i -м шаге означает переход к (i + I)-му шагу итерации, заключающемуся в выборе нового вектора S_{i+1} значений исходных данных лицом, принимающим решение.

Для реализации итеративности процесса проектирования необходимо создать алгоритмический аппарат, позволяющий участвовать проектировщику в процессе расчета, изменять значения исходных переменных, возвращаться на различные участки процесса расчета, одним словом, проектировщик должен уметь управлять процессом расчета. При этом стратегия управления должна позволять каждому пользователю "идти своим путем", т.е. строить свой эффективный путь получения проектных решений.

Отметим, что процесс проектирования не ограничивается описанными общесистемными свойствами; по мере выявления новых свойств будет решаться задача их алгоритмизации и вписывания в сложившуюся систему.

I.4. Способы описания объекта проектирования

Под методом автоматизированного проектирования будем понимать совокупность средств и способов, направленных на решение сложных проектных задач на ЭВМ с использованием минимальных знаний по программированию. К таким относятся языковые средства пользователя, информационные, алгоритмические и аппаратные средства системы.

Под языковыми будем понимать средства описания основных объектов процесса проектирования в форме, удобной для пользователя. Такими объектами являются математическая модель объекта проектирования, решаемая пользователем задача, результаты решения задачи. С помощью этих средств будем осуществлять взаимодействие пользователя с системой, с решаемой задачей, а также выполнение неформализуемых процедур.

Под информационными будем понимать средства описания основных объектов процесса проектирования в форме, удобной для работы ЭВМ, средства их преобразования из одной формы в другую.

Под алгоритмическими будем понимать средства описания основных объектов процесса проектирования в форме, удобной для интерпретации проектных задач, для выполнения формализуемых процедур.

А п п а р а т н ы е средства включают в себя ЭВМ и комплекс технических средств.

Математической модели объекта (I.I) поставим в соответствие ее языковое, информационное и алгоритмическое описания.

Языковое описание функции $f(x)$ будем задавать именем функции и списком имен входящих в нее переменных. Языковое описание модели $M=(F,X)$ определим как совокупность языковых описаний входящих в нее функций $F=(f)$.

Информационное описание модели будем задавать тройкой

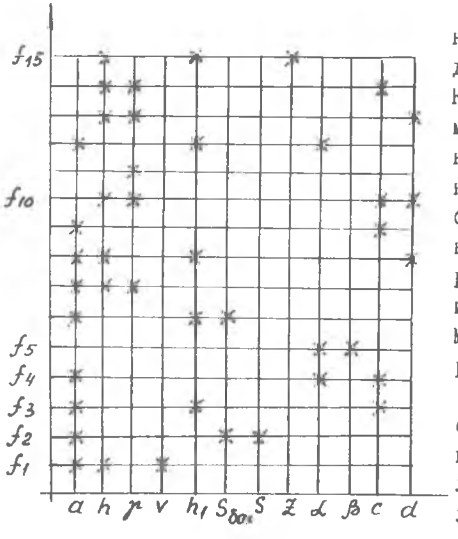
$$M_u = (KX, KF, U), \qquad (I.3)$$

где KX – каталог имен проектных переменных из множества X ;

KF – каталог имен функций из множества F ;

U – множество информационных связей модели $U \subset X \times F$.

Множество информационных связей будем называть еще в ы ч и с л и т е л ь н о й м о д е л ь ю.



Р и с. I.5. Множество связей

Пример. Информационное описание математической модели пирамиды, описанной в I.I.3 $F=(f_1,\dots,f_{15})$. Каталог KX состоит из имен переменных множества X : сторона основания, высота пирамиды, высота грани, угол γ , угол α , угол β , боковая поверхность, полная поверхность, объем пирамиды, диагональ, ребро, ограничения. Каталог KF состоит из имен функций множества F . Множество связей U показано на рис. I.5.

Под алгоритмическим описанием функции $f(x)$ будем понимать программу $P_f(x)$, написанную на любом алгоритмическом языке и реализующую функцию $f(x)$ в любой точке области определения функции. Будем предполагать, что программой $P_f(x)$ функция $f(x)$ реализуется в каждой точке с заданной точностью. Алгоритмическое описание функции может быть получено программистом.

Функция $f(x) \in F$ называется **базисной**. Функция $\varphi(x)$ называется **составной**, если она является суперпозицией базисных функций. Суперпозиция базисных функций определяется на множестве информационных связей U.

Будем предполагать, что для каждой базисной функции задано ее алгоритмическое описание. Совокупность алгоритмических описаний базисных функций $F=(f)$ назовем **библиотекой** $B(F)$ **проблемных модулей**.

Под алгоритмическим описанием составной функции $\varphi(x)$ будем понимать программу $P_\varphi(x)$, являющуюся суперпозицией программ базисных функций, входящих в описание составной функции.

I.5. Определение расчетной задачи

Пусть имеется математическая модель объекта проектирования (I.I), информационное описание которой имеет вид $M_U=(KX, KF, U)$. На множестве переменных X выделим подмножества S, Y, Z. Назовем $S \in R^n$ вектором исходных переменных, заданных своими значениями, $Y \in R^m$ - вектором вычисляемых переменных, $Z \in R^k$ -вектором вычисляемых переменных, на которые наложены ограничения. И пусть на модели (I.I) существуют вектор-функции $Y=F(S)$, $Z=\Phi(S)$.

Расчетной задачей на модели M назовем вектор-функцию $Y=F(S)$, позволяющую по заданным значениям вектора переменных S и ограничениям $\Phi(S) \leqslant \bar{Z}$ вычислить вектор переменных Y. Такое определение совпадает с определением задачи анализа, данным в п.I.2.

Расчетной задаче поставим в соответствие ее языковое, информационное и алгоритмическое описания. Языковое описание определяет постановку задачи пользователем - непрограммистом. Информационное описание содержит постановку задачи в форме, удобной для работы ЭВМ. Алгоритмическое описание содержит необходимую и достаточную информацию для адекватного выполнения алгоритма, реализующего функции F и Φ.

Языковым описанием расчетной задачи на модели M назовем совокупность вида

$$P3(M)=(S, Y, Z, \bar{Z}, \Pi(S)),$$

где $\bar{Z}$ - вектор значений ограничений;

$\Pi(S)$ - параллелепипед действий, или область определения расчетной задач.

Пусть $S=(S_1, \dots, S_n)$, $S \in R^n$. Будем говорить, что расчетная задача определена в точке n -мерного пространства, если каждая компонента вектора S задана одним значением $S_i=\alpha_i$, $i=\overline{1,n}$ $\Pi(S)=(\alpha_1, \alpha_2, \dots, \alpha_n)$. Если функция F определена в этой точке, то решением расчетной задачи будет m -мерный вектор Y.

Будем говорить, что расчетная задача определена в n -мерном параллелепипеде, если каждая компонента вектора S задана интервалом значений $S_i \in [\alpha_i, \beta_i]$, $i=\overline{1,n}$, $\Pi(S)=[\alpha_1,\beta_1]\times[\alpha_2,\beta_2]\times\ldots\times[\alpha_n,\beta_n]$.

Если функция F определена и непрерывна во всех точках параллелепипеда, то решением расчетной задачи в пространстве Y будет m -мерная ограниченная область.

Сложные вектор-функции F и Φ формируются на математической модели объекта проектирования путем указания переменных S, Y, Z и представляют собой частично упорядоченные последовательности функциональных зависимостей f из библиотеки проблемных модулей $B(F)$:

$F=(f_{i1}, f_{i2}, \ldots, f_{im'})$,
где $f_{ij} \in B(F)$, $j=\overline{1,m}$ – базисные функции;

$\Phi=(f_{j1}, f_{j2}, \ldots, f_{jK})$,
где $f_{jq} \in B(F)$, $q=\overline{1,K}$ – базисные функции.

Функции F и Φ определяют алгоритмическое описание расчетной задачи и получаются автоматически по языковому описанию расчетной задачи.

Итак, в данном разделе проведена классификация инженерных задач. Дано формальное определение задач анализа, параметрического и структурного синтеза; введено три способа описания математической модели: языковое, информационное, алгоритмическое; определена расчетная задача, являющаяся задачей анализа.

Вопросы и упражнения

1. Сформулируйте для каждого примера, приведенного в 1.1.1, 1.1.3 задачи анализа и синтеза.

2. Постройте математическую модель для треугольника. Постройте множество связей для этой модели.

3. Постройте множество связей для примеров, приведенных в 1.1.1, 1.1.2.

4. Сформулируйте задачи анализа и синтеза на модели треугольника. Приведите языковое описание расчетной задачи.

5. Постройте математическую модель для движения тела массы m с наклонной плоскости под углом α и коэффициентом трения K. Укажите множество связей. Приведите языковое описание расчетной задачи.

2. ИССЛЕДОВАНИЕ ЗАДАЧ, ДОПУСКАЮЩИХ АВТОМАТИЧЕСКОЕ РЕШЕНИЕ

2.I. Типы расчетных задач

В п.I.2 была дана классификация инженерных задач с точки зрения пользователя. Здесь будет представлена классификация инженерных задач с точки зрения их реализации на модели базовой САПР.

Пусть имеется языковое описание расчетной задачи и для определенности $S \in R^n$, $Y = y_1 \in R^1$, $Z = \emptyset$. Алгоритмическим описанием расчетной задачи является программа, реализующая сложную функцию $y_1 = F(S)$, $S \in \Pi(S)$. Будем предполагать, что функция $y_1 = F(S)$ определена на всей области $\Pi(S)$, непрерывна и дифференцируема.

По способу реализации сложной функции $y_1 = F(S)$, $S \in \Pi(S)$ в зависимости от области задания исходного вектора S будем различать два типа задач: точечные и функциональные. Для т о ч е ч н о й задачи каждая компонента вектора S задана своим значением. Областью определения точечной задачи является точка в n - мерном пространстве. Будем писать $\Pi(S) = (\alpha_1, \alpha_2, \ldots, \alpha_n)$. Образом этой точки будет точка на числовой оси y_1 . Реализация точечной задачи сводится к однократному выполнению сложной функции $y_1 = F(S)$ в заданной точке.

Ф у н к ц и о н а л ь н у ю задачу будем называть одномерной, если одна компонента вектора S задана интервалом значений, например, $S_1 \in [\alpha_1, \beta_1]$ при фиксированных значениях остальных компонент вектора S . Функциональную задачу будем называть n - мерной, если n -компонент вектора S заданы интервалом значений. Областью определения одномерной задачи будет отрезок прямой $[\alpha_1, \beta_1]$ при фиксированных значениях других компонент вектора S : $\Pi(S) = [\alpha_1, \beta_1] \times (\alpha_2, \ldots, \alpha_n)$.

Образом сложной функции $y_1 = F(S)$, одномерной, либо n -мерной, будет ограниченный отрезок на числовой оси y_1 . Реализация одномерной задачи сводится к n_1 -кратному выполнению сложной функции $y_1 = F(S)$ в дискретных точках отрезка прямой.

Дискретные точки на отрезке $[\alpha_1, \beta_1]$ определим по формуле

$S_1^i = \alpha_1 + (i-1)h_1$, $i = \overline{1, n_1}$,

где i - порядковый номер точки;

h_1 - шаг дискретизации по переменной S_1 ;

n_1 - число дискретных точек на отрезке $[\alpha_1, \beta_1]$.

$$n_1 = \left[\frac{\beta_1 - \alpha_1}{h_1}\right] + 1,$$

где $[a] = \begin{cases} a & \text{, если } a \text{ целое;} \\ \text{ближайшее большее целое число, если } a \text{ - дробное.} \end{cases}$

Реализация двумерной задачи сводится к ($n_1 \cdot n_2$)-кратному выполнению сложной функции $y_1 = F(S)$ в дискретных точках прямоугольника (S_1^i , S_2^j):

$$S_2^j = \alpha_2 + (j-1)h_2,$$

$$n_2 = \left[\frac{\beta_2 - \alpha_2}{h_2}\right] + 1. \qquad j = \overline{1, n_2}.$$

Размерность расчетной задачи зададим как (n, m, k),

где n - размерность области определения расчетной задачи, соответствующая числу переменных, заданных интервалом значений;

m - размерность вектор-функции $Y = F(S)$;

k - размерность вектор-функции $Z = \Phi(S)$.

При n = 0 будем иметь точечную задачу.

Для точечной задачи размерности (0, 2, 0) сложная функция $Y = F(S)$ вектору S^* = ($\alpha_1, \ldots, \alpha_n$) ставит в соответствие точку на плоскости F : ($\alpha_1, \ldots, \alpha_n$) $\to$ (y_1 , y_2). Для расчетной задачи размерности (n , 2, 0) образом области определения будет ограниченная область Ω в плоскости $y_1 0 y_2$.

2.I.I. Задачи без ограничений

Задана расчетная задача, в которой отсутствует вектор Z . Задачи без ограничений могут быть точечными и функциональными. Область определения сложной функции $Y = F(S)$ для таких задач задана в ее языковом описании. Выполнение расчетных задач без ограничений сводится к построению образа сложной функции $Y = F(S)$ по области $\Pi(S)$.

2.I.2. Типы ограничений и их влияние на область определения расчетной задачи

Пусть $\Pi(S)$ = $(\alpha_1, \ldots, \alpha_n)$, $Z = Z_1 \in R^1$, задана функция ограничений $Z_1 = \Phi(S)$ и значение ограничения $Z_1 = \bar{Z}_1$.

Обозначим $Z_1^{выч} = \Phi(\alpha_1, \ldots, \alpha_n)$.

При сравнении $Z_1^{выч}$ со значением ограничения $\bar{Z}_1$ в точке S могут возникнуть случаи, когда

$$|z_1^{выч} - \bar{z}_1| < \varepsilon, \tag{2.I}$$

где $\varepsilon > 0$ - заданная точность;

$$z_1^{выч} > \bar{z}_1 ; \tag{2.2}$$

$$z_1^{выч} < \bar{z}_1 . \tag{2.3}$$

Точка $S^* = (\alpha_1, \ldots, \alpha_n)$ удовлетворяет ограничению типа равенство по переменной z_1, если выполняется условие (2.I).

Точка $S^* = (\alpha_1, \ldots, \alpha_n)$ удовлетворяет ограничению строгого неравенства типа больше $\Phi(S) > \bar{z}_1$, если выполняется условие (2.2).

Точка $S^* = (\alpha_1, \ldots, \alpha_n)$ удовлетворяет ограничению строгого неравенства типа меньше $\Phi(S) > \bar{z}_1$, если выполняется условие (2.3).

Точка $S^* = (\alpha_1, \ldots, \alpha_n)$ удовлетворяет ограничению типа не меньше $\Phi(S) \geqslant \bar{z}_1$, если выполняется условие $(z_1^{выч} - \bar{z}_1) \geqslant \varepsilon$.

Точка $S^* = (\alpha_1, \ldots, \alpha_n)$ удовлетворяет ограничению типа не больше $\Phi(S) \leqslant \bar{z}_1$, если выполняется условие $(z_1^{выч} - \bar{z}_1) \leqslant \varepsilon$.

Пусть функция $z_1 = \Phi(S)$ определена, непрерывна, дифференцируема и однозначна на области $\Pi(S) = [\alpha_1, \beta_1] \times (\alpha_2, \ldots, \alpha_n)$.

Приведем без доказательства несколько теорем [7].

<u>Теорема I</u>. Точка $S^* = (S_1^*, \alpha_2, \ldots, \alpha_n)$, $S_1^* \in [\alpha_1, \beta_1]$ удовлетворяет ограничению типа равенство тогда и только тогда, когда является решением уравнения

$$\Phi(S) = \bar{z}_1 . \tag{2.4}$$

<u>Теорема 2.</u> Для любой точки $S^* = (S_1^*, \alpha_2, \ldots, \alpha_n)$, $S_1^* \in [\alpha_1, \beta_1]$ области определения функции $z_1 = \Phi(S)$ выполняется какое-нибудь из условий (2.I)-(2.3).

<u>Теорема 3.</u> Все решения уравнения (2.4), принадлежащие отрезку $[\alpha_1, \beta_1]$, разбивают его на интервалы, каждый из которых удовлетворяет только одному из условий (2.2) либо (2.3). Любые два соседних интервала не могут удовлетворять одному и тому же условию.

Пусть уравнение (2.4) имеет единственное решение на отрезке $[\alpha_1, \beta_1]$, разбивающее этот отрезок на два полуинтервала: $[\alpha_1, S_1^*)$, $(S_1^*, \beta_1]$. Если $\Phi'(S_1^*) > 0$, то на полуинтервале $[\alpha_1, S_1^*)$ выполняется условие (2.3), а на полуинтервале $(S_1^*, \beta_1]$ условие (2.2). Если $\Phi'(S_1^*) < 0$, то на полуинтервале $[\alpha_1, S_1^*)$ выполняется условие (2.2), а на полуинтервале $(S_1^*, \beta_1]$ условие (2.3).

Пусть на отрезке $[\alpha_1, \beta_1]$ заданы две функции $z_1 = \Phi_1(S_1)$, $z_2 = \Phi_2(S_1)$, заданы типы ограничений и их значения по каждой переменной. Построим алгоритм выделения на отрезке $[\alpha_1, \beta_1]$ интервалов, удовлетворяющих совокупным ограничениям.

1. Найдем решения первого уравнения $\Phi_1(S_1) = \bar{z}_1$ на отрезке $[\alpha_1, \beta_1]$, упорядочим их и построим интервалы. Выделим интервалы, удовлетворяющие заданному типу ограничений по переменной z_1.

2. Найдем решения второго уравнения $\Phi_2(S_1) = \bar{z}_2$ на отрезке $[\alpha_1, \beta_1]$, упорядочим их и построим интервалы. Выделим интервалы, удовлетворяющие заданному типу ограничений по переменной z_2.

3. Сравним интервалы, удовлетворяющие заданным ограничениям по переменным z_1 и z_2, и выделим их общие части, удовлетворяющие одновременно ограничениям по обеим переменным. Такие общие части, если они не пусты, образуют область определения расчетной задачи с заданными ограничениями. Пусть функция $z_1 = \Phi(S)$ определена, непрерывна, дифференцируема и однозначна на плоскости $\Pi(S) = [\alpha_1, \beta_1] \times [\alpha_2, \beta_2] \times (\alpha_3, \dots, \alpha_n)$.

Теорема 4. Точка $(S_1^*, S_2^*) \in [\alpha_1, \beta_1] \times [\alpha_2, \beta_2]$ удовлетворяет ограничению типа равенство тогда и только тогда, когда лежит на плоской кривой

$$\Phi(S_1, S_2) = \bar{z}_1 . \tag{2.5}$$

Для этого случая теорема 2 верна, и ее смысл заключается в том, что кривая (2.5), пересекающая прямоугольник, делит его на области, удовлетворяющие ограничениям разного типа. И для любой точки прямоугольника $\Pi(S)$ выполняется какое-нибудь из условий (2.1)-(2.3).

2.1.3. Задачи с ограничениями

Ограничение типа равенство. Для точечной задачи имеем $S^* = (\alpha_1, \dots, \alpha_n)$, $z_1 \in R^1$, $z_1 = \Phi_1(S^*)$, $z_1 = \bar{z}_1$. По теореме 1 точка S^* удовлетворяет значению ограничения $\bar{z}_1$, если она является решением уравнения (2.4).

Так как сложная функция $z_1 = \Phi_1(S^*)$ может быть вычислена непосредственно, то в случае выполнения неравенства

$$|\Phi_1(\alpha_1, \dots, \alpha_n) - \bar{z}_1| < \varepsilon \tag{2.6}$$

точка $S^* = (\alpha_1, \dots, \alpha_n)$ удовлетворяет значению ограничения.

Рассмотрим расчетную задачу размерности (I, m , I). По теореме I решения уравнения (2.4), принадлежащие отрезку $[\alpha_1, \beta_1]$, определяют область задания функции F . В зависимости от вида функции $z_1 = \Phi(S_1)$ и значения ограничения $z_1 = \bar{z}_1$ уравнение (2.4) может:

не иметь решений на отрезке $[\alpha_1, \beta_1]$;

иметь единственное решение на этом отрезке;

иметь множество решений, состоящее из набора дискретных точек.

Рассмотрим расчетную задачу размерности (2, m , I). Функция $z_1 = \Phi(S_1, S_2)$ определена на прямоугольнике
$\Pi(S) = [\alpha_1, \beta_1] \times [\alpha_2, \beta_2] \times (\alpha_3, \dots, \alpha_n)$.
По теореме 4 областью задания функции F будет кривая (2.5), определяемая как линия пересечения поверхности $z_1 = \Phi(S_1, S_2)$ с плоскостью $z_1 = \bar{z}_1$ и принадлежащая прямоугольнику $\Pi(S)$. Такую кривую будем задавать множеством пар точек (S_1^i , S_2^i), $i = \overline{1, M}$, где M - число точек на кривой, и обозначать как $L_{\bar{z}_1}(S_1, S_2)$.

Множество точек, удовлетворяющих значению ограничения по переменной z_1 может вырождаться в единственную точку, если плоскость $z_1 = \bar{z}_1$ касается поверхности $z_1 = \Phi(S_1, S_2)$; кривая $L_{\bar{z}_1}(S_1, S_2)$ либо пересекается с прямоугольником $\Pi(S)$ в единственной узловой точке, либо касается одной из его сторон.

Множество точек, удовлетворяющих значению ограничения по переменной z_1 , являясь точками одной кривой, может быть и не связным.

Рассмотрим расчетную задачу размерности (2, m , 2). Функции $z_1 = \Phi_1(S_1, S_2)$, $z_2 = \Phi_2(S_1, S_2)$ определены на прямоугольнике $\Pi(S) = [\alpha_1, \beta_1] \times [\alpha_2, \beta_2] \times (\alpha_3, \dots, \alpha_n)$. Аналогично теореме I точка (S_1^* , S_2^*) $\in \Pi(S)$ удовлетворяет ограничению типа равенство тогда и только тогда, когда является решением системы уравнений

$$\left.\begin{aligned} \Phi_1(S_1, S_2) &= \bar{z}_1, \\ \Phi_2(S_1, S_2) &= \bar{z}_2 \end{aligned}\right\}. \qquad (2.7)$$

Решением системы (2.7), а следовательно, областью определения функции F , может быть одна точка или конечное число точек.

Для расчетной задачи размерности (3, m , 2) функции $z_1 = \Phi_1(S_1, S_2, S_3)$, $z_2 = \Phi_2(S_1, S_2, S_3)$ определены на параллелепипеде:
$\Pi(S) = [\alpha_1, \beta_1] \times [\alpha_2, \beta_2] \times [\alpha_3, \beta_3] \times (\alpha_4, \dots, \alpha_n)$.

Ограничениям типа равенство по переменным z_1 и z_2 удовлетворяют точки кривой, полученной пересечением поверхностей $L_{\bar{z}_1}(S_1,S_2,S_3)$ и $L_{\bar{z}_2}(S_1,S_2,S_3)$. Для расчетной задачи размерности (n , m , 2) функции $z_1=\Phi_1(S)$, $z_2=\Phi_2(S)$ определены на n -мерном параллелепипеде $\Pi(S)=[\alpha_1,\beta_1]\times\ldots\times[\alpha_n,\beta_n]$. Множеством точек, удовлетворяющих ограничениям типа равенство по переменным z_1 и z_2 является гиперповерхность ($n-2$) -го порядка.

Аналогично рассматриваются случаи для $z\in R^K$.

О г р а н и ч е н и е т и п а н е р а в е н с т в о . Расчетные задачи такого типа рассмотрим на примере неравенства ($<$).

Рассмотрим расчетную задачу размерности (I, m , I). Пусть для определенности функция $z_1=\Phi(S_1)$ монотонно возрастает по переменной S_1 на отрезке $[\alpha_1,\beta_1]$. И пусть решение S_1^* уравнения $\Phi(S_1)=\bar{z}_1$ принадлежит отрезку $[\alpha_1,\beta_1]$. Согласно теореме 3 областью задания сложной функции F является полуинтервал $[\alpha_1, S_1^*)$.

Для расчетной задачи размерности (2, m , I) при пересечении плоскостью $z_1=\bar{z}_1$ поверхности $z_1=\Phi(S_1,S_2)$ все точки прямоугольника $\Pi(S)$ делятся кривой $\Phi(S_1,S_2)=\bar{z}_1$ на два множества точек, для которых:

$\Phi(S_1,S_2)<\bar{z}_1$;

$\Phi(S_1,S_2)>\bar{z}_1$.

Областью задания сложной функции F будет множество точек прямоугольника $\Pi(S)$, удовлетворяющих неравенству $\Phi(S_1,S_2)<\bar{z}_1$, ограниченное сторонами прямоугольника $\Pi(S)$ и кривой $L_{\bar{z}_1}(S_1,S_2)$. Множество точек, для которых выполняется неравенство $\Phi(S_1,S_2)<\bar{z}_1$, может быть и несвязным внутри прямоугольника $\Pi(S)$.

Для расчетной задачи размерности (n , m , I) областью задания сложной функции F будет множество точек параллелепипеда $\Pi(S)$, удовлетворяющих неравенству $\Phi(S)<\bar{z}_1$, ограниченное гиперплоскостями параллелепипеда $\Pi(S)$ и гиперповерхностью $L_{\bar{z}_1}(S_1,\ldots,S_n)$. Область задания сложной функции F может быть и не связной внутри параллелепипеда $\Pi(S)$.

Рассмотрим расчетную задачу размерности (0, m , 2) $\Pi(S)=$ $=S^*=(\alpha_1,\ldots,\alpha_n)$, введем две функции: $z_1^{выч}=\Phi_1(S^*)$, $z_2^{выч}=\Phi_2(S^*)$ и зададим значения ограничений $z_1=\bar{z}_1$, $z_2=\bar{z}_2$. Точка S^* удовлетворяет ограничениям типа неравенство ($<$) по совокупности переменных, если она одновременно удовлетворяет неравенствам:

$$\left.\begin{aligned}\Phi_1(S^*) &< \bar{z}_1\\ \Phi_2(S^*) &< \bar{z}_2,\end{aligned}\right\} \tag{2.8}$$

Для расчетной задачи размерности (I, m, 2) в 2.I.2 показано, как выделить множество точек, удовлетворяющих совокупным ограничениям.

Для расчетной задачи размерности (2, m, 2) обозначим

$\Pi_{\bar{z}_1}(S) = \{S \in \Pi(S) / \Phi_1(S_1, S_2) < \bar{z}_1\}$,

$\Pi_{\bar{z}_2}(S) = \{S \in \Pi(S) / \Phi_2(S_1, S_2) < \bar{z}_2\}$.

Областью определения сложной функции F будет область, удовлетворяющая совокупным ограничениям $\Pi_{\bar{z}}(S) = \Pi_{\bar{z}_1}(S) \cap \Pi_{\bar{z}_2}(S)$.

Для расчетной задачи размерности (n, m, 2) заданным ограничением из параллелепипеда $\Pi(S)$ выделяется n -мерное тело, ограниченное гиперповерхностями n -го порядка вида $\Phi_1(S) = \bar{z}_1$, $\Phi_2(S) = \bar{z}_2$ и гранями и ребрами параллелепипеда $\Pi(S)$. Образованное тело и является областью задания сложной функции F. Аналогично рассматривают случаи для $z \in R^K$.

С м е ш а н н ы е о г р а н и ч е н и я . Точка S^* = = (α_1, ..., α_n) удовлетворяет смешанным ограничениям, если одновременно выполняются условия:

$\Phi_1(S^*) = \bar{z}_1$, $\Phi_2(S^*) < \bar{z}_2$.

Рассмотрим расчетную задачу размерности (I, m, 2)

$S_1^* \in [\alpha_1, \beta_1]$, $\Phi_1(S_1^*) = \bar{z}_1$, $\tilde{S}_1 \in [\alpha_1, \beta_1]$, $\Phi_2(\tilde{S}_1) = \bar{z}_2$;

пусть для определенности $(\tilde{S}_1, \beta_1] \subset [\alpha_1, \beta_1]$ - интервал, на котором выполняется неравенство $\Phi_2(S_1) < \bar{z}_2$. Точка S_1^* удовлетворяет смешанным ограничениям, если $S_1^* \in (\tilde{S}_1, \beta_1]$.

Рассмотрим расчетную задачу размерности (3, m, K). Пусть K = 2 и на переменную z_1 наложено ограничение типа равенство, а на переменную z_2 наложено ограничение типа неравенство ($<$). Обозначим как $L_{\bar{z}_1}$ (S_1, S_2, S_3) - поверхность, описываемую равенством Φ_1 (S_1, S_2, S_3) = $\bar{z}_1$; $\Pi_{\bar{z}_2}$ (S_1, S_2, S_3) = $= \{S \in \Pi(S) / \Phi_2(S_1, S_2, S_3) < \bar{z}_2\}$ - область в пространстве (S_1, S_2, S_3), полученная из неравенства. Областью определения расчетной задачи, удовлетворяющей смешанным ограничениям, являются точки поверхности $L_{\bar{z}_1}$ (S_1, S_2, S_3), принадлежащие области $\Pi_{\bar{z}_2}$ (S_1, S_2, S_3).

Пусть K = 3, и на первые две переменные наложено ограничение типа равенство. Областью определения расчетной задачи, удовлетворяющей смешанным ограничениям, является кривая, полученная пересечением двух поверхностей $L_{\bar{z}_1}(S_1, S_2, S_3)$ и $L_{\bar{z}_2}(S_1, S_2, S_3)$ и принадлежащая области $\Pi_{\bar{z}_3}(S_1, S_2, S_3)$.

Пусть $K = 4$, и на первые три переменные наложено ограничение типа равенство. Областью определения расчетной задачи, удовлетворяющей смешанным ограничениям, является точка, являющаяся решением системы уравнений

$$\begin{cases} \Phi_1(S_1, S_2, S_3) = \bar{z}_1, \\ \Phi_2(S_1, S_2, S_3) = \bar{z}_2, \\ \Phi_3(S_1, S_2, S_3) = \bar{z}_3, \end{cases}$$

и принадлежащая области $\Pi_{\bar{z}_4}(S_1, S_2, S_3)$.

Рассмотрим расчетную задачу размерности (n , m , $K+1$), и пусть на первые K переменных наложены ограничения типа равенство:

$K < n$: областью задания расчетной задачи, удовлетворяющей смешанным ограничениям, является гиперповерхность ($n-K$)-го порядка, полученная пересечением K гиперповерхностей: $L_{\bar{z}_1}(S_1, \ldots, S_n)$, ... , $L_{\bar{z}_K}(S_1, \ldots, S_n)$; n-го порядка и принадлежащая области $\Pi_{\bar{z}_{K+1}}(S_1, \ldots, S_n)$;

$K = n$: областью задания расчетной задачи, удовлетворяющей смешанным ограничениям, является точка в n -мерном пространстве, полученная пересечением K гиперповерхностей n -го порядка и принадлежащая области $\Pi_{\bar{z}_{K+1}}(S_1, \ldots, S_n)$ (если такая точка существует).

Обозначим через ℓ - размерность вектора переменных z , на которые наложены ограничения типа неравенство. Мы рассмотрели случаи, когда ℓ = I. Для $\ell \geq 1$ область задания сложной функции принадлежит пересечению областей $\Pi_{\bar{z}_{K+1}}(S), \ldots, \Pi_{\bar{z}_{K+\ell}}(S)$.

2.2. Алгоритм решения расчетной задачи

Пусть описана расчетная задача размерности (n , m , K) на модели M (рис.2.I): $P3(M) = (S, Y, z, \bar{z}, \Pi(S))$.

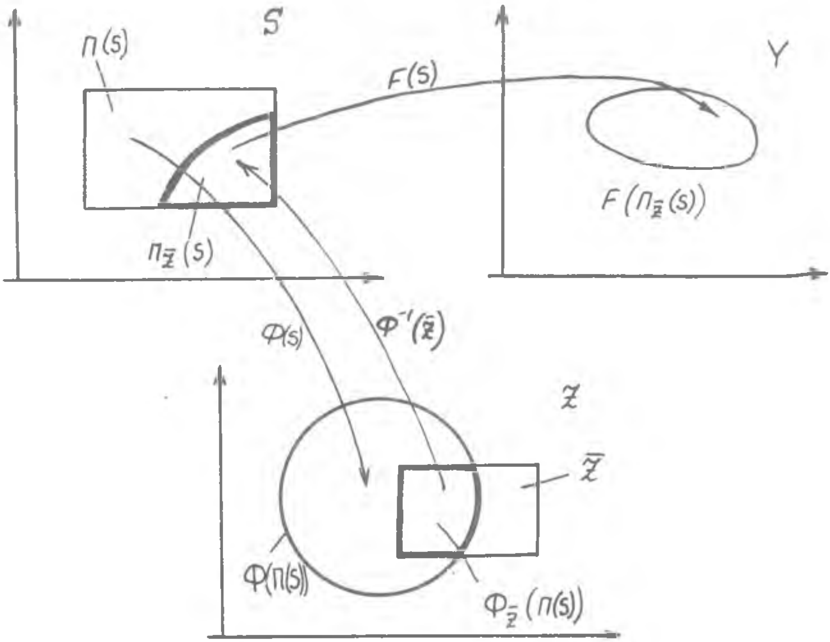
Обозначим:

$\Pi(S)$ - параллелепипед действия или область определения расчетной задачи;

$Y = F(S)$ - сложная функция, по которой вычисляется вектор переменных Y ;

$z = \Phi(S)$ - сложная функция, по которой вычисляется вектор ограничений z ;

$F(\Pi(S))$ - образ области $\Pi(S)$ в пространстве Y ;



Р и с. 2.1. Геометрическая иллюстрация к решению расчетной задачи

$\Phi(\Pi(S))$ - образ области $\Pi(S)$ в пространстве Z ;
$\Phi_{\bar z}(\Pi(S))=\{z\in Z \mid \Phi(S)<\bar z\}$ - образ той части параллелепипеда $\Pi(S)$ в пространстве Z , в точках которого выполняются заданные ограничения; $\Pi_{\bar z}(S)=\Phi^{-1}(\bar z)=\{S\in\Pi(S) \mid \Phi(S)<\bar z\}$ - прообраз образа $\Phi_{\bar z}(\Pi(S))$ в пространстве S или область задания сложной функции F ;
$F(\Pi_{\bar z}(S))=\{y\in Y \mid y=F(S),\ \Phi(S)<\bar z\}$ - образ множества $\Pi_{\bar z}(S)$ в пространстве Y .

Решением расчетной задачи назовем образ области $\Pi_{\bar z}(S)$ в пространстве Y , т.е.

$$Y=F(S),\ S\in\Pi_{\bar z}(S),$$
$$\Pi_{\bar z}(S)=\{S\in\Pi(S) \mid \Phi(S)<\bar z\}.$$

Алгоритм решения расчетной задачи состоит из двух частей:

построения прообраза $\Pi_{\bar z}(S)$ K -мерной вектор-функции $z=\Phi(S)$, удовлетворяющего вектору ограничений $\bar{\bar{Z}}$.

построения образа области $\Pi_{\bar{\bar{Z}}}(S)$ в пространстве Y .

Для задач без ограничений область $\Pi_{\bar{\bar{Z}}}(S)$ совпадает с параллелепипедом действий $\Pi(S)$.

При пересечении образа параллелепипеда действий с областью ограничений возможны случаи, когда:

ограничение по совокупности переменных не выполняется ни в одной точке параллелепипеда действий $\Pi(S)$, т.е. $\Phi(\Pi(S)) \cap \bar{\bar{Z}} = \phi$. Это означает, что области задания сложной функции F не существует;

ограничение по совокупности переменных выполняется во всех точках параллелепипеда действий $\Pi(S)$, т.е. $\Phi(\Pi(S)) \subset \bar{\bar{Z}}$. Это означает, что областью задания сложной функции F является весь параллелепипед действий;

ограничение по совокупности переменных выполняется в некоторых точках параллелепипеда действий $\Pi(S)$, т.е. $\Phi(\Pi(S)) \cap \bar{\bar{Z}} \neq \phi$. Это означает, что областью задания сложной функции F является некоторое подмножество области $\Pi(S)$, подлежащее вычислению.

Рассмотрим построение области $\Pi_{\bar{\bar{Z}}}(S)$. На параллелепипед действий $\Pi(S)$ накладывается дискретная прямоугольная сетка, размеры которой задаются пользователем. Во всех узловых точках вычисляются значения функции ограничений $Z = \Phi(S)$. Поведение функции между двумя соседними узлами будем определять по значениям функции в узлах сетки и значению ограничения.

Сложность и трудоемкость такого подхода заключается в том, что его построение существенно зависит как от размерности задачи, так и от соотношения величин n и k . В одних случаях областью задания сложной функции будет точка либо совокупность конечного числа точек, в других случаях - это кривая, заданная на плоскости либо в пространстве, которая может быть и не связной, и последний случай - поверхности сложной формы, которые также могут быть несвязными. Если в задаче отсутствуют ограничения типа равенство, то областью задания сложной функции F будет тело параллелепипеда $\Pi(S)$, ограниченное сложными поверхностями.

Построение образа области $\Pi_{\bar{\bar{Z}}}(S)$ в пространстве Y состоит в вычислении значений функции $Y = F(S)$ в заданных точках $S \in \Pi_{\bar{\bar{Z}}}(S)$.

2.3. Решение уравнения на отрезке

Рассмотрим уравнение вида

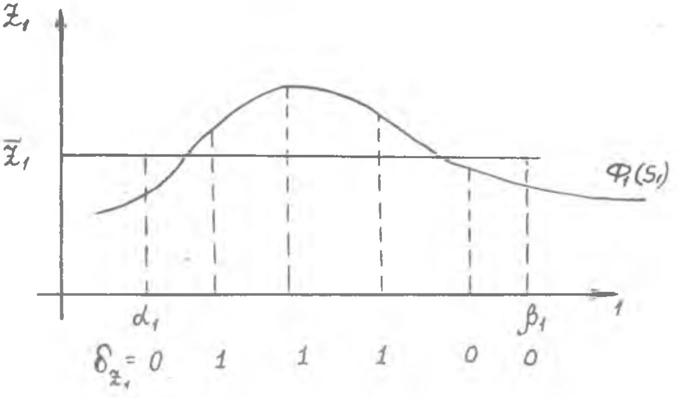
$$\Phi_1(S_1) = \bar{z}_1, \tag{2.9}$$

определенное на отрезке $[\alpha_1, \beta_1]$. Функция $z_1 = \Phi_1(S_1)$ определена, непрерывна, однозначна на отрезке $[\alpha_1, \beta_1]$ и может быть вычислена в любой точке отрезка заданным набором процедур. Обратная функция может быть неоднозначной. Требуется найти решение уравнения (2.9), принадлежащее отрезку $[\alpha_1, \beta_1]$, если оно существует.

Разобьем отрезок $[\alpha_1, \beta_1]$ дискретными точками с шагом h_1. В каждой дискретной точке вычислим значение функции и для сравнения его со значением ограничения введем логическую переменную

$$\delta_{z_1}(j) = \begin{cases} 0, \text{ если } & \Phi_1(S_1^j) > \bar{z}_1, \\ 1, \text{ если } & \Phi_1(S_1^j) < \bar{z}_1, \\ 2, \text{ если } & |\Phi_1(S_1^j) - \bar{z}_1| < \varepsilon. \end{cases}$$

Анализ значения логической переменной в соседних узлах дискретной сетки позволяет определить те участки дискретизации, на которых функция пересекается значением ограничения (рис.2.2). При этом возможны случаи, когда:



Р и с. 2.2. Применение логической переменной для решения уравнения

$\delta_{z_1}(j-1)=0, \delta_{z_1}(j)=2$, j - я дискретная точка является решением уравнения (2.9). Функция ограничений убывает на участке между (j - I) - й и j -й дискретными точками;

$\delta_{z_1}(j-1)=2, \delta_{z_1}(j)=0$, $(j-1)$- я дискретная точка является решением уравнения (2.9). Функция ограничений возрастает на участке между (j - I) - й и j - й дискретными точками;

$\delta_{z_1}(j-1)=1, \delta_{z_1}(j)=2$, (j) - я дискретная точка является решением уравнения (2.9). Функция ограничений возрастает на участке между (j - I) - й и j - й дискретными точками;

$\delta_{z_1}(j-1)=2, \delta_{z_1}(j)=1$, $(j-1)$- я дискретная точка является решением уравнения (2.9). Функция ограничений убывает на участке между (j - I) - й и j - й дискретными точками;

$\delta_{z_1}(j-1)=0, \delta_{z_1}(j)=1$ - решение существует в некоторой точке S_1^* на участке между (j - I) - й и j - й дискретными точками. Функция ограничений на этом участке убывает;

$\delta_{z_1}(j-1)=1, \delta_{z_1}(j)=0$ - решение существует в некоторой точке S_1^* на участке между (j - I) - й и j - й дискретными точками. Функция ограничений возрастает на этом участке;

$\delta_{z_1}(j-1)=\delta_{z_1}(j)=0$ - решения на участке $[(j-1), j]$ не существует. На всем участке функция превышает значения ограничения;

$\delta_{z_1}(j-1)=\delta_{z_1}(j)=1$ - решения на участке $[(j-1), j]$ не существует. На всем участке функция не превышает значение ограничения.

Решение уравнения (2.9), получаемое между точками дискретизации, найдем методом линейной интерполяции.

Обозначим $z_1^{j-1}=\Phi(S_1^{j-1})$, $z_1^j=\Phi(S_1^j)$.

Через точки (S_1^{j-1}, z_1^{j-1}) и (S_1^j, z_1^j) проведем прямую

$$\frac{S_1-S_1^{j-1}}{S_1^j-S_1^{j-1}}=\frac{z_1-z_1^{j-1}}{z_1^j-z_1^{j-1}}.$$

Подставляя в уравнение значение ограничения $z_1=\bar{z}_1$, найдем решение уравнения

$$S_1^*=\frac{\bar{z}_1-z_1^{j-1}}{z_1^j-z_1^{j-1}}(S_1^j-S_1^{j-1})+S_1^{j-1}.$$

Для повышения точности решения уравнения (2.9) разделим пополам отрезок $[S_1^{j-1}, S_1^j]$ и выделим ту его половину, которой принадлежит решение S_1^* первого приближения. В точке, делящей отрезок пополам, $S_1^{j/2}$ вычислим значение функции ограничений $z_1^{j/2}=\Phi_1(S_1^{j/2})$. Тогда в новом приближении

$$S_1^{**}=\frac{\bar{z}_1-z_1^{j/2}}{z_1^j-z_1^{j/2}}(S_1^j-S_1^{j/2})+S_1^{j/2}, \text{ если } S_1^*\in[S_1^{j/2}, S_1^j];$$

$$S_1^{**} = \frac{\bar{z}_1 - z_1^{j-1}}{z_1^{j/2} - z_1^{j-1}} (S_1^{j/2} - S_1^{j-1}) + S_1^{j-1}, \text{ если } S_1^* \in [S_1^{j-1}, S_1^{j/2}]$$

и т.д.

Если на отрезке $[\alpha_1, \beta_1]$ уравнение (2.9) имеет несколько корней, то все они вычисляются аналогично.

2.4. Построение плоской кривой на прямоугольнике

Рассматривается одно уравнение с двумя неизвестными

$$\Phi_1(S_1, S_2) = \bar{z}_1, \tag{2.10}$$

заданное на прямоугольнике $\Pi(S) = [\alpha_1, \beta_1] \times [\alpha_2, \beta_2] \times (\alpha_3, \ldots, \alpha_n)$. Требуется построить кривую, заданную уравнением (2.10) в пределах прямоугольника $\Pi(S)$. Функция $z_1 = \Phi_1(S_1, S_2)$ определена, непрерывна, однозначна на $\Pi(S)$ и вычисляема в каждой точке прямоугольника заданным набором процедур.

Будем предполагать, что уравнение (2.10) по каждой переменной при фиксированном значении другой переменной на одном шаге дискретизации не может иметь двух корней. Кривую будем задавать набором точек в порядке их следования. Несвязные части кривой будем рассматривать независимо.

Разобьем прямоугольник $\Pi(S)$ на $(n_1 \cdot n_2)$ дискретных точек с шагом h_1 и h_2 по переменным S_1, S_2 соответственно.

Теорема 5. Для того, чтобы точка (S_1^*, S_2^*) удовлетворяла уравнению (2.10), необходимо и достаточно, чтобы точка S_1^* являлась решением уравнения

$$\Phi_1(S_1, S_2^*) = \bar{z}_1, \text{ либо} \tag{2.11}$$

точка S_2^* была решением уравнения

$$\Phi_1(S_1^*, S_2) = \bar{z}_1. \tag{2.12}$$

Алгоритм строится следующим образом:

1. Фиксируем i -ю дискретную точку по переменной S_2.
2. Решаем уравнение (2.11) относительно переменной S_1: $\Phi_1(S_1, S_2^i) = \bar{z}_1$.
3. Повторяем пп.1-2 для всех $i = \overline{1, n_2}$.
4. Фиксируем j -ю дискретную точку по переменной S_1.

6-1150

5. Решаем уравнение (2.12) относительно переменной S_2 :

$\Phi_1(S_1^j, S_2) = \bar{z}_1$.

6. Повторяем пп.4–5 для всех $j=\overline{1,n_1}$.

7. Упорядочиваем все точки на связной части кривой.

Заметим, что данный алгоритм строит плоскую кривую, принадлежащую прямоугольнику $\Pi(S)$, путем полного перебора. В принципе для монотонных кривых можно его не делать. Если формируемая кривая неоднозначна, то прямоугольнику $\Pi(S)$ могут принадлежать ее несвязные части. Поиск отдельных частей кривой приведет к полному перебору по прямоугольнику $\Pi(S)$.

Рассмотрим построение монотонной кривой. Введем определение монотонности. Выберем произвольно на кривой $\Phi_1(S_1, S_2) = \bar{z}_1$ две точки (S_1', S_2'), (S_1'', S_2'') и будем говорить, что кривая $\Phi_1(S_1, S_2) = \bar{z}_1$ монотонно возрастает (убывает), если для любых S_1', S_1'', удовлетворяющих условию $S_1' < S_1''$, выполняется неравенство $S_2' < S_2''$ ($S_2' > S_2''$).

Для каждого фиксированного $i=\overline{1,n_2}$ уравнение $\Phi_1(S_1, S_2^i) = \bar{z}_1$ имеет единственное решение, а совокупность решений для всех i линейно упорядочена по переменной S_1 : $S_1^1 \leqslant S_1^2 \leqslant \ldots \leqslant S_1^{n_2}$. Аналогично для каждого фиксированного $j=\overline{1,n_1}$ уравнение $\Phi_1(S_1^j, S_2) = \bar{z}_1$ имеет единственное решение, а совокупность всех решений линейно упорядочена по переменной S_2 : $S_2^1 \leqslant S_2^2 \leqslant \ldots \leqslant S_2^{n_1}$.
Теперь все полученные решения – (S_1^1, S_2^i) ... $(S_1^{n_2}, S_2^i)$, (S_1^j, S_2^1), ... , $(S_1^j, S_2^{n_1})$ надо упорядочить вдоль кривой.

Для монотонно возрастающей кривой упорядочивание проводится по одной переменной, например S_1.

2.5. Решение системы двух уравнений

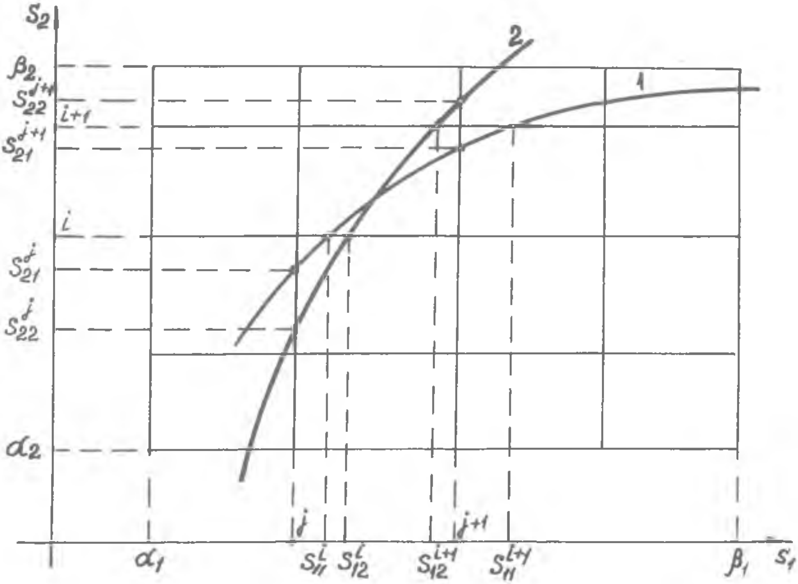
Рассмотрим систему уравнений

$$\Phi_1(S_1, S_2) = \bar{z}_1,$$

$$\Phi_2(S_1, S_2) = \bar{z}_2, \qquad (2.13)$$

определенную на прямоугольнике $\Pi(S) = [\alpha_1, \beta_1] \times [\alpha_2, \beta_2] \times (\alpha_3, \ldots, \alpha_n)$.
Функции $z_1 = \Phi_1(S_1, S_2)$, $z_2 = \Phi_2(S_1, S_2)$ непрерывны, однозначны на прямоугольнике $\Pi(S)$ и могут быть вычислены в любой точке прямоугольника заданным набором процедур.

Требуется найти решение системы (2.13) на $\Pi(S)$, если оно существует. Под решением системы будем понимать точку пересечения двух кривых (рис.2.3).



Р и с. 2.3. Решение системы двух уравнений: $1-\Phi_1(S_1,S_2)=\bar{z}_1$, $2-\Phi_2(S_1,S_2)=\bar{z}_2$

Решим систему, предположив, что обе кривые являются монотонно возрастающими. Введением дискретной сетки по переменным S_1 и S_2 мы разобъем прямоугольник $П(S)$ на элементарные прямоугольники. Обозначим, I – номер интервала между i -й и $(i+1)$ – й дискретными точками по переменной S_2 , $i=\overline{1,n_2-1}$; J – номер интервала между j и $(j+1)$ – й дискретными точками по переменной S_1 , $j=\overline{1,n_1-1}$. Тогда (I, J) – номер элементарного прямоугольника.

Теорема 6. Решение системы (2.13) принадлежит I интервалу, если для i ,$(i+1)$ дискретных точек выполняются соотношения $S_{11}^{i}<S_{12}^{i}$, $(S_{11}^{i}>S_{12}^{i})$, $S_{11}^{i+1}>S_{12}^{i+1}$, $(S_{11}^{i+1}<S_{12}^{i+1})$,
где S_{11}^{i} – решение уравнения $\Phi_1(S_1,S_2^{i})=\bar{z}_1$,
S_{12}^{i} – решение уравнения $\Phi_2(S_1,S_2^{i})=\bar{z}_2$.

Аналогично, для того, чтобы решение системы (2.13) принадлежало J -му интервалу, должны выполняться соотношения для j -й и $(j+1)$ – й дискретных точек.

$S_{21}^{j} > S_{22}^{j}$, $(S_{21}^{j} < S_{22}^{j})$ и

$S_{21}^{j+1} < S_{22}^{j+1}$, $(S_{21}^{j+1} > S_{22}^{j+1})$,

где S_{21}^{j} - решение уравнения $\Phi_1(S_{11}^{j}, S_2) = \bar{z}_1$,

S_{22}^{j} - решение уравнения $\Phi_2(S_1^{j}, S_2) = \bar{z}_2$.

Таким образом, мы определим элементарный прямоугольник (I, J), которому принадлежит решение системы.

Известно, что по способу построения плоской кривой в каждом элементарном прямоугольнике она описывается не более чем двумя точками. Такими точками являются (см.рис.2.3): для первой кривой – (S_{11}^{i}, S_{21}^{i}), $(S_{11}^{j+1}, S_{21}^{j+1})$, для второй кривой (S_{12}^{i}, S_{22}^{i}), $(S_{12}^{i+1}, S_{22}^{i+1})$. Между двумя точками каждая кривая интерполируется отрезком прямой, и решение системы (2.I3) находится из решения системы линейных алгебраических уравнений.

Таким образом, проанализировав расчетные задачи по размерности и типам ограничений, мы пришли к выводу, что:

область задания вектор-функции F для расчетных задач с ограничениями типа равенство сужается до набора отдельных точек, части кривой, или части поверхности, ограниченных параллелепипедом действий;

область задания вектор-функции F для расчетных задач с ограничениями типа неравенство сужается в тело, ограниченное поверхностями $\Phi(S) = \bar{z}$ и параллелепипедом действий;

область задания вектор-функции F для расчетных задач со смешанными ограничениями определяется ограничениями типа равенство внутри тела, полученного с помощью ограничений типа неравенство;

основной операцией при построении области задания функции F является решение уравнения на отрезке. Эту операцию используют при построении кривой, поверхности, решении системы уравнений.

Все приведённые алгоритмы могут быть выполнены автоматически.

Вопросы и упражнения

1. Определите размерность расчетных задач, описанных в п.I.I.
2. Постройте расчетную задачу размерности (2.I.I) с ограничением типа равенство, определенной: в единственной точке, на несвязной кривой.
3. В каких случаях расчетная задача размерности (I.I.2) с двумя ограничениями типа неравенство определена в единственной точке, неопределена на всем отрезке?

4. Для какого вида функции $Z_1=\Phi_1(S)$ расчетная задача размерности (2.I.I) будет определена на прямой в плоскости, на окружности?
5. Постройте алгоритм решения системы трех уравнений.
6. Постройте прообраз расчетной задачи размерности (2.I.2), содержащей одно ограничение типа равенство и одно ограничение типа неравенство.
7. Постройте прообраз расчетной задачи размерности (2.I.2), содержащей два ограничения типа неравенство.

3. ИССЛЕДОВАНИЕ ЗАДАЧ, РЕШАЕМЫХ АВТОМАТИЗИРОВАННЫМИ СРЕДСТВАМИ

3.I. Определение проектной задачи

В предыдущем разделе были рассмотрены типы расчетных задач и методы их решения. При построении прообраза $\Pi_{\bar{Z}}(S)$ в качестве основной операции использовали вычисление в узловых точках значений функции ограничений $Z=\Phi(S)$, заданной набором вычислительных процедур, и сравнение вычисленных значений со значениями ограничений. При построении образа области $\Pi_{\bar{Z}}(S)$ в пространстве Y основной операцией является вычисление значений функции $Y=F(S)$, $S\in\Pi_{\bar{Z}}(S)$. Эти операции выполняются последовательно и при наличии вычислительных процедур могут быть выполнены автоматически, т.е. без вмешательства человека. Всегда ли это возможно?

Пусть имеется некоторая цепочка взаимосвязанных вычислительных процедур

$$z_1, z_2, \dots, z_K, \tag{3.I}$$

где z_i — i-я вычислительная процедура, $i=\overline{1,K}$; и пусть для всех $i=\overline{1,K-1}$ между z_i и z_{i+1} процедурами имеется блок принятия решений.

Тогда цепочка (3.I) будет иметь вид

$$z_1, pz_1, z_2, pz_2, \dots, z_{K-1}, pz_{K-1}, z_K, \tag{3.2}$$

где pz_i - блок принятия решения после z_i вычислительной процедуры.

Под блоком принятия решений pz_i будем понимать некоторую функцию управления, которая в зависимости от вычисленных процедурой значений передает управление в наперед заданное место цепочки (3.2). В частности, передача управления на начало цепочки означает корректировку значений исходных данных.

Блок принятия решений pr_i будем называть ф о р м а л и з у е м ы м , если его функция управления определена. Блок принятия решений pr_i будем называть н е ф о р м а л и з у е м ы м, если его функция управления неопределена.

Будем говорить, что вычислительная цепочка (3.2) может быть выполнена автоматически, если она содержит только формализуемые блоки принятия решений.

Вычислительную цепочку (3.2) будем называть а в т о м а т и з и р о в а н н о й , если в ней имеется хотя бы один неформализуемый блок принятия решений.

Будем предполагать, что неформальное принятие решения осуществляется человеком. Принятое им решение определяется значениями вычисленных переменных и имеет силу при одном проходе по вычислительной цепочке (3.2). Если выполнение цепочки повторяется, то на каждом проходе принимается новое решение.

Рассмотрим задачи, в которых встречаются неформализуемые блоки принятия решений.

Проектной задачей на модели M назовем вектор-функцию $Y=F(S)$, вычисляющую вектор переменных Y по заданным значениям вектора переменных S при заданных ограничениях $\Phi(S)\leqslant \bar{Z}$, и функцию принятия решения $S_{i+1}= f(S_i, Y_i, Y_ц)$, позволяющую корректировать значения переменных S с целью выполнения условия включения (1.2). Такое определение совпадает с данным определением задачи синтеза (см.рис.1.3).

Проектной задаче поставим в соответствие ее языковое, информационное и алгоритмическое описание.

Языковым описанием проектной задачи на модели M назовем совокупность вида

$$ПЗ(M) = (РЗ(M), Y_ц), \tag{3.3}$$

где $РЗ(M)$ - языковое описание расчетной задачи на модели M, соответствующее вычислительной части проектной задачи;

$Y_ц$ - целевое подмножество в пространстве вычисляемых характеристик Y.

Отметим, что для проектной задачи остается справедливым все сказанное относительно расчетной задачи. Информационное описание расширяется описанием целевого подмножества, алгоритмическое описание расширяется некоторыми программными модулями, имеющими системный характер.

3.2. Проектные задачи без ограничений

Пусть целевое подмножество соответствует критерию максимума $Y_ц = y_{max}$. Имеется проектная задача $ПЗ(M) = (S, y, П(S), y_{max})$ размерности (n, I, 0). Областью определения задачи является параллелепипед действий $П(S)$. Он же является и областью задания сложной функции $y = F(S)$. Решением проектной задачи будет являться решение задачи математического программирования:

$$y_{max} = \max_{П(S)} F(S).$$

Пусть целевое подмножество является двумерным и соответствует критерию максимума по каждой переменной. Решением будет множество векторов Парето.

Пусть целевое подмножество соответствует допустимой области, заданной системой равенств и неравенств. Рассмотрим проектную задачу размерности (n, $m_1 + m_2$, 0), где m_1 - число равенств в определении целевого подмножества, m_2 - число неравенств в определении целевого подмножества. Решением проектной задачи, если оно существует, будет прообраз расчетной задачи размерности (n, 0, $m_1 + m_2$), для которой функция F является функцией ограничений.

Рассмотрим проектную задачу размерности $(n, m_1 + m_2 + m_3, 0)$, где m_1 - число равенств в определении целевого подмножества, m_2 - число неравенств в определении целевого подмножества, m_3 - число переменных, подлежащих максимизации. Если $m_3 = 1$, то находят прообраз $П_{\bar{Y}}(S)$ расчетной задачи размерности $(n, 1, m_1 + m_2)$, и далее, если прообраз существует, решается задача математического программирования $y_{3max} = \max_{П_{\bar{Y}}(S)} F_3(S)$. Если $m_3 > 1$, то находят прообраз $П_{\bar{Y}}(S)$ расчетной задачи размерности $(n, m_3, m_1 + m_2)$, и строится множество векторов Парето.

3.3. Проектные задачи с ограничениями

Пусть целевое подмножество соответствует критерию максимума $Y_ц = y_{max}$. Имеется проектная задача $ПЗ(M) = (S, y, z_1, \bar{z}_1, П(S), y_{max})$, размерности ($n$, I, $k_1 + k_2$). Областью определения задачи является параллелепипед действий $П(S)$. Областью задания сложной функции F является область $П_{\bar{Z}}(S) \subset П(S)$, если она существует. Решением проектной задачи является решение задачи математического программирования.

$$y_{max} = \max_{\Pi_{\bar{Z}}(S)} F(S).$$

Пусть целевое подмножество является двумерным и соответствует критерию максимума по каждой переменной. Решением будет множество векторов Парето, определяемых на области $\Pi_{\bar{Z}}(S)$.

Рассмотрим проектную задачу размерности $(n\ m_1+m_2+m_3, K_1+K_2)$ со смешанным целевым подмножеством, где m_1 и m_2 соответственно число равенств и неравенств, m_3 - число переменных, подлежащих максимизации. Если $m_3 =$ I, то ищем прообраз $\Pi_{\bar{Z}\bar{Y}}(S)$ расчетной задачи размерности (n , I, (m_1+K_1) + (m_2+K_2)). И далее, если прообраз существует, решаем задачу математического программирования

$$y_{3\,max} = \max_{\Pi_{\bar{Z}\bar{Y}}(S)} F_3(S).$$

Если $m_3 > 1$, то находим прообраз $\Pi_{\bar{Z}\bar{Y}}(S)$ расчетной задачи размерности (n , m_3, (m_1+K_1) + (m_2+K_2)) и далее строим множество векторов Парето, заданное на области $\Pi_{\bar{Z}\bar{Y}}(S)$.

3.4. Автоматизированные средства решения задач

В пп. 3.2, 3.3 показано, что решение проектной задачи сводится к решению расчетной задачи при условии, что соответствующий прообраз существует. Как быть, когда прообраз не существует?

Под *автоматизированными средствами* решения проектных задач будем понимать средства принятия решений человеком по согласованию значений конструктивных параметров S со значениями ограничений $\bar{Z}$. Такое согласование означает по сути построение человеком прообраза $\Pi_{\bar{Z}}(S)$ расчетной задачи. Будем предполагать, что функция принятия решения

$$S_{i+1} = f(S_i, Y_i, Y_Ц) \tag{3.4}$$

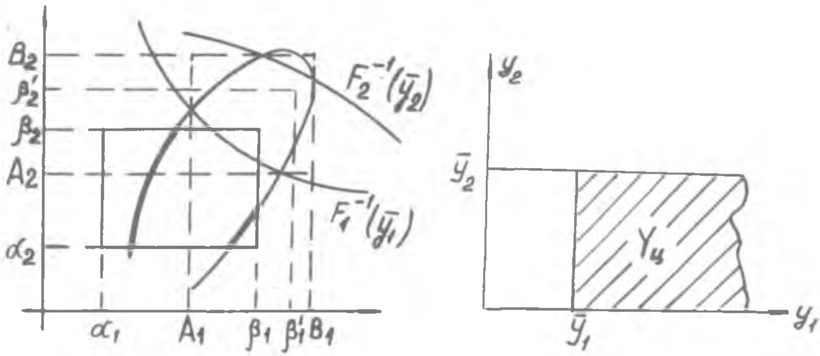
не определена. Для иллюстрации ее смысла рассмотрим следующий пример. Пусть задана проектная задача $ПЗ(M) = (S, Y, Z, \bar{Z}, \Pi(S), Y_Ц)$ размерности (2.2.I):

$\Pi(S) = [\alpha_1, \beta_1] \times [\alpha_2, \beta_2]$, $Y = (y_1, y_2)$,

$Z = z_1$, $\bar{Z} = \bar{z}_1$, $Y_Ц = \{(y_1, y_2) | y_1 \geq \bar{y}_1, y_2 \leq \bar{y}_2\}$

и по z_1 имеем ограничение типа равенство.

Решение этой задачи сведется к решению расчетной задачи размерности (2,0,I+2). Прообразом $\Pi_{\overline{ZY}}(S)$ расчетной задачи будут точки кривой $\Phi_1(S_1, S_2) = \bar{Z}_1$, принадлежащие телу, ограниченному кривыми $F_1(S_1, S_2) = \bar{y}_1$, $F_2(S_1, S_2) = \bar{y}_2$, и прямоугольником $\Pi(S)$. Как видим из рис.3.I, решения такой задачи на прямоугольнике $\Pi(S)$ не существует. Однако нетрудно за-



Р и с. 3.I. Построение области существования проектной задачи

метить, что полный прообраз $\Pi_{\overline{ZY}}(S)$ можно получить, если решать задачу на прямоугольнике $\Pi'(S) = [A_1, B_1] \times [A_2, B_2]$. Непустой проообраз можно получить путем построения прямоугольника $\Pi''(S) = [\alpha_1, \beta_1'] \times [\alpha_2, \beta_2']$.

Как скорректировать исходный параллелепипед действий, чтобы за конечное число шагов построить непустой прообраз? Задачу перехода от прямоугольника $\Pi(S)$ к прямоугольнику $\Pi''(S)$ либо $\Pi'(S)$ назовем з а д а ч е й п о с т р о е н и я с т р а т е г и и п о и с к а р е ш е н и я. Существует множество стратегий перехода от $\Pi(S)$ к $\Pi''(S)$. Задачи поиска эффективной стратегии рассматриваются в курсах теории игр и исследования операций.

Сформулировать задачу, которая решается в системе автоматизированного проектирования, можно следующим образом: разработать и реализовать такой аппарат или такие средства управления процессом расчета, которые позволяют каждому пользователю выбирать свою стратегию поиска решения проектной задачи. Реализовать эти средства возможно путем создания входного проблемно-ориентированного языка диалогового управления ходом вычислительного процесса.

Перечислим свойства диалогового языка:

непрерывность вычислений, вычисления можно продолжить с прерванного места;

повторяемость вычислений с измененными значениями исходных либо промежуточных данных.

Для реализации этих свойств вводятся следующие операторы языка.

1. Оператор ВЫЗВАТЬ < имя переменной > . Оператор формирует на экране дисплея вычисленное значение указанной переменной.

2. Оператор ИСПРАВИТЬ < имя переменной > = < новое значение переменной >. Под переменной понимается переменная из множества S.

3. Оператор ИЗМЕНИТЬ < старое имя программного модуля > , < новое имя программного модуля > .

4. Оператор ПРЕРВАТЬ < имя переменной > . Осуществляет прерывание после вычисления значения переменной, указанной в операторе.

5. Оператор ВОЗВРАТ НА ШАГ. Позволяет анализировать результаты, начиная с конца цепочки со сдвигом на одну вычислительную процедуру к началу цепочки.

6. Оператор ДОБАВИТЬ < ограничение > . Позволяет наложить более жесткие ограничения, если область существования решения задачи позволяет это.

7. Оператор УДАЛИТЬ < ограничение > . Применяется в случае отсутствия решения. За счет удаления одного, двух ограничений условия становятся более слабыми и возможно появление решения в заданном параллелепипеде действий.

8. Оператор СТОП. Означает окончание единовременного сеанса. Содержимое проектной задачи запоминается на внешнем носителе под именем пользователя.

Итак, в данной главе вводятся автоматизированные средства решения проектных задач. С этой целью: проектная задача представлена цепочкой процедур двух типов: вычислительными и процедурами принятия решения; процедуры принятия решения делятся на формализуемые и неформализуемые; формализуемая проектная задача, как и расчетная задача, может быть выполнена автоматически; неформализуемая проектная задача выполняется автоматизированными средствами; автоматизированные средства предполагают участие человека в части принятия решения средствами диалогового проблемно-ориентированного языка.

Вопросы и упражнения

1. Постройте проектную задачу размерности (2,1,1) с ограничением типа равенство: а) $Y_ц = y_{max}$; б) $Y_ц = [\bar{y}_1, \bar{y}_1]$; в) когда задачи а) и б) совпадают.

2. Постройте проектную задачу размерности (2,1,2) с ограничениями типа неравенство. Пусть $Y_ц = [\bar{y}_1, \bar{y}_1]$ и $F(П_{\bar{z}}(S)) \cap Y_ц = \phi$. Постройте функцию принятия решения диалоговыми средствами такую, чтобы на некотором шаге выполнялось $F(П_{\bar{z}}(S)) \cap Y_ц \neq \phi$ для случая: а) $y = a_1 S_1 + a_2 S_2$, $a_1, a_2 = const$; б) $y = a_1 S_1 S_2$, $a_1 = const$; в) $y = a_1 S_1 S_2 + a_2 S_1 + a_3 S_2$, $a_1, a_2, a_3 = const$.

3. Дано: $П_{\bar{z}}(S) = \{(S_1, S_2) | S_1 \geq 0, S_2 \geq 0, S_1 + S_2 = 5\}$

$$П(S) = [0;6] \times [0;6]; \; y = (S_1 + 2S_2)^2, \; Y_ц = [110;130].$$

Постройте функцию принятия решения такую, чтобы: а) $F(П_{\bar{z}}(S)) \subset Y_ц$; б) $F(П_{\bar{z}}(S)) \cap Y_ц \neq \phi$

4. Решите задачу

$П(S) = [0;5] \times [0;5]$; $П_{\bar{z}}(S) = \{(S_1, S_2) | S_1^2 = 2S_2\}$;

а) $y = S_1^2 + S_2^2 + S_1 S_2 + S_1 + S_2$; $Y_ц = y_{max}$.

Найти $S^* = \arg\max y(S)$, $S \in П_{\bar{z}}(S)$.

б) $y = S_1 + S_2$; $Y_ц = 5$.

Найти $S^* \in П_{\bar{z}}(S)$ такое, что $F(S^*) = 5$.

5. Дано: $П(S) = [2;6] \times [2;6]$;

$z = (S_1 - 4)^2 + (S_2 - 4)^2$; $\bar{z} = 4$;

ограничение типа неравенство $z \geq 4$.

$y = S_1^2 + S_2^2 + S_1 S_2 + S_1 + S_2$; $Y_ц = y_{max}$.

Найти $S^* = \arg\max\limits_{П_{\bar{z}}(S)} y(S)$

6. Решите задачу

$П(S) = [0;5] \times [0;5]$; $П_{\bar{z}}(S) = \{(S_1, S_2) | S_1 + S_2 = 5\}$;

$y_1 = S_1 S_2$; $y_2 = 5 - S_1 S_2$; $Y_ц = (y_{1max}, y_{2max})$.

Построить область Парето.

7. К чему приведет:

изменение значения ограничения в проектной задаче;

изменение типа ограничения.

4. СТРУКТУРА И ЛОГИКА ФУНКЦИОНИРОВАНИЯ БАЗОВОЙ САПР

4.I. Определение базовой САПР

Рассмотрев внутреннее содержание расчетных и проектных задач и процесс их решения, именуемый процессом проектирования, и выделив общие закономерности этого процесса с точки зрения их адекватного воспроизведения, постановку задачи создания базовой САПР определим следующим образом.

Пусть имеется математическая модель M (I.I) объекта проектирования, заданная множеством $\mathcal{F}$ дифференцируемых вектор-функций - $\mathcal{F}=(f_i)$, $i=\overline{1,\ell}$. Каждая функция f_i представлена вычислительной процедурой (программным модулем), совокупность которых по всем функциям образует библиотеку $B(\mathcal{F})$ вычислительных процедур. Будем называть ее библиотекой проблемных модулей, отражающих проблематику изучаемого объекта. На модели M ставится проектная задача $ПЗ(M)=(S, Y, Z, \bar{Z}, П(S), Y_ц)$. Требуется разработать программные средства для:

автоматического формирования в памяти ЭВМ математической модели M объекта проектирования;

автоматического формирования в памяти ЭВМ постановки проектной задачи;

автоматического формирования на модели M функциональных зависимостей $Y=F(S)$, $Z=\Phi(S)$;

автоматического формирования прообразов $\Phi^{-1}(\bar{Z})$ и $F^{-1}(Y_ц)$ в пространстве S .

Требуется разработать:

интерпретатор функций F и Φ на всей области определения задачи;

диалоговый язык автоматизированного управления процессом расчета;

форму технического отчета.

Введем: функцию A_1 , переводящую всякое языковое описание математической модели объекта ТЕКСТ I в его информационное описание ИОМ, A_1 : ТЕКСТ I $\rightarrow$ ИОМ;

функцию A_2 , переводящую языковое описание ТЕКСТ 2 проектной задачи в ее внутреннее (машинное) представление, A_2 :ТЕКСТ 2 $\rightarrow$ ВППЗ;

функцию A_3 , которая из информационного описания математической модели объекта выделяет информационное описание проектной задачи по ее внутреннему представлению, A_3 :(ОИМ, ВППЗ) → ИОЗ. Функция A_3 из математической модели объекта выделяет функциональные зависимости $Y = F(S)$, $Z = \Phi(S)$;

функцию A_4 , формирующую алгоритмическое описание проектной задачи, заданной на параллелепипеде действий: A_4 :(ИОЗ, ВППЗ) → АОЗ;

функцию A_5 , интерпретирующую выполнение проектной задачи: A_5 : (АОЗ, $B(\mathcal{F})$) → R , где R - результаты расчета;

функцию A_6 : (ТЕКСТ З) → $(\Pi'(S), R')$, функцию управления процессом расчета;

функцию A_7 : (М, ПЗ (М), R) → ТД, формирующую техническую документацию.

Базовой САПР для решения проектных задач на математической модели объекта назовем совокупность вида

$$\mathcal{F} = (A_1, A_2, A_3, A_4, A_5, A_6, A_7). \qquad (4.1)$$

4.2. Общая структура и схема функционирования

Условно базовую САПР (рис.4.1) можно разделить на три уровня. На языковом уровне осуществляется взаимодействие пользователя с системой в форме, удобной для пользователя - непрограммиста. Он может сформировать математическую модель объекта в памяти ЭВМ (A_1), сформулировать проектную задачу (A_2) с вызовом ее на выполнение, управлять процессом расчета (A_6). Подаваемые на вход системы языковые описания изучаемых процессов автоматически преобразуются в информационные описания этих процессов, которые образуют внутреннюю структуру данных системы.

На информационном уровне осуществляется взаимодействие программных средств системы с целью автоматического формирования текста рабочей программы, реализующей функции F и Φ (A_3 , A_4), формируется алгоритмическое описание проектной задачи.

На алгоритмическом уровне осуществляется выполнение проектной задачи с получением результата.

Вся система функционирует в замкнутом цикле, введение которого соответствует выполнению функции управления процессом расчета в случае получения неудовлетворительного результата. Выход из системы заканчивается выдачей технической документации.

Р и с. 4.I. Общая структурная схема базовой САПР

На рис.4.2 показано информационное взаимодействие подсистем, рассматриваются введенные нами функции. Основным элементом схемы взаимодействия являются информационные структуры. Наиболее распространенные структуры: информационное описание математической модели $\mathcal{M}$ объекта, в частности ее вычислительная модель, и внутреннее представление проектной задачи. Они используются почти во всех подсистемах.

На рисунке введены обозначения:

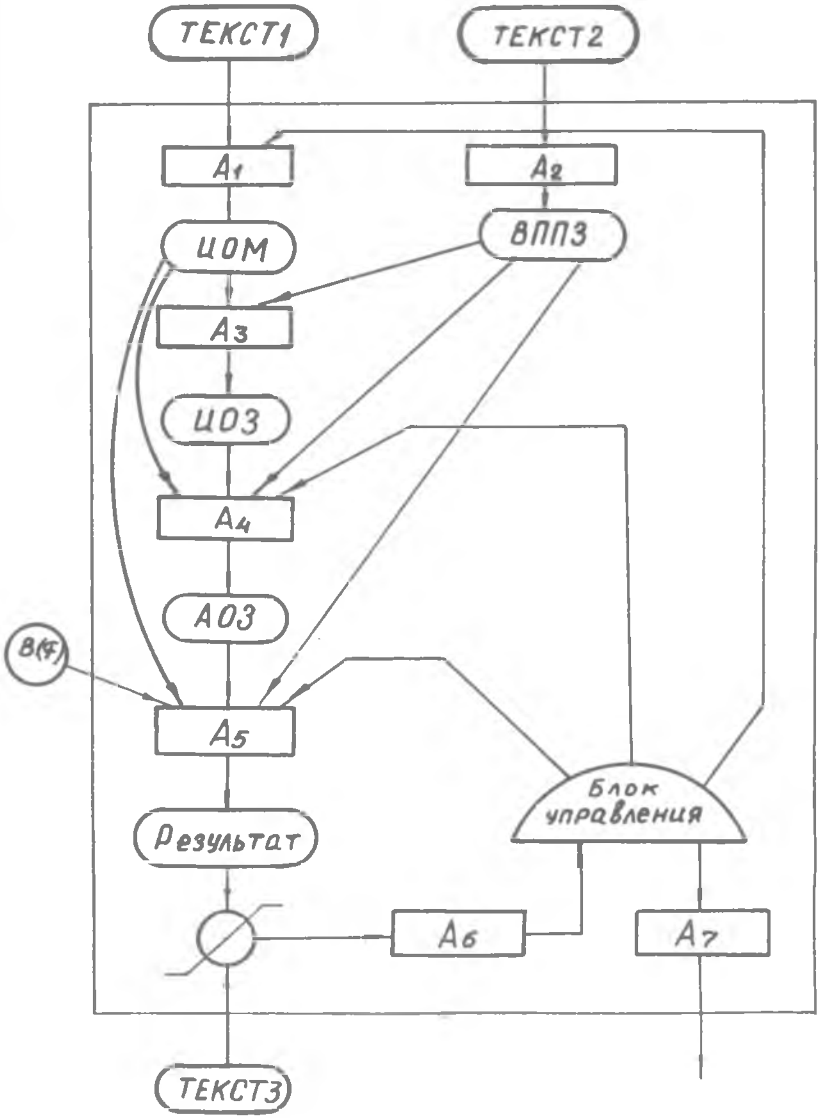
ТЕКСТ I – текст пользователя или языковое описание математической модели объекта;

ТЕКСТ 2 – текст пользователя или языковое описание постановки проектной задачи;

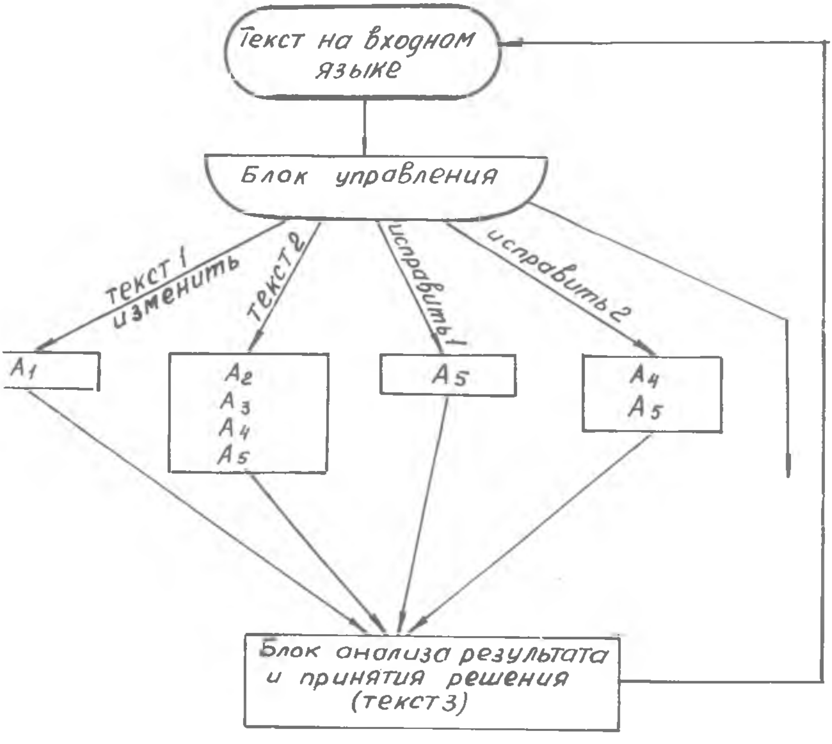
ТЕКСТ 3 – текст пользователя, содержащий директивы управления процессом расчета;

A_1, A_2, ..., A_7 – функции или подсистемы базовой САПР.

По схеме взаимодействия легко просматривается порядок работы системы. Подсистема A_1 может выполняться один раз на несколько выполнений остальных подсистем, если на одной модели решать несколько проектных задач. Блок передачи управления служит для запуска в действие соответствующих подсистем в зависи-



Р и с. 4.2. Схема взаимодействия подсистем мости от поступившей управляющей директивы пользователя. Система организована по замкнутому принципу (система с обратной связью). Функционирование системы (рис.4.3) осуществляется под действием



Р и с. 4.3. Схема функционирования системы

монитора, координирующего всю ее работу. Его основные функции: осуществление взаимодействия с пользователем; вызов на выполнение соответствующих тексту пользователя подсистем, либо их совокупности; реализация информационных операторов языка управления процессом расчета.

Под действием текста ТЕКСТ I монитор вызывает подсистему A_1 и передает ей управление. После того как информационное списание математической модели объекта будет сформировано, сеанс может быть окончен и продолжен в следующий выход на ЭВМ.

Выполнение подсистемы A_2 под действием текста ТЕКСТ 2 может быть осуществлено в случае, если ему предшествовало выполнение подсистемы A_1. Под действием ТЕКСТ 2 автоматически выполняются подсистемы A_2, A_3, A_4, A_5 с выдачей результата проектной задачи. На этом сеанс может быть окончен и продолжен в следующий выход на ЭВМ. В случае неудовлетворительного результата пользователь может его скорректировать посредством операторов языка управления процессом расчета.

4.3. Внутренняя структура данных системы

4.3.I. Информационное описание математической модели объекта

Множеству функций $\mathcal{F}$ было поставлено в соответствие: KX - каталог переменных, KF - каталог функций, U - вычислительная модель, реализующая свойство взаимосвязи проектных переменных.

Каталог переменных будем представлять парой ($KATX$, DX), где $KATX$ - массив символьной информации, состоящий из записей переменной длины. Каждая запись соответствует имени одной переменной. Имена переменных заносятся в $KATX$ в порядке их появления в языковом описании функции. Каждой переменной ставится в соответствие целочисленный номер, соответствующий номеру записи этой переменной; DX - целочисленный массив, I-й элемент которого содержит число символов в I-й записи. На информационной структуре ($KATX$,DX) определены две функции: g_1: имя переменной $\to$ номер переменной, g_2: номер переменной $\to$ имя переменной.

Аналогично парой ($KATF$, DF) будем представлять каталог функций. На этой паре также определены взаимно однозначные функции: φ_1: имя функции $\to$ номер функции; φ_2: номер функции $\to$ имя функции.

Вычислительная модель U - это линейный целочисленный массив, состоящий из трех частей. В первой части содержится описание функциональных зависимостей. Во второй части описаны переменные модели. Третья часть включает адресные ссылки из первых двух частей. Описание функциональных зависимостей состоит из записей фиксированной длины. Каждая запись описывает одну функцию. Поля этой записи имеют следующий смысл (табл.4.I).

Т а б л и ц а 4.I

Описание функции

№ поля	Идентификатор	Наименование
I	N_f	Номер функции
2	m_{11}	Число входных переменных функции
3	m_{21}	Число выходных переменных функции
4	A_{11}	Адресная отсылка в третью часть массива, настроенная на начальный адрес списка входных переменных
5	A_{21}	Адресная отсылка в третью часть массива, настроенная на начальный адрес списка выходных переменных

Описание переменных состоит из записей фиксированной длины. Каждая запись описывает одну переменную. Поля этой записи имеют следующий смысл (табл.4.2).

Т а б л и ц а 4.2

Описание переменных

№ поля	Идентификатор	Наименование
I	N_x	Номер переменной
2	m_{12}	Число функций, для которых эта переменная выходная
3	m_{22}	Число функций, для которых эта переменная входная
4	A_{12}	Адресная отсылка в третью часть массива, настроенная на начальный адрес списка выходных функций

Окончание табл.4.2

№ поля	Идентификатор	Наименование
5	A_{22}	Адресная отсылка в третью часть массива, настроенная на начальный адрес списка входных функций

На вычислительной модели определим следующие функции: g_3: номер функции $\to$ описание функции, g_4: номер переменной $\to$ описание переменной.

4.3.2. Информационное описание проектной задачи

Множеству $S \subset X$ исходных переменных поставим в соответствие информационную структуру D, представляющую линейный целочисленный массив, i-й элемент которого содержит номер S_i-й переменной.

Объединению множеств Y и Z поставим в соответствие информационную структуру N, представляющую линейный целочисленный массив.

Функциям F и Φ поставим в соответствие информационную структуру $RPOSL$, представляющую линейный целочисленный массив, i-м элементом которого является номер элементарной вычислительной процедуры из библиотеки проблемных модулей $B(\mathcal{F})$. Структура $RPOSL$ формируется автоматически (функция A_3) на вычислительной модели U как переход от D к N, $RPOSL: D \xrightarrow{U} N$.

Введем правило размерности для проектных переменных. Обозначим размерность 0 типа - скаляр, размерность I типа - вектор, размерность 2 типа - прямоугольная матрица и т.д.

Пусть задана функция двух переменных $x = f(S_1, S_2)$.

Правило сложения размерностей формулируется так: размерность вычисляемой переменной x равна сумме размерностей исходных независимых переменных, с которыми переменная x связана. Если исходные переменные S_1, S_2 размерности типа 0, то x размерности типа 0. Если S_1 -размерности типа 0, а S_2 - размерности типа I, то x -размерности типа I. Если S_1, S_2 размерности типа I, то x размерности типа 2 и т.д.

Исходную переменную $s \in S$ будем называть скаляром, если она задана одним значением. Переменную $s \in S$ будем называть вектором, если она задана интервалом значений.

На вычислительной модели U построим функцию $y_j = f(s_i)$, в предположении, что она существует. Функция представлена цепочкой вычислительных процедур

$$f = f_{i1}, f_{i2}, \ldots, f_{ik}. \tag{4.2}$$

Переменные $x_{i1}, \ldots, x_{ik-1}$, посредством которых осуществляется связь $y_j = (f_{i1}, \ldots, f_{ik})(s_i)$, назовем п р о м е ж у т о ч н ы м и. Заметим, что эти переменные не входят ни в одно из множеств S, Y, Z. Очевидно, что промежуточная переменная связана с переменной s_i.

Обозначим D –множество промежуточных переменных, посредством которых осуществляется переход $S \to (Y, Z)$. Тогда все переменные множеств S, Y, Z, D назовем п е р е м е н н ы м и р е ш а е м о й з а д а ч и. Определим размерность каждой переменной, за исключением исходных S. Понятие размерности переменной распространяется на переменную только в пределах одной задачи. Пронумеруем все переменные решаемой задачи и отдельно переменные каждой размерности.

Переменным решаемой задачи поставим в соответствие информационную структуру $MADR$, представленную целочисленной прямоугольной матрицей. Каждая строка матрицы описывает одну переменную и состоит из следующих полей (табл.4.3).

Т а б л и ц а 4.3

Размерности и адреса переменных

№ поля	Идентификатор	Наименование
1	N_x	Номер переменной
2	i_x	Признак размерности 1 типа $i_x = \begin{cases} 0, \text{ если } N_x \text{ размерности типа } 0 \\ i, \text{ если } N_x \text{ размерности 1 типа} \end{cases}$ i – порядковый номер переменной в классе переменных размерности 1 типа

Окончание табл.4.3

№ поля	Идентификатор	Наименование
3	j_x	Признак размерности 2 типа $j_x=\begin{cases}0, & \text{если } N_x \text{ размерности 1 типа,}\\ j, & \text{если } N_x \text{ размерности 2 типа}\end{cases}$ j - порядковый номер переменной в классе переменных размерности 2 типа
4	K_x	Признак размерности 3 типа $K_x=\begin{cases}0, & \text{если } N_x \text{ размерности 2 типа}\\ K, & \text{если } N_x \text{ размерности 3 типа}\end{cases}$ K - порядковый номер переменной в классе переменных размерности 3 типа

Примечание. Если $i_x = 0$, то номер строки переменной в $MADR$ есть ее порядковый номер в классе переменных размерности 0 типа.

Всем значениям вычисляемых переменных поставим в соответствие информационную структуру $RABFL$, представляющую линейный вещественный массив, состоящий из записей переменной длины. В первой записи содержатся значения переменных размерности 0 типа. Длина записи равна числу всех переменных решаемой задачи. Во второй записи будем хранить значения переменных размерности 1 типа, если отсутствуют переменные с высшей размерностью.

Для установления соответствия между именем переменной (точнее номером) и ее значением на структуре $MADR$ введем функцию формирования адреса q_5 : номер переменной $\rightarrow$ адрес переменной, по которому в $RABFL$ хранится ее значение.

Числовым данным, описывающим параллелепипед действий, значение ограничений, допустимое целевое подмножество, поставим в соответствие информационную структуру, представленную парой (MS, SM), где MS - целочисленный линейный массив, содержащий номера переменных, образующих параллелепипед действий, множество ограничений, допустимое целевое подмножество; SM - вещественная прямоугольная матрица, структура которой представлена в табл.4.4. Между строками массивов MS и SM установлено взаимно однозначное соответствие: в J -й строке SM описана переменная, номер которой хранится в J -м элементе MS. Таблица 4.4 состоит из трех частей. В первой части описаны переменные, образующие параллелепипед действий, во второй части описаны переменные, на которые наложены ограничения, в третьей части описаны переменные целевого подмножества.

Каждая строка таблицы состоит из четырех полей, имеющих следующий смысл.

Таблица 4.4

Текущее состояние проектной задачи

№ поля*	Идентификатор	Наименование
I	S_i	Текущее значение переменной, заданной интервалом значений
	$z_i^{выч}$, $(y_i^{выч})$	Текущее значение вычисленной переменной в S_i -й точке
2	α_i	Начальное значение переменной, заданной интервалом значений
	$\bar{z}_i$ $(y_{i\,зад})$	Значение ограничения либо граница целевого подмножества
3	β_i	Конечное значение переменной, заданной интервалом значений
	ε_z	Точность сравнения вычисляемого значения переменной с заданным
4	h_i	Шаг изменения значений переменной

* Все поля имеют вещественный формат длиной в одно слово.

4.4. Алгоритмы функционирования подсистем

4.4.I. Алгоритм автоматического формирования информационного описания математической модели объекта (A_1)

П о с т а н о в к а з а д а ч и . Дано: языковое описание математической модели объекта проектирования (ТЕКСТ I); информационное описание математической модели в форме (KX, KF, U).

Требуется построить алгоритм, преобразующий ТЕКСТ I в форму (KX , KF , U).

П р а в и л а   н а п и с а н и я   в х о д н о г о   т е к с т а. ТЕКСТ I состоит из однотипных предложений. Анализ текста сводится к анализу одного предложения. Каждое предложение представляет запись переменной длины и описывает одну функцию. Каждая запись состоит из трех полей переменной длины: поле имени функций, поле входных переменных функций, поле выходных переменных. Поля отделяются разделителями. Началом первого поля служит начало всей записи. Разделитель между первым и вторым полем является признаком конца первого поля и началом второго поля. Аналогично между вторым и третьим полем. Признаком конца третьего поля служит разделитель конца записи.

Приведем синтаксические правила написания ТЕКСТ I:

< ТЕКСТ I > : : = НАЧАЛО < предложение > [< предложение >] КОНЕЦ

< предложение > : : = < разделитель I >< имя функции >< разделитель 2 > < список входных переменных >< разделитель 3>< список выходных переменных >< разделитель 4 >

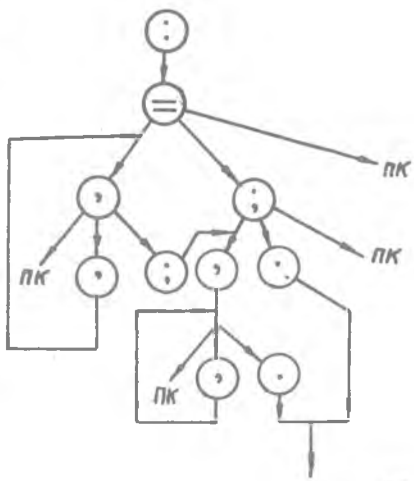
< разделитель I > : : = :
< разделитель 2 > : : = =
< разделитель 3 > : : = ;
< разделитель 4 > : : = .

< имя функции > : : = набор алфавитно-цифровых символов, начинающийся с буквы и длиной не более I6 символов

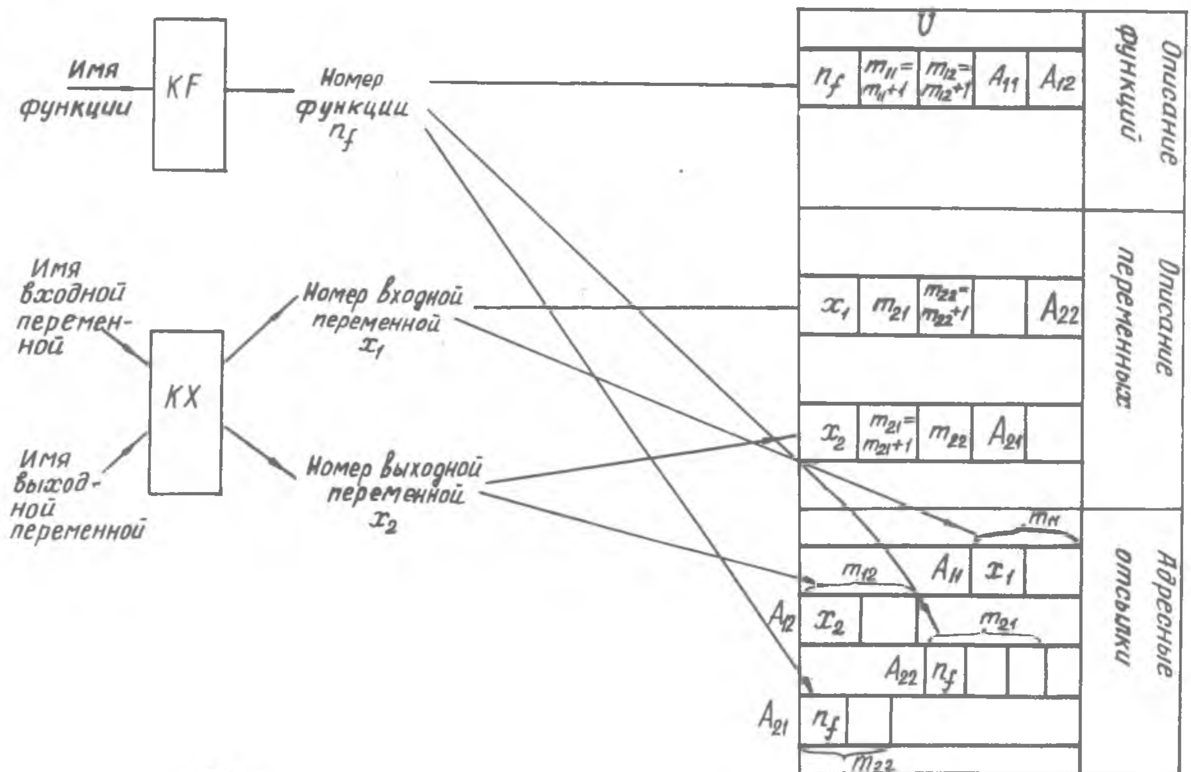
< список входных переменных > : : = < имя переменной > [, < имя переменной >]

< список выходных переменных > : : = < имя переменной > [, < имя переменной >]

< имя переменной > : : = набор алфавитно-цифровых символов, начинающийся с буквы, и длиной не более 32 символов.



Р и с. 4.4. Граф грамматического разбора предложения ТЕКСТ I (*ПК* – признак конца перфокарты)



Р и с. 4.5. Семантический анализ предложения ТЕКСТ I

В соответствии с указанными синтаксическими правилами на рис. 4.4 представлен граф грамматического разбора предложения [8].

Семантический анализ предложения приведен на рис. 4.5. Стрелками показано, куда засылаются номера функций и переменных. На модели U адресные отсылки A_{11}, A_{12}, A_{22}, A_{21} показаны в третьей части модели по месту расположения.

4.4.2. Алгоритм автоматического формирования внутреннего представления проектной задачи в памяти ЭВМ (A_2)

П о с т а н о в к а з а д а ч и. Дано языковое описание постановки проектной задачи (ТЕКСТ 2); внутреннее представление проектной задачи в форме (D , N , $MADR$, MS , SM , $RABFL$).

Требуется построить алгоритм, преобразующий ТЕКСТ 2 в форму (D , N , $MADR$, MS , SM , $RABFL$) [9].

П р а в и л о н а п и с а н и я в х о д н о г о т е к с т а. ТЕКСТ 2 состоит из трех разнотипных предложений, соответствующих трем операторам входного языка: ДАНО, НАЙТИ, ПРИ УСЛОВИИ. В предложении ДАНО описываются исходные переменные S задачи и параллелепипед действий. В предложении НАЙТИ описываются вычисляемые переменные Y . В предложении ПРИ УСЛОВИИ описываются заданные ограничения и допустимое целевое подмножество. Анализ текста сводится к анализу этих предложений. Каждое предложение представляет запись переменной длины. Предложения отделяются разделителями. Начало второго предложения является признаком конца первого предложения. Начало третьего предложения является признаком конца второго предложения.

Приведем синтаксические правила написания ТЕКСТ 2.

< ТЕКСТ 2 > : : = НАЧАЛО < предложение I > < предложение 2 >
[< предложение 3 >] КОНЕЦ

<предложение I > : : = ДАНО < список исходных переменных >

<предложение 2 > : : = НАЙТИ <список вычисляемых переменных >

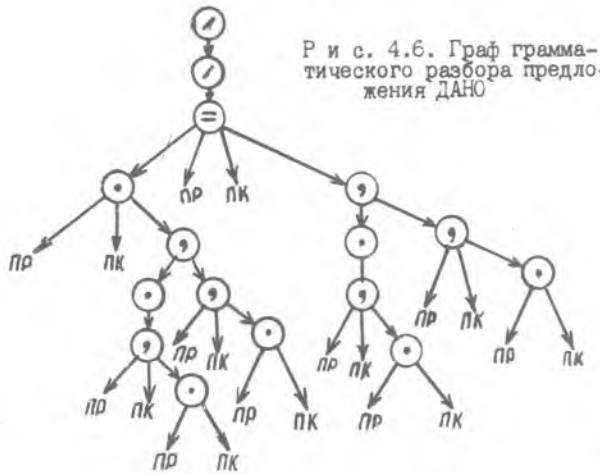
<предложение 3 > : : = ПРИ УСЛОВИИ < список ограничений >

<список исходных переменных > : : = < описание переменной >
[< описание переменной >]

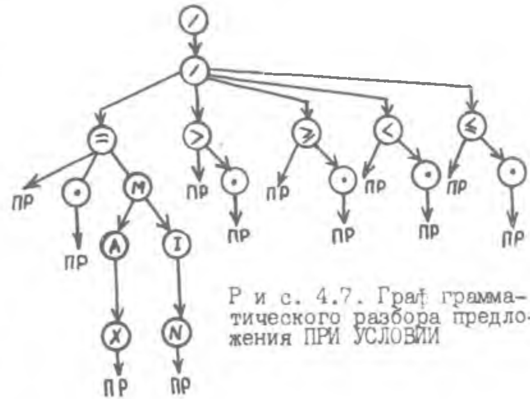
< описание переменной > : : = '< имя переменной >'< разделитель I >
< значение переменной >|'< имя переменной >'< разделитель I >
< интервал изменения переменной >
< имя переменной > : : = набор алфавитно-цифровых символов, начинающихся с буквы и длиной не более 32 символов
< разделитель I > : : = =
< значение переменной > : : = < целое > [· [< целое >]]
< целое > : : = 0|1|2|3|4|5|6|7|8|9
< интервал изменения переменной > : : = < начальное значение переменной >< разделитель 2 >< конечное значение переменной> <разделитель 2 >< шаг изменения переменной >
<начальное значение переменной > : : = < значение переменной >
< конечное значение переменной > : : = < значение переменной >
< шаг изменения переменной > : : = < значение переменной >
< разделитель 2 > : : = ,
< список вычисляемых переменных > : : = '< имя переменной >' ['< имя переменной >']
< список ограничений> : : = < ограничение > [< ограничение >]
< ограничение > : : = '< имя переменной >'< тип ограничения > < значение ограничения >
< тип ограничения > : :=<|>|≤|≥|=
< значение ограничения > : : =<значение переменной >| MAX|MIN

С т р у к т у р а а л г о р и т м а.

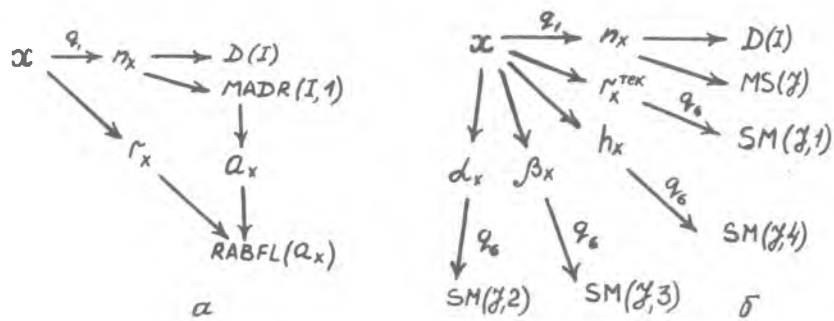
В соответствии с указанными синтаксическими правилами на рис.4.6 представлен граф грамматического разбора предложения ДАНО, введены следующие обозначения: ПР признак конца описания переменной, ПК – признак конца перфокарты. На графе грамматического разбора предложения НАЙТИ проводится анализ только на открывающий и закрывающий апострофы. Граф грамматического разбора предложения ПРИ УСЛОВИИ показан на рис. 4.7. На рис. 4.8 приведен семантический анализ предложения ДАНО, где q_6 – функция перевода вещественного числа из символьного формата в вещественный. Двойная стрелка (⟷) означает взаимно однозначное соответствие по строке между именем переменной и ее значением. Стрелками показано, куда засылаются соответствующие переменные и их значения. На рис. 4.9 приведен семантический анализ предложений НАЙТИ и ПРИ УСЛОВИИ.



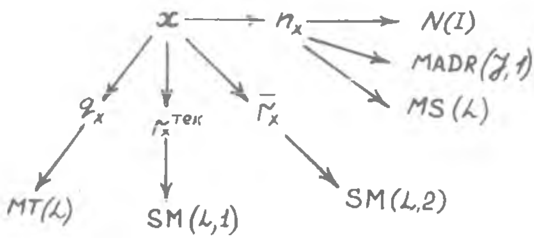
Р и с. 4.6. Граф грамматического разбора предложения ДАНО



Р и с. 4.7. Граф грамматического разбора предложения ПРИ УСЛОВИИ



Р и с. 4.8. Семантический анализ предложения ДАНО: а – для переменной типа скаляр; б – для векторной переменной



Р и с. 4.9. Семантический анализ предложений НАЙТИ и ПРИ УСЛОВИИ

4.4.3. Алгоритм автоматического формирования информационного описания проектной задачи (A_3)

Данный алгоритм формирует функции F и Φ проектной задачи. Так как на вычислительной модели всякая функция задается цепочкой элементарных вычислительных процедур и так как эти процедуры могут выполняться строго последовательно, то будем формировать функции F и Φ в виде одной линейной частично упорядоченной последовательности вычислительных процедур, и будем называть ее *расчетной последовательностью*.

Постановка задачи. Дано множество переменных $S=(S_1, \dots, S_n)$, заданных своими значениями, и множество переменных $(Y,Z)=(y_1,\dots,y_m,z_1,\dots z_K)$, подлежащих вычислению. Задана вычислительная модель U, отражающая свойство взаимосвязи проектных переменных. Требуется на модели U построить расчетную последовательность $RPOSL$, включающую в себя функции:

$$y_1=f_1(S_1,\dots,S_n),$$
$$\dots\dots\dots\dots$$
$$y_m=f_m(S_1,\dots,S_n),$$
$$z_1=\varphi_1(S_1,\dots,S_n),$$
$$\dots\dots\dots\dots$$
$$z_K=\varphi_K(S_1,\dots,S_n).$$

Примечание. Для вычислительной процедуры типа система двух уравнений функция имела бы вид

$$(y_1, y_2)=f_1(a_{11}, a_{12}, a_{21}, a_{22}, b_1, b_2).$$

Такая запись означает, что существует вычислительная процедура решения системы двух линейных уравнений с двумя неизвестными.

Задача решается в два этапа. Сначала осуществляется прямое планирование - от исходных переменных к вычисляемым: $S \to (Y, Z)$. При этом строится расчетная последовательность $RPOSL$, обладающая избыточностью, т.е. в ней могут быть процедуры, для которых заданы как входные, так и выходные данные. На расчетной последовательности $RPOSL$ осуществляется обратное планирование вида $(Y, Z) \to S$, позволяющее выявить и удалить избыточные процедуры,

Для понимания сущности алгоритма достаточно рассмотреть прямое планирование.

Для каждой переменной $s_i \in S$ на модели строится множество всех цепочек, исходящих из этой переменной. При построении цепочек модель U просматривается полностью. При этом цепочки могут быть фиктивными или искомыми. Искомая та цепочка, в которой встречается процедура, вычисляющая переменную из объединения множеств (Y, Z).

Запоминаем искомые цепочки, исходящие из переменной S_i. Аналогично запоминаем искомые цепочки, исходящие из всех остальных переменных множества S. Далее формируем расчетную последовательность из всех этих цепочек, сохраняя при этом отношение порядка между процедурами и включая всякую вычислительную процедуру в расчетную последовательность не более одного раза.

Рассмотрим, как на модели U осуществляется построение цепочки, исходящей из переменной $s_i \in S$.

Обозначим $z(x_i)$ -множество функций, для которых переменная x_i является входной; $x(f_i)$ - множество переменных, для которых функция $f_i \in r(x_i)$ является выходной. Задачу построения цепочки на вычислительной модели можно рассматривать как задачу построения пути на двудольном ориентированном графе с начальной вершиной.

Имеем переменную S_i, заданную своим номером N_{S_i}. На модели U с помощью функции q_4 определяем совокупность функций $z(S_i)$, для которых переменная S_i является входной. По определенному правилу выбираем одну функцию $f_{i1} \in z(S_i)$, заданную своим номером, и с помощью функции q_3 на модели U определяем множество переменных $x(f_{i1})$, для которых функция f_{i1} является выходной. Для построения одной цепочки по определенному правилу выбираем одну переменную $x_{i1} \in x(f_{i1})$, заданную своим номером. Таким образом, на данный момент нами построена цепочка (S_i, x_{i1}), содержащая процедуру f_{i1} . Для продолжения построения цепочки осуществляем проверку переменной x_{i1} . Возможны следующие случаи:

x_{i1} - выходная переменная, не принадлежащая объединению множеств Y, Z . Это означает, что цепочка построена до конца, является фиктивной и надо переходить к построению новой цепочки;

x_{i1} - выходная переменная, принадлежащая множеству Y либо Z . Это означает, что построена искомая цепочка и надо переходить к построению новой цепочки;

x_{i1} - промежуточная переменная, не принадлежащая множествам Y, Z . Это означает продолжение построения цепочки на модели U ;

x_{i1} - промежуточная переменная, принадлежащая множеству Y либо Z . Это означает, что надо запомнить построенную часть цепочки и продолжать ее построение.

Операция продолжения построения цепочки повторяет описанную нами процедуру, если в качестве начальной вершины принять переменную x_{i1} . За один проход этой процедурой мы получим цепочку (x_{i1}, x_{i2}), содержащую функцию f_{i2} , вычисляющую переменную x_{i2} . Осуществляем проверку, результатом которой возможен один из четырех случаев.

Операция запоминания цепочки сводится к запоминанию номера функции в отведенном для этого массиве.

Операция переход к построению новой цепочки осуществляется путем "возврата на шаг" по старой цепочке. Это означает следующее. Не меняя цепочки, например (f_{i1} , f_{i2}), по заданному правилу на множестве $x(f_{i2})$ выбираем новую переменную x_{j2} . Здесь возможны два случая, когда: а) переменная x_{j2} существует; б) множество $x(f_{i2})$ пусто.

В первом случае после перехода к новой переменной из множества $x(f_{i2})$ выполняется операция продолжения построения цепочки с начальной вершиной x_{j2} . За один проход этой операции мы получим новую цепочку (S_i, x_{j3}), содержащую процедуры f_{i1} , f_{i2}, f_{i3} . Далее осуществляется анализ переменной x_{j3} , результатом которого возможен один из четырех случаев.

Во втором случае осуществляется "возврат на шаг" по старой цепочке с переходом ко множеству $z(x_{i1})$ и с удалением последней вычислительной процедуры из цепочки. На множестве $z(x_{i_1})$ по заданному правилу выбираем новую функцию f_{j2} , если $z(x_{i_1}) \neq \phi$. Для функции f_{j2} строим множество $x(f_{j2})$: выбираем переменную x_{j2} и подвергаем ее анализу, результатом которого может быть один из четырех случаев. Если выполняется условие $z(x_{i_1}) = \phi$, то выполняем операцию "возврат на шаг" и переходим ко множеству $x(f_{i1})$. Выбираем промежуточную переменную $x_{j1} \in x(f_{i1})$ при условии, что $x(f_{i1}) \neq \phi$, и подвергаем ее анализу, результатом которого возможен один из четырех случаев. Если выполняется условие $z(x_{i_1}) = \phi$, то выполнение операций "возврат на шаг" может привести к исходному множеству S и к построению цепочек с новой начальной вершиной. Алгоритм прямого планирования заканчивает свою работу тогда, когда при выполнении операции "возврат на шаг" осуществляется переход к исходному множеству S и когда это множество пусто.

Различные цепочки, начиная с некоторой процедуры, могут совпадать. Чтобы не повторять уже пройденные цепочки при переходе ко множеству $z(x_i)$ и выборе процедуры f_i , будем осуществлять проверку этой процедуры на принадлежность расчетной последовательности. В случае принадлежности надо переходить к операции построения новой цепочки, а построенную начальную часть цепочки вписать в расчетную последовательность так, чтобы она предшествовала процедуре f_i . В случае невыполнения условия принадлежности - продолжить построение цепочки.

4.4.4. Формирование алгоритмического описания проектной задачи

Под алгоритмическим описанием функции мы понимаем программу, написанную на алгоритмическом языке и реализующую функцию по всей ее области определения. Рассмотрим вопрос создания программы, реализующей вектор-функцию $Z = \Phi(S)$, $S \in \Pi(S)$ и вектор-функцию $Y = F(S)$, $S \in \Pi_{\bar{Z}}(S)$.

В основу автоматического построения программы, реализующей проектную задачу, положен алгоритм решения расчетной задачи. Опишем этот алгоритм:

1. Вычисление вектор-функции $Z = \Phi(S)$, $S \in \Pi(S)$.
2. Формирование области $\Pi_{\bar{Z}}(S)$.
3. Проверка

$$\Pi_{\bar{Z}}(S) = \phi \begin{cases} Да \rightarrow п^\circ 4, \\ Нет \rightarrow п^\circ 5. \end{cases}$$

4. Диалоговое управление процессом расчета с автоматическим выходом на п.I.
5. Вычисление вектор-функции $Y = F(S),\ S \in \Pi_{\bar{z}}(S)$.
6. Вычисление Y_{max}.

Заметим, что алгоритм является инвариантным по отношению к различным проектным задачам, к различным математическим моделям. Выполнение пп.I и 5 связано с расчетной последовательностью (4.2)и носит проблемный характер. Остальные пункты реализуются общесистемными средствами.

Разобъем расчетную последовательность (4.2) на две части согласно функциям $z = \Phi(S)$, $Y = F(S)$, обозначив их соответственно

$\varphi = (\varphi_{i1}, \ldots, \varphi_{ik})$ и $f = (f_{i1}, \ldots, f_{ip})$.

При этом может оказаться, что часть процедур из последовательности f используется в φ . Такое деление условно и осуществляется так: точкой деления является последняя в последовательности процедура, вычисляющая переменную вектора z .

Запишем последовательность следующим образом: (φ, f). Обозначим $S_m(S, \delta_S)$ системный модуль, формирующий очередную точку параллелепипеда действий $\Pi(S)$ и логическую переменную δ_S , принимающую истинное значение, если сгенерированы все точки области $\Pi(S)$; $pz(\delta_S)$ - процедура принятия решения по значению переменной δ_S .

Впишем эти процедуры в расчетную последовательность следующим образом:

$(\varphi, S_m(S, \delta_S), pz(\delta_S), f)$,

где процедура принятия решения является формализованной; если δ_S -ложно, то процедура $pz(\delta_S)$ автоматически передает управление на начало расчетной последовательности, в противном случае управление передается следующей процедуре.

Обозначим $S_m(\delta_z)$ - системный модуль, осуществляющий сравнение вычисленного значения переменной z со значением ограничения $\bar{z}$ и формирование логической переменной δ_z .

В последовательности φ выделим все процедуры, вычисляющие переменные z . Например, таких процедур две: φ_{i2} , φ_{ik} . Расширим текст подпоследовательности φ соответствующими системными модулями и получим следующую расчетную последовательность:

$(\varphi_{i1}, \varphi_{i2}, S_m(\delta_{z1}), \varphi_{i3}, \ldots, \varphi_{ik} S_m(\delta_{z2}), S_m(S, \delta_S), pz(\delta_S), f)$.

Системные модули $S_m(\delta_{z_1}), S_m(\delta_{z_2})$ будем считать неотъемлемой частью вычислительных процедур φ_{i2}, φ_{ik}, и свойства, характерные для вычислительных процедур, будем распространять на системные модули.

Обозначим φ_i^k k-кратное выполнение процедуры φ_i. Понятие кратности выполнения процедуры согласуется с понятием размерности переменной. Рассмотрим процедуру $y=\varphi(x_1, x_2)$, где переменные x_1, x_2 размерности I типа и k_1, k_2 - длины векторов x_1, x_2 соответственно. Будем говорить, что процедура φ является k_1 -кратной по переменной x_1, если эта переменная является для нее входной. Аналогично процедура φ является k_2 -кратной по переменной x_2.

Процедура может быть одновременно кратна по двум независимым переменным, если она кратна по каждой переменной. Кратность процедуры φ по двум переменным x_1 и x_2 определим как произведение кратностей ($k_1 \cdot k_2$). Если переменная S_i связана с переменными x_1, x_2, то кратность процедуры φ по S_i не должна превышать длину вектора S_i.

Соотношения между размерностью переменной и кратностью выполнения процедуры определим следующим образом:

размерности нулевого типа соответствует единичная кратность;

размерности I типа соответствует k_1 -кратность, где k_1 - длина вектора переменной;

размерности 2 типа соответствует $(k_1 \cdot k_2)$ кратность, где k_1 - длина строки прямоугольной таблицы, k_2 - длина столбца.

Если переменная x размерности нулевого типа связана с переменной y цепочкой вычислительных процедур, то каждая процедура будет единичной кратности по переменной x.

Определим кратность выполнения процедур подпоследовательности φ по исходным переменным следующим образом. Для каждой вычисляемой и промежуточной переменной x определим ее связность с исходными переменными S задачи. Определим размерность каждой переменной x, зная размерность исходных переменных и связность. Как следствие, определяются длины векторов, размеры прямоугольных таблиц и, следовательно, кратности вычислительных процедур. Предположим, что мы вычислили кратности выполнения процедур подпоследовательности φ и они имеют следующие значения $(q_1, q_2, \dots, q_k)$. Тогда с учетом кратности выполнения процедур при вычислении вектор-функции $Z=\Phi(S)$ по области $\Pi(S)$, расчетная последовательность будет иметь вид

$$\mathcal{S}_{i1}^{q_1}(\mathcal{S}_{i2}\,S_m(\delta_{z_1}))^{q_2}\mathcal{S}_{i3}^{q_3},\ldots,(\mathcal{S}_{ik}\,S_m(\delta_{z_2}))^{q_k},$$
$$S_m(S,\delta_S),\ pz(\delta_S),f$$

Обозначим $S_m(\delta_n)$ - системный модуль, формирующий область $\Pi_{\bar{z}}(S)$ и логическую переменную δ_n, принимающую истинное значение, если область $\Pi_{\bar{z}}(S)$ существует. После вычисления вектор-функции $z=\Phi(S)$ по области $\Pi(S)$ со сравнением на выполнение ограничения необходимо выделить область $\Pi_{\bar{z}}(S)$, если такая существует. Для этого вставим в текст расчетной последовательности системный модуль $S_m(\delta_n)$:

$$\mathcal{S}_{i1}^{q_1},(\mathcal{S}_{i2},S_m(\delta_{z_1}))^{q_2},\mathcal{S}_{i3}^{q_3},\ldots,(\mathcal{S}_{ik},S_m(\delta_{z_2}))^{q_k},S_m(S,\delta_S),pz(\delta_S),S_m(\delta_n),pz(\delta_n),f,$$

где процедура принятия решения является формализованной и автоматически передает управление на вычисление вектор-функции $Y=F(S)$, если δ_n истинно. В противном случае для расчетных задач решение заканчивается указанием, что область $\Pi_{\bar{z}}(S)$ не существует. Для проектных задач управление автоматически передается на диалоговые средства управления процессом расчета с безусловной передачей управления на начало расчетной последовательности. Если область $\Pi_{\bar{z}}(S)$ существует, то осуществляется k кратное выполнение подпоследовательности f, где число k не может быть вычислено, пока не построена область $\Pi_{\bar{z}}(S)$.

До начала вычислений по размерности задачи можно узнать вид прообраза. Кратность выполнения подпоследовательности f можно вычислить автоматически, если указать правило формирования дискретных точек вдоль кривой, либо вдоль поверхности, либо по телу.

Предположим, что в задаче имеется условие $Y_ц = y_{max}$. Обозначим через $S_m(max)$ - системный модуль поиска максимального значения на множестве вычисленных значений переменной; $S_m(\delta_S)$ - системный модуль генерации точек области $\Pi_{\bar{z}}(S)$. Модуль формирует логическую переменную δ_S, принимающую истинное значение в случае, когда все точки области сгенерированы.

Окончательно расчетная последовательность будет иметь вид

$$\mathcal{S}^q,S_m(S,\delta_S),pz(\delta_S),S_m(\delta_n),pz(\delta_n),f^k,S_m(\delta_S),pz(\delta_S),S_m(max). \quad (4.3)$$

Здесь процедура принятия решения является формализованной и автоматически передает управление на вычисление вектор-функции $Y = F(S)$, т.е. на начало подпоследовательности f в очередной точке области $\Pi_{\bar{z}}(S)$. В противном случае управление передается следующей в последовательности процедуре. Расчетную последовательность вида (4.3) назовем **алгоритмическим описанием проектной задачи**.

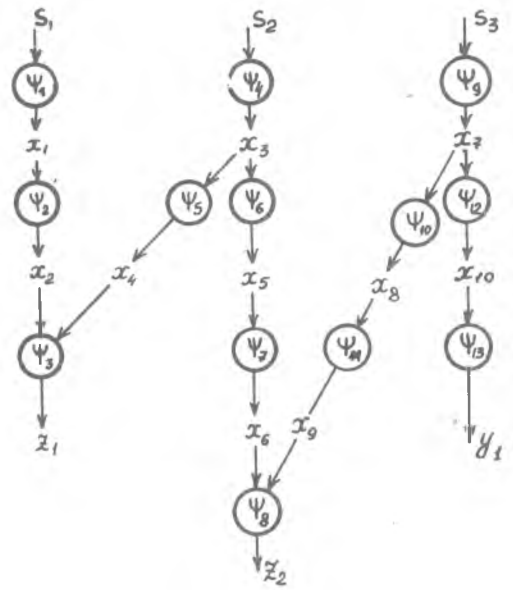
Таким образом, задача построения алгоритмического описания заключается в определении кратностей вычислительных процедур, во вписывании системных модулей в текст расчетной последовательности, в организации вычислительного процесса.

4.4.5. Алгоритм автоматического выполнения проектной задачи

Вопрос организации вычислительного процесса сводится к упорядочению процедур в расчетной последовательности. С учетом понятия кратности есть смысл говорить об упорядочении подпоследовательностей разных кратностей. Этот вопрос представляет значительный интерес, поскольку тесно связан с методом решения задачи. Пусть, к примеру, рассматривается проектная задача, в которой три переменных S_1 , S_2 , S_3 заданы интервалом значений. Рассмотрим три случая, содержащих ограничения типа равенство. В первом случае одно ограничение, во втором - два, в третьем - три. Это равносильно тому, что в первом случае областью $\Pi_{\bar{z}}(S)$ будет поверхность, во втором случае - кривая, в третьем - точка (в предположении, что решение во всех случаях существует).

Алгоритмическое списание желательно построить так, чтобы в одном случае легко было получить точки поверхности, а затем их генерировать, во втором случае - точки кривой, в третьем - расчетная последовательность должна быть близка к методу решения трех уравнений с тремя неизвестными и т.д. Поэтому автоматическое выполнение проектной задачи в значительной мере опирается на ее размерность, на вид функциональных зависимостей, реализуемых посредством процедур, и в каждом конкретном случае задача решается поразному.

Рассмотрим задачу организации вычислительного процесса на примере проектной задачи размерности (3,I,2) (рис.4.I0). Расчетная последовательность задачи может иметь следующий вид:



Р и с. 4.10. Вычислительная модель проектной задачи (3, 1, 2)

$\psi_1, \psi_2, \psi_4, \psi_5, \psi_3, \psi_6, \psi_7, \psi_9, \psi_{10}, \psi_{11}, \psi_8, \psi_{12}, \psi_{13}$.

Определим кратность вычислительных процедур предполагая, что длины векторов S_1, S_2, S_3 соответственно равны K_1, K_2, K_3. Расчетная последовательность с учетом кратностей будет иметь следующий вид: $\psi_1^{K_1}, \psi_2^{K_1}, \psi_4^{K_2}, \psi_5^{K_2}, \psi_3^{K_1K_2}, \psi_6^{K_2}, \psi_7^{K_2}, \psi_9^{K_3}, \psi_{10}^{K_3}, \psi_{11}^{K_3}, \psi_8^{K_2K_3}, \psi_{12}^{K}, \psi_{13}^{K}$.

На вычислительной модели выделим подпоследовательности одинаковой кратности и перепишем расчетную последовательность

$$f_1^{K}, f_4^{K_2}, f_3^{K_2}, f_2^{K_1 \cdot K_2}, f_5^{K_2}, f_8^{K_3}, f_7^{K_3}, f_6^{K_2 \cdot K_3}, f_9^{K}, \qquad (4.4)$$

где $f_1 = (\psi_1, \psi_2)$, $f_2 = \psi_3$, $f_3 = \psi_5$, $f_4 = \psi_4$, $f_5 = (\psi_6, \psi_7)$, $f_6 = \psi_8$, $f_7 = (\psi_{10}, \psi_{11})$, $f_8 = \psi_9$, $f_9 = (\psi_{12}, \psi_{13})$,

K – кратность заданной на области $\Pi_{\bar{Z}}(S)$ подпоследовательности f_9, вычисляющей переменную $y_1 \in Y$.

Приведем без доказательства утверждение: любые две подпоследовательности, кратные по разным переменным, не связаны. Следствие: любые две подпоследовательности, кратные по разным переменным, могут быть переставлены в расчетной последовательности.

Приведем последовательность (4.4) к более компактному виду путем объединения подпоследовательностей одинаковой кратности.

Получим

$$p_1^{K_1}, p_2^{K_2}, p_3^{K_1K_2}, p_4^{K_3}, p_5^{K_2 \cdot K_3}, p_6^{K}. \qquad (4.5)$$

где $p_1 = f_1$, $p_2 = (f_4, f_3, f_5)$, $p_3 = f_2$, $p_4 = (f_8, f_7)$, $p_5 = f_6$, $p_6 = f_9$.

Справедливо следующее утверждение: в расчетной последовательности число подпоследовательностей разной кратности не более 2^n, где n - число исходных переменных S.

Для организации вычислительного процесса построим таблицу кратностей (табл. 4.5) следующим образом:

$$a_{ji} = \begin{cases} 1, \text{ если подпоследовательность } p_i \text{ кратна по переменной } S_j, \\ 0, \text{ в противном случае.} \end{cases}$$

Таблица 4.5

Кратность выполнения процедур

Исходные переменные	Подпоследовательности					
	p_1	p_2	p_3	p_4	p_5	p_6
S_1	I	0	I	0	0	0
S_2	0	I	I	0	I	0
S_1	0	0	0	I	I	I

Зададим функцию: $q_8 : S_j \to (p_{i1}, \ldots, p_{ik})$, позволяющую определять подпоследовательности, кратные по S_j переменной. Каждой последовательности p_i поставим в соответствие пару

$q_9 : p_i \to (l_i ; np_i)$,

где l - длина i -й подпоследовательности, определяемая числом входящих в нее вычислительных процедур; np_i - начальный адрес i -й подпоследовательности.

Вычислительный процесс на расчетной последовательности организован следующим образом. Генерируется очередная g -я узловая точка (S_1^g , S_2^g , S_3^g) параллелепипеда действий. Для $n = 3$ всякая g-я точка может отличаться от $(g-1)$ -й одной, двумя, либо тремя координатами. Пусть к примеру произошло изменение по переменной S_2. Тогда на g -м шаге выполняются подпоследовательности, кратные по переменной S_2. С помощью функции q_8 по табл. 4.5 находим такие подпоследовательности. Управление передается на выполнение соответствующей подпоследовательности согласно соотношению q_9.

4.4.6. Диалоговое управление процессом расчета

Для реализации свойства итеративности процесса проектирования, соответствующего пошаговому процессу принятия решения (3.6), введем взаимодействие проектанта с решаемой задачей посредством операторов двух типов: информационных и управляющих. *Информационные* операторы предназначены для работы с внутренней базой данных и выдачи информации о процессе расчета. *Управляющие* операторы предназначены для корректировки процесса расчета.

Информационные операторы

1. Оператор ВЫЗВАТЬ < имя переменной >

Предназначен для контроля вычисленных значений промежуточных переменных. Он дает информацию о протекании процесса расчета в интересующих нас точках и тем самым позволяет принять решение. Выполняется под действием монитора системы. После его выполнения управление передается монитору системы.

2. Оператор ПРЕРВАТЬ < имя переменной >

Осуществляется прерывание после вычисления значения переменной, указанной в операторе. Дает возможность решать задачу по этапам. Число этапов и их место в цепочке вычислений определяется на промежуточном уровне пользователем. Оператор должен быть описан в постановке задачи (ТЕКСТ 2). После его выполнения управление передается монитору системы. Продолжение счета осуществляется по пустому оператору.

3. Оператор ВОЗВРАТ НА ШАГ < имя переменной >

Позволяет проанализировать результаты расчетов в их обратной последовательности. Действие оператора распространяется до переменной, указанной в операторе. После его выполнения управление передается монитору системы.

Управляющие операторы

1. Оператор ИСПРАВИТЬ < имя переменной > = < новое значение переменной > . В действии оператора возможны две ситуации:

старое значение переменной меняется на новое;

старый интервал значений переменной либо заменяется одним значением, не принадлежащим этому интервалу, либо старое значение переменной заменяется интервалом значений.

В первом случае управление передается подсистеме A_5 , интерпретирующей выполнение проектной задачи с новым значением переменной. Задача выполняется с того места расчетной последовательности, где впервые появляется эта переменная.

Во втором случае управление передается подсистеме A_4 , формирующей алгоритмическое описание проектной задачи с последующим ее выполнением.

2. Оператор ИЗМЕНИТЬ < старое имя программного модуля > , < описание нового программного модуля >

Под действием этого оператора управление передается подсистеме A_1 , формирующей информационное описание математической модели объекта. Осуществляется коррекция математической модели объекта. На этом сеанс может быть окончен или продолжен с постановки задачи. После выполнения оператора управление передается монитору системы.

3. Оператор ДОБАВИТЬ < ограничение >

В действии оператора возможны две ситуации:

вводимое дополнительное ограничение осуществляется по вычисляемой переменной, входящей в проектную задачу;

вводимое дополнительное ограничение осуществляется по переменной, отсутствующей в проектной задаче.

В первом случае, если это ограничение типа равенство и $n > K1$, то управление передается блоку формирования решения задачи путем построения кривой с размерностью на единицу меньше размерности исходной задачи. При $n = K1$ решением будет точка в n -мерном пространстве. При $n < K1$ осуществляется проверка: удовлетворяет ли имеющееся решение в виде точки уравнению кривой из дополнительного ограничения. Если дополнительное ограничение типа неравенство, то строится новое решение как подмножество решения исходной задачи.

Во втором случае управление передается на выполнение подсистем A_3 , A_4 , A_5 . Алгоритмическое описание новой задачи будет содержать в качестве подмножества алгоритмическое описание исходной задачи. Но так как исходная задача уже решена, то можно ограничиться выполнением дополнительной цепочки для вычисления значений новой переменной. Далее проводится целиком тот же, что в первом случае, анализ.

4. Оператор УДАЛИТЬ < ограничение >

Под действием этого оператора изменится прообраз $\Phi^{-1}(\bar{z})$ проектной задачи. Если удалено ограничение типа равенство, то исходный прообраз в виде точки перейдет в кривую, а исходная кривая - в поверхность. Требуется построить новый более широкий прообраз. Если удаляемое ограничение типа неравенство, то тело исходного прообраза может расшириться за счет удаления одной из граничных поверхностей. Требуется построить новый более широкий прообраз.

Оператор наиболее употребителен в случае, когда в исходной задаче прообраз пуст.

5. Пустой оператор. По пустому оператору продолжается выполнение проектной задачи с прерванного места.

6. Оператор СТОП. Означает окончание единовременного сеанса. Содержимое проектной задачи запоминается на внешнем носителе под именем пользователя.

Для возобновления сеанса в следующий выход на ЭВМ в мониторе системы должны быть средства вызова с внешнего носителя данных задачи по имени пользователя.

4.4.7. Формирование технической документации

Во всяком техническом документе может быть как формализуемая, так и неформализуемая часть. Последняя должна быть написана человеком без применения ЭВМ.

Рассмотрим вопросы автоматического формирования формализуемой части документации.

Под технической документацией будем понимать унифицированный технический отчет, описывающий процесс математического моделирования проектной задачи в САПР с помощью текстовой и графической информации. Основным унифицированным элементом отчета является понятие проектной задачи, а также основные разделы отчета.

Раздел описания математической модели объекта. Пункт первый: физическая постановка задачи. Описание принципа работы процесса. Этот пункт может быть неформализуемым. Пункт второй: описание математической модели объекта. Содержит каталог всех переменных и соответствующие им идентификаторы. Описание расчетных формул с использованием идентификаторов переменных. Каталог функциональных зависимостей и соответствующие им идентификаторы. Пункт третий: вычислительная модель объекта. Содержит двудольный ориентированный граф связей переменных. В качестве исходных данных для этого раздела используется языковое описание объекта проектирования.

Раздел описания постановки задачи. Приводится языковое описание проектной задачи. Даются дополнительные более жесткие ограничения, если это допустимо решением. Указывается, значения каких ограничений надо либо изменить, либо вообще удалить из задачи, если при начальных ограничениях решения не существует.

Раздел описания метода решения. Известно, что проектной задаче на вычислительной модели соответствует не единственное алгоритмическое описание. Оно определяется как перестановкой подпоследовательностей, так и правилом генерации точек параллелепипеда действий $\Pi(S)$. Фиксированное алгоритмическое описание соответствует определенному методу решения.

Раздел описания результатов машинного эксперимента.

Содержит прообраз $\Pi_{\bar{z}}(S)$ задачи, значения вычисляемых переменных в виде таблиц и графиков. Построение прообраза $\Pi_{\bar{Z}}(S)$ в трехмерном пространстве можно осуществлять графически. Прообраз в четырехмерном пространстве можно представить либо как K_1 трехмерных сечений, K_1 - число дискретных точек по четвертой координате, либо как $K_1 K_2$ двумерных сечений, K_2 - число дискретных точек по третьей координате. С ростом размерности пространства S геометрическое построение прообраза становится технически трудоемким и длительным.

Внутренняя структура данных системы позволяет представить результаты расчета в виде таблиц и графиков. При разработке соответствующих сервисных программ над внутренней базой данных выходные результаты можно выдавать в различных формах.

Таким образом, на основе анализа процесса проектирования разработано программное обеспечение базовой САПР, содержащее общую структуру модели базовой САПР и логику ее функционирования, внутреннюю структуру данных системы, алгоритмы функционирования основных подсистем, задачу формирования технической документации.

Вопросы и упражнения

1. Объясните, как реализуются в модели базовой САПР основные свойства процесса проектирования: взаимосвязь проектных переменных, итеративность, многоэтапность.

2. Гарантируют ли диалоговые средства принятия решения сходимость проектной задачи за конечное число шагов?

3. Какие возможности предоставляют диалоговые средства для пользователя?

4. Изменится ли алгоритмическое описание проектной задачи, если: а) изменить значение ограничения; б) тип ограничения; в) расширить параллелепипед действий; г) увеличить размерность параллелепипеда действий?

5. Как изменится алгоритмическое описание проектной задачи, если удалить одно ограничение?

6. Можно ли после получения результата проектной задачи: а) повторить расчет с промежуточной точки; б) изменить порядок расчета; в) проанализировать значения вычисленных промежуточных переменных?

7. Что надо сделать, если после 20 итераций Вы не получили удовлетворительного решения?

Л и т е р а т у р а

1. Общеотраслевые руководящие методические материалы. - М.: Статистика, 1980. - 118 с.

2. Андреевский В.В. Основы теории полета: Учебное пособие. - М.: МАИ, 1981. - 70 с.

3. Краснощеков П.С., Морозов В.В., Федоров В.В. Декомпозиция в задачах проектирования. - Изв. АН СССР, сер. Техн.кибернетика, 1979, № 2, с. 7-17.

4. Краснощеков П.С., Морозов В.В., Федоров В.В. Последовательное агрегирование в задачах внутреннего проектирования технических систем. - Изв. АН СССР, сер. Техн.кибернетика, 1979, № 5, с. 5-12.

5. Бусленко Н.П., Калашников В.В., Коваленко И.Н. Лекции по теории сложных систем. - М.: Сов.радио, 1973. - 438 с.

6. Мишин В.П., Осин М.И. Введение в машинное проектирование летательных аппаратов. - М.: Машиностроение, 1978. - 128 с.

7. Стинрод М., Чинн У. Первые понятия топологии. - М.: Мир, 1967. - 223 с.

8. Грис Д. Конструирование компиляторов для цифровых вычислительных машин. - М.: Мир, 1975. - 544 с.

9. Калентьев А.А., Штейнберг С.М. Язык описания расчетных задач в базовой САПР: Методические указания. - Куйбышев: КуАИ, 1984. - 27 с.

О г л а в л е н и е

Предисловие 3
I. АНАЛИЗ ПРОЦЕССА ПРОЕКТИРОВАНИЯ ТЕХНИЧЕСКОГО ОБЪЕКТА...... 5
I.I. Примеры инженерных задач.............................. 5
I.2. Классификация инженерных задач........................ 8
I.3. Свойства процесса проектирования...................... II
I.4. Способы описания объекта проектирования............... I3
I.5. Определение расчетной задачи.......................... I5
2. ИССЛЕДОВАНИЕ ЗАДАЧ, ДОПУСКАЮЩИХ АВТОМАТИЧЕСКОЕ РЕШЕНИЕ... I7
2.I. Типы расчетных задач.................................. I7
2.2. Алгоритм решения расчетной задачи..................... 24
2.3. Решение уравнения на отрезке.......................... 27
2.4. Построение плоской кривой на прямоугольнике........... 29
2.5. Решение системы двух уравнений........................ 30
3. ИССЛЕДОВАНИЕ ЗАДАЧ, РЕШАЕМЫХ АВТОМАТИЗИРОВАННЫМИ СРЕДСТВАМИ.. 33
3.I. Определение проектной задачи.......................... 33
3.2. Проектные задачи без ограничений...................... 35
3.3. Проектные задачи с ограничениями...................... 35
3.4. Автоматизированные средства решения задач............. 36
4. СТРУКТУРА И ЛОГИКА ФУНКЦИОНИРОВАНИЯ БАЗОВОЙ САПР......... 40
4.I. Определение базовой САПР.............................. 40
4.2. Общая структура и схема функционирования.............. 4I
4.3. Внутренняя структура данных системы................... 45
4.4. Алгоритмы функционирования подсистем.................. 50
Л и т е р а т у р а.. 70

Св.план 1985, поз. 2149

Анатолий Алексеевич К а л е н т ь е в

ВВЕДЕНИЕ В АВТОМАТИЗИРОВАННОЕ ПРОЕКТИРОВАНИЕ

Редактор Е.Д.А н т о н о в а
Техн.редактор Н.М.К а л е н ю к
Корректор Н.С.К у п р и я н о в а

Подписано в печать 21.03.1985 г. ЕО 00271.
Формат 60х84 1/16. Бумага оберточная белая.
Печать оперативная. Усл.п.л. 4,0. Уч.-изд.л. 4,5.
Т. 500 экз. Заказ 2453 Цена 15 к.

Куйбышевский ордена Трудового Красного Знамени
авиационный институт имени академика С.П.Королева,
г. Куйбышев, ул.Молодогвардейская, 151.

Обл.тип.им.В.П.Мяги, г.Куйбышев, ул.Венцека, 60.